METHODS TOWARDS PRECISION BIOINFORMATICS IN SINGLE CELL ERA

YUE CAO

A thesis submitted in fulfilment of
the requirements for the degree of
Doctor of Philosophy

School of Mathematics and Statistics
Faculty of Science
The University of Sydney

February 2023

This thesis is dedicated to my family.

STATEMENT OF ORIGINALITY

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Signature:

Yue Cao, February 2023

ABSTRACT

Single-cell technology enables the observation of the molecular landscape of individual cells. This offers unprecedented opportunity to explore diversity of cells and is transforming precision medicine in recent years. Key to the effective use of single-cell data to understand disease is the effective analysis of information through bioinformatics methods. In this thesis, we examine and address several challenges in single-cell bioinformatics methods for precision medicine.

While most of the current single-cell analytical tools employ statistical and machine learning methods, deep learning technology has gained tremendous success in the computer science field. Combined with ensemble learning, this further improves model performance. Through a review article (Cao *et al.*, 2020), we share recent key development on this front and examine their contribution to bioinformatics research. We envisage that the common challenges and opportunities identified can inspire future application of ensemble deep learning technology to the single-cell field.

Bioinformatics tools often use simulation data to assess the proposed methodology but evaluation of the quality of single-cell RNA-sequencing (scRNA-seq) data simulation tools and the quality of the simulation data they produce is lacking. To address this gap, we develop a comprehensive framework (Cao *et al.*, 2021), SimBench, that examines a broad range of aspects from data properties to the ability to maintain biological signals, scalability and applicability. Using 35 scRNA-seq experimental datasets, we uncover performance differences among current simulation methods and highlight the varying difficulties and challenges in the simulation landscape.

While the key to precision medicine is the understanding of individual patient, there is yet little consensus on the best ways to compress information from

the complex data structures that single-cell technology produces to summary statistics that represent each individual. To address this gap, we present scFeatures (Cao *et al.*, 2022b), an approach that creates interpretable cellular and molecular representations for individuals. We demonstrate, using a collection of 17 datasets across different diseases, that summarising a broad collection of features at the sample level is both important for understanding disease mechanisms and for accurately classifying disease status.

Finally, using multiple COVID-19 single-cell data in a case study, we utilise scFeatures to generate molecular characterisation of individuals and illustrate the effect of ensemble learning as well as deep learning on improving disease outcomes prediction.

Overall, this thesis addresses several gaps in precision bioinformatics in the single-cell field by highlighting current research advances, developing methodologies towards this front and illustrating the practical uses of the methodologies through experimental datasets and case studies.

PUBLICATIONS

Most of the work presented in this thesis has appeared in publication. These are listed below:

- **Cao, Y.**, Geddes, T., Yang, J. and Yang, P., 2020. Ensemble deep learning in bioinformatics. *Nature Machine Intelligence*, 2(9), pp.500-508.
- **Cao, Y.**, Yang, P. and Yang, J., 2021. A benchmark study of simulation methods for single-cell RNA sequencing data. *Nature Communications*, 12(1).
- **Cao, Y.***, Lin, Y.* , Ormerod, J., Yang, P., Yang, J. and Lo, K., 2019. scDC: single cell differential composition analysis. *BMC Bioinformatics*, 20(S19).
- **Cao, Y.**, Lin, Y., Patrick, E., Yang, P. and Yang, J., 2022. scFeatures: Multi-view representations of single-cell and spatial data for disease outcome prediction. *Bioinformatics*, btac590

Besides the above studies which I led, during my candidature I have also collaborated with others and contributed on a number of projects. While not included in this thesis, some of these projects also contributes to the development of bioinformatics methods for the single-cell field. These works are:

- Lin, Y., **Cao, Y.**, Kim, H., Salim, A., Speed, T., Lin, D., Yang, P. and Yang, J., 2020. scClassify: sample size estimation and multiscale classification of cells using single and multiple reference. *Molecular Systems Biology*, 16(6).
- Yu, L., **Cao, Y.**, Yang, J. and Yang, P., 2022. Benchmarking clustering algorithms on estimating the number of cell types from single-cell RNA-sequencing data. *Genome Biology*, 23(1).

*Equal contribution

AUTHORSHIP ATTRIBUTION STATEMENT

Chapter 2 of this thesis is published as Cao *et al.* (2020). I conducted the literature review and wrote the majority of the manuscript under the supervision of A/Prof. Pengyi Yang and Prof. Jean Yang, with input from Thomas Geddes.

Chapter 3 of this thesis is published as Cao *et al.* (2021). I collected the data, conducted the experiment, interpreted the results and wrote the majority of the manuscript under the supervision of Prof. Jean Yang and A/Prof. Pengyi Yang.

Chapter 4 of this thesis is published as Cao *et al.* (2022b). I collected the data, conducted majority of the experiment and interpreted the results with input from Dr. Yingxin Lin and Dr. Ellis Patrick. I wrote the majority of the manuscript under the supervision of Prof. Jean Yang and A/Prof. Pengyi Yang.

I acknowledge that the publisher's policy grants permission for the use of published material within this thesis.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Signature:

Yue Cao, February 2023

As primary supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signature:

Jean Yang, February 2023

As secondary supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signature:

Pengyi Yang, February 2023

As secondary supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signature:

Shila Ghazanfar, February 2023

ACKNOWLEDGMENTS

While writing this thesis, it struck me that what I thought would be a long journey is ending much sooner than I had anticipated. This journey would not have been possible without the help of numerous people, for whom I express my gratitude here.

First, my gratitude goes to my supervisors Professor Jean Yang, Associate Professor Pengyi Yang and Dr Shila Ghazanfar for their support throughout my PhD period. I considered myself very fortunate to have had the opportunity to work with them during my PhD study and gain from their expertise of the field as well as their enthusiasm for research. They will continue to be my life-long role models.

I would also like to acknowledge the research group, Sydney Precision Bioinformatics Alliance, in general. My presentation at the Friday single-cell meeting and the research seminar to the group have always received many valuable feedback, which has greatly helped me to improve my work. It is also in this group that I have met many of my friends. Beside the intellectual exchange we had, this friendship brings lots of fun and joy to my PhD time. In particular, Yingxin and Yunwei for always being there to talk about research and life.

In addition, I gratefully acknowledge the funding received towards my PhD from the Australian Government and the University of Sydney. These funds have enabled me to make the most of my PhD candidature by focusing solely on my studies without having to make time for a part-time job.

Lastly, my sincerest gratitude goes to my family for their unwavering love and support. I could not have came this far without them.

CONTENTS

1

INTRODUCTION

Precision medicine involves measuring individuals' molecular profiles and predicting, based on the data, the disease subgroup these patients fall into and the treatment they would potentially best respond to. In 2018, the Office of the Chief Scientist of Australia announced the 2030 goal of public healthcare of quantifying the risk profiles in individuals and customising the care for each individual including early intervention and personalised treatments. The driving force behind the transition from standard care to precision care in the near future is the advances in high-throughput profiling technologies such as DNA and RNA sequencing and in particular the recent rise of single-cell technologies.

To transform healthcare through precision medicine is not without challenges. In particular, in the era of single-cell sequencing technology, single-cell data exhibit great differences to traditional "bulk" sequencing data generated from cell population. First of all, unlike bulk sequencing data with minimal zero values, single-cell RNA-seq data (scRNA-seq) data is characterised by its sparsity, with many data containing more than 90% of the values being zero due to both technical and biological effect. Considerable thought is required to handle and model this sparsity in all downstream analysis. Secondly, with the change in granularity from patient to cell being the sampling unit, the size of the data dramatically increases from a typical 20,000 genes by 50 samples matrix to a 20,000 genes by 1,000,000 cells matrix being the norm. This calls for the need to develop scalable methods. It is important to note that all these challenges stem from data science problem and central to precision medicine in the single-cell era is a data analysis challenge.

Our contribution to the field is as follows. First, we survey the literature space on the recent success of deep learning models and their application in the bioinformatics field to inspire future single-cell deep learning approaches. Second, we develop a systematic evaluation framework in response to the exponential increase in the number of single-cell bioinformatics tools and apply it to benchmark single-cell data simulation methods. Third, we develop a method to address the lack of approach on summarising the molecular profile at individual level for downstream precision medicine analysis. Together, these works contribute to precision medicine approaches using single-cell data and provide insights for future method development towards the field.

The rest of this chapter provides the background for the work presented in this thesis. Section 1.1 provides the background concept and introduces bioinformatics approaches for single-cell data analysis and outlines their applications in precision medicine. Section 1.2 then discusses the challenges of single-cell data analysis for precision medicine. Finally, section 1.3 presents an outline of the works presented in this thesis.

1.1 PRECISION BIOINFORMATICS IN THE SINGLE-CELL ERA

1.1.1 *Precision medicine*

Precision medicine describes the approach where the difference between individuals is taken into account to provide targeted medical advice or treatment for each group of patients. Often individuals' disease phenotypes and their therapeutic responses can be affected by the genetic profiles. A classic example of precision medicine is the improvement in drug targeting for patients with cystic fibrosis (Ashley, 2016). Molecular characterisation of patients discovered that 5% of patients harbour a genetic mutation that renders them ineffective to the medicine traditionally used for cystic fibrosis. This led to the development of a new therapeutic approach to specifically target the subgroup of patients.

It is to note that despite the promise of precision medicine, significant challenges from multiple aspects must be addressed. In terms of the technical aspects, there exist significant challenges in analysing data generated by the latest single-cell technology, as the data exhibit great differences to traditional "bulk" sequencing data. Besides the technical challenges that this thesis will shed light upon, a number of hurdles in other aspects must be addressed before realising the potential of precision medicine. For example, in terms of resources, precision medicine requires significant resources such as sequencing machines and experts to analyse the data. The resource could be a challenge for putting precision medicine into practice. In terms of data, there are questions regarding the cost of storing and maintaining the patient's genetic data and the privacy issues behind the storage and access of the data. Even when these hurdles are addressed, the implementation of precision medicine into everyday clinical practice requires significant change in the way healthcare is delivered currently and can potentially take decades for this transition to occur. There is still a long road ahead to achieve the goal of delivering better outcomes for patients.

1.1.2 *Single-cell sequencing technology*

Precision medicine is achieved through sequencing technology, a transformative technology that enables the molecular characterisation of individuals. Since the first-generation sequencing that costed $2.7 billion USD to sequence 20 individuals in 2001, decade later, the next-generation sequencing technology brought the cost per human genome down to less than $1000 USD (Goodwin *et al.*, 2016) and opens the door for sequencing technology to be readily used as a tool by clinicians and researchers. In 2009, single-cell technology is introduced (Tang *et al.*, 2009) and brings new promises into precision medicine. Before single-cell technology was introduced, sequencing technology captures the averaged measurement across multiple cells in a sample and is referred to bulk sequencing. On the other hand, single-cell technology sequences each individual cell and allows us to study molecular characteristics one cell at a time (Figure 1.1). This unlocks an unprecedented amount of novel analysis for precision

medicine by enabling scientists to extract cell type specific information such as cell type proportion and cell type interactions that are previously unattainable by bulk sequencing. The different types of single-cell technology introduced in recent years (e.g. single-cell transcriptomics sequencing, single-cell DNA sequencing, single-cell multiomics sequencing, single-cell spatial transcriptomics) and their potential to transform research are acknowledged by Nature as methods of the year in 2013, 2019 and 2021 (Nature Methods, 2014; Teichmann and Efremova, 2020; Marx, 2021). The current decade is truly an exciting time for single-cell technology and calls for the development of new data analysis tools to fully unleash the power of this new technology for precision medicine.

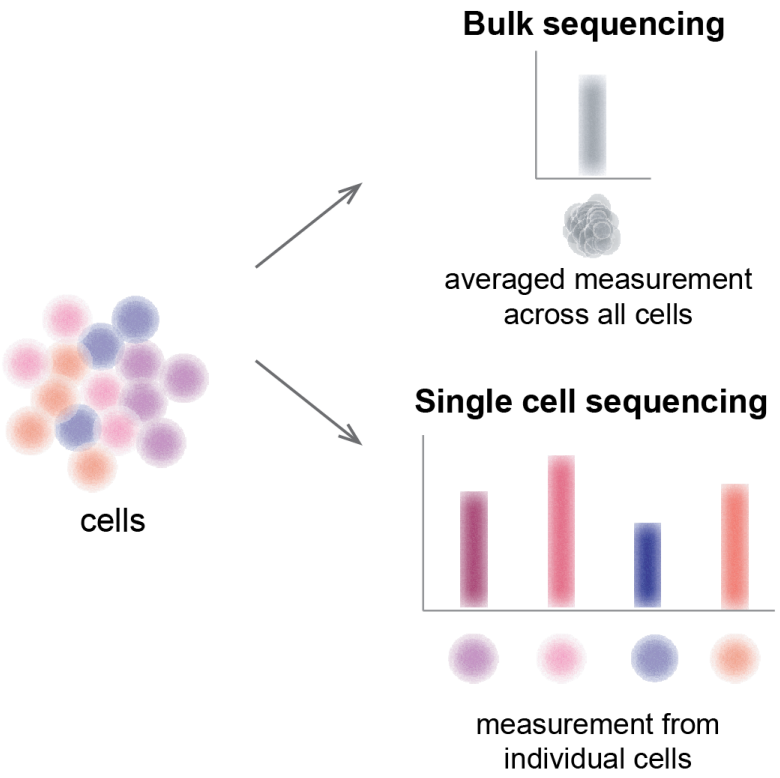


Figure 1.1: Schematic representation of bulk sequencing versus single-cell sequencing

1.1.3 *Bioinformatics approaches to single-cell data*

Similar to the introduction of new biotechnologies, the complexity of data generated by the new technology urges the development of new methodologies to extract information and understand the data. The data produced by single-cell technology is characterised by extreme sparsity, with a typical data containing 90% of zero entries (compared to traditional bulk sequencing where almost all values are non-zero) due to both biological and technical reasons. Here the entries refer to the gene expression level for a gene in a particular cell. In addition, since the data is at single-cell resolution, the sample size of typical data increases exponentially from a typical number of 50 (individuals) as in bulk sequencing to 1,000,000 (cells). The complex data structure at ultra-high-resolution of individual cell level has hence inspired more than 1000 methods for extracting meaningful information from this data (Zappia and Theis, 2021).

In order to extract novel insight from the molecular profile of individual cells captured by single-cell technology, current single-cell bioinformatics analytical approaches typically focus on the cell and gene levels. For example, differential expression methods identify potential features (such as genes) involved in biological condition by finding the difference in expression pattern (Squair *et al.*, 2021). Trajectory inference methods infer the relative position of each individual cell in context of their underlying biological process such as the pathogenesis from healthy to disease state (Saelens *et al.*, 2019). Gene regulatory network construction algorithms explore the regulatory relationship between genes and identify functional modules that are impacted by diseases (Pratapa *et al.*, 2020). Velocity analysis estimates the direction of transcriptomic change in each cell and provides information on the dynamic transition between cells (Bergen *et al.*, 2021).

Since 2021, with the maturation of sequencing platforms and the reduction in sequencing cost and labour, there is an increasing number of multi-condition multi-patient studies (Junttila *et al.*, 2022). A challenge for precision medicine applications in the current single-cell era has emerged as the development of

single-cell methodologies that represent individuals by summarising the information extracted at gene and cell levels for better understanding at the individual level.

1.1.4 *Precision medicine applications leveraging single-cell data*

Single-cell data can facilitate precision medicine by revealing the characteristic of individual cell in disease progress and treatment response and enabling the identification of critical cellular mechanism in a cell type specific manner. Here we describe two applications of precision medicine that are enabled by single-cell data.

An active focus of precision medicine research is to dissect intratumor heterogeneity. Heterogeneity refers to the fact that the molecular characteristics of cells vary across individual cells, even for cells belonging to the same cell type. This has important implications for diseases such as cancer, as the direct consequence is that not all cells respond equally to cancer treatment (Goldman *et al.*, 2019). This can be naturally explored using single-cell data as single-cell data is at the resolution of individual cells. A recent single-cell paper (Kinker *et al.*, 2019) highlighted the concept of defining patients not by cancer type but by their programs of intratumor heterogeneity. Through examining single cells from 22 cancer types, it is found that subpopulations of cells across multiple cancer types display common patterns of heterogeneity associated with biological processes. As some of these recurring programs of heterogeneity are related to drug resistance, these findings point to the possibility of defining tumors by expression programs instead of cancer type and elucidate new insights for future cancer therapeutics.

The recent COVID-19 pandemic is another highlighting example of how precision medicine actively uses single-cell technology to understand the differential impact and response across individuals to the virus. The clinical consequences of COVID-19 infection lie on a broad spectrum, from asymptomatic, to severe conditions requiring ventilation and fatality. In 2021, within two years since

COVID-19 virus dominated worldwide, there have been 62 published studies that utilised single-cell technology including flow cytometry, mass cytometry (CyTOF), scRNA-seq, CITE-seq, scBCR/TCR-seq to study COVID-19 patients using various approaches such as differential expression and cell-cell communication (Tian *et al.*, 2022) and a number of curated databases on COVID-19 single-cell datasets (Jin *et al.*, 2021; Qi *et al.*, 2022; Tian *et al.*, 2022) . Leveraging the cellular characterisation at single-cell resolution, these studies have enabled a number of novel discoveries at cell type specific levels, such as the significant difference in cell type composition between mild and severe conditions, the response mechanism in each cell type and the key signalling network between cell types in different severity.

1.2 CHALLENGES IN PRECISION BIOINFORMATICS IN THE SINGLE-CELL ERA

While single-cell offers unprecedented opportunities for precision medicine, the significant difference between single-cell sequencing data from traditional bulk sequencing data poses a significant challenge to the effective utilisation of single-cell data for precision medicine. Central to this is a data analysis challenge, where novel bioinformatics approaches must be developed to unmask the pattern behind the data to enable downstream application in precision medicine. Here, we outline a number of major challenges for precision bioinformatics in the single-cell era and how we contribute to the field by addressing these challenges.

1.2.1 *Model stability*

A unique characteristic of single-cell sequencing data is that typically 90% of the values are zero (Ding *et al.*, 2020). This extreme sparsity, also called zero inflation, arises from two factors, the biological factor that not all genes are expressed in a given time in a cell and the technical factor of the limitation of current sequencing technology. On the other hand, bulk sequencing data, where

the data are aggregated measurements across many cells, does not encounter such issues and generally has sparsity as low as 10% (Deaton *et al.*, 2011). The extreme sparsity in single-cell data therefore poses a number of challenges to existing statistical methods for analysing bulk sequencing data for precision medicine applications.

One of the key challenges caused by the data sparsity is the reduction of model stability compared to those built from traditional bulk sequencing data. This is because in each individual cell, the amount of signal detected is much less compared to bulk sequencing, where the signal comes from the pool of thousands of cells (Figure 1.2). Statistical models therefore become more unstable as the stability of the signal reduces. This calls for the development of more robust methods such as pooling signatures from similar features to enable more stable parameter estimation (Hafemeister and Satija, 2019).

Ensemble learning, the strategy of combining multiple models, is known to increase the stability of model output (Dong *et al.*, 2020) and has been applied to the area of bioinformatics (Yang *et al.*, 2010a). In this thesis, we examine recent ensemble strategies in conjunction of deep learning model that have been developed in various areas of bioinformatics research including disease study and discuss the common challenges and opportunities. We envisage the examination will inspire the development of novel and more stable learning models in the single-cell field for precision medicine applications.

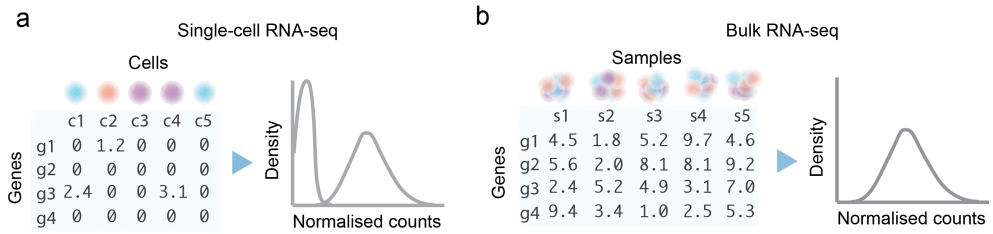


Figure 1.2: Data characteristics of single-cell RNA-seq and bulk RNA-seq. (a) shows the normalised count matrix of single-cell RNA-seq, where the majority of values are zero, resulting in a bimodal distribution. (b) shows the normalised count matrix of bulk RNA-seq, which has a normal distribution.

1.2.2 *Model scalability*

With the shift in resolution from individual sample to individual cells being the unit of measurement, another key analytical challenge is the scalability of methods. In bulk RNA-sequencing, a sample size of 50 patients (and thus a dataset with 50 columns) is a typical number. In contrast, single-cell data is being increasingly large in size with datasets in recent years containing millions of cells (Figure. 1.3) (and thus datasets with millions of columns). Due to the enormous dataset size, many methods originally developed for bulk sequencing are no longer computationally feasible on single-cell data. Scalability is therefore a common concern for methods developed for any aspect of single-cell analysis including analysis for precision medicine.

Deep learning is known to thrive with large input data in terms of model performance and at the same time being highly scalable unlike traditional machine learning. It has gained tremendous success in the computer vision field. Recently, deep learning has also received increasing attention in the single-cell field (Zappia and Theis, 2021; Bao *et al.*, 2022) and saw application in a range of scalable analysis of large-scale single-cell data such as imputation (He *et al.*, 2020), clustering (Xie *et al.*, 2020), batch correction (Zou *et al.*, 2021), and joint analysis of single-cell multi-omics (Gayoso *et al.*, 2021).

In this thesis, the aforementioned survey explores the current landscape of the deep learning methodology in conjunction with ensemble learning for bioinformatics applications. We envisage the survey will inspire the development of novel scalable and robust learning approaches in the single-cell field for precision medicine applications.

1.2.3 *Method evaluation*

The unique characteristic of single-cell data renders a number of well-established statistical models for bulk sequencing data no longer optimal for single-cell data and has prompted a new wave of method development specifically for

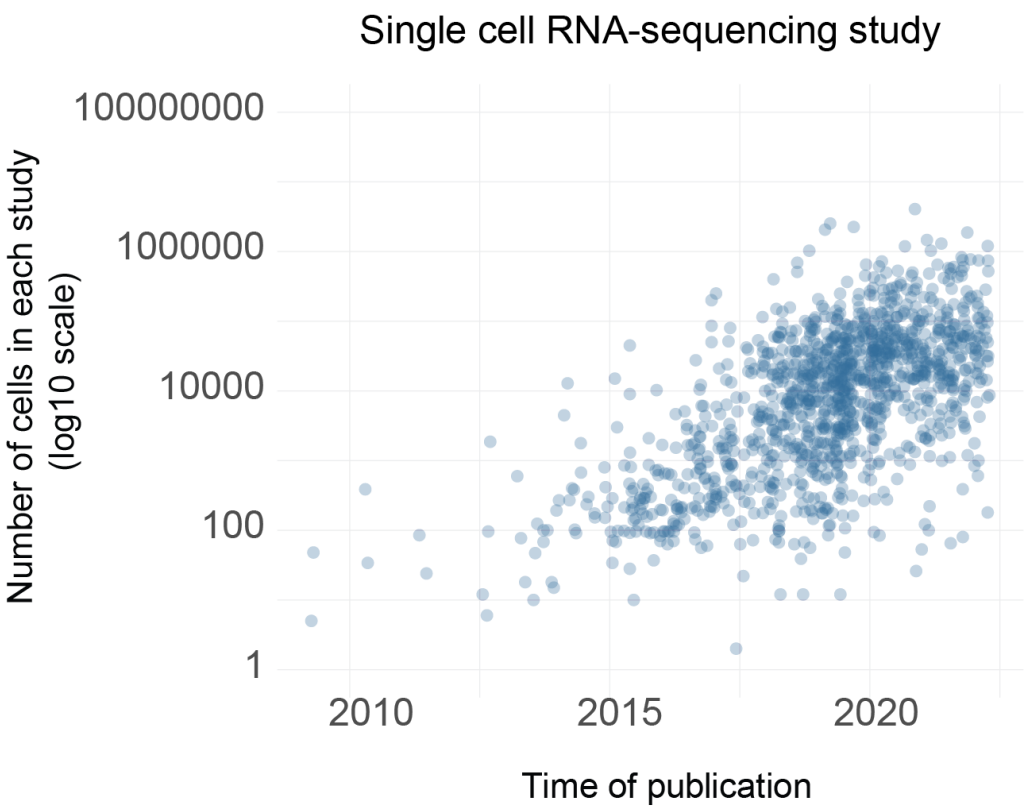


Figure 1.3: The increase in the number of cells in single-cell RNA-sequencing study, as demonstrated by data sourced from the `www.nxn.se/single-cell-studies/gui` database as of August 4^{th} 2022.

single-cell data. In precision medicine, a common analysis, for example, is the identification of differentially expressed (DE) genes between different conditions. While the limma package (Ritchie *et al.*, 2015a) is one of commonly used methods for this task, it is developed for bulk sequencing data and is based on linear model which assumes a unimodal distribution. In single-cell data, the zero inflation results in a bimodal pattern, requiring new methods that accommodates for this (Soneson and Robinson, 2018a; Mou *et al.*, 2020). During the past few years, over 100 tools are available for single-cell differential expression alone (Figure 1.4).

As the number of tools continues to increase, a challenge associated with it is the selection of the most appropriate methodologies for the data and research question at hand. Having a set of guidelines or comparison of existing methods becomes increasingly necessary for applied researchers in precision medicine

to perform best-practice data analysis and would accelerate scientific discovery. Therefore, there is an urgent need for the establishment of benchmarking datasets and frameworks and the comprehensive evaluations of single-cell methods beyond their publications.

In the precision medicine field, of particular need is the benchmarking of simulation methods. In precision medicine, the ground truth behind patient's molecular profile is unattainable. Therefore, simulation data is often needed to assess the performance of new methods. Over the past few years, a number of single-cell data simulation methods have been developed on this front for various use cases such as the simulation of DE genes with known fold change for the development of DE methods. However, there is a current lack of benchmark studies on these simulation methods or a systematic evaluation framework for performing such benchmark studies. As the quality of simulation data and their ability to accurately reflect biological data can directly impact the downstream methodological development on precision medicine, a pressing challenge is to comprehensively assess current single-cell data simulation tools. Furthermore, the establishment of such benchmarking datasets and framework can be used in or inspire future benchmarking of single-cell methods in other applications.

In this thesis, we address both of the above challenges via the development of benchmark datasets and evaluation framework for assessing scRNA-seq simulation tools. We curate a collection of scRNA-seq datasets containing of 35 data that have been carefully selected to cover a broad range of sequencing platforms, sampling tissues, cell types, number of cells and organisms and can be used for future evaluation studies on other single-cell methodologies. We develop a evaluation framework that contains multiple evaluation aspects that are both specific for assessing scRNA-seq simulation tools and also general criteria such as scalability analysis and usability assessment that can inspire evaluation studies on other single-cell methodologies. Moreover, using the datasets and framework, we benchmark published scRNA-seq simulation tools and address the lack of such study in the current literature.

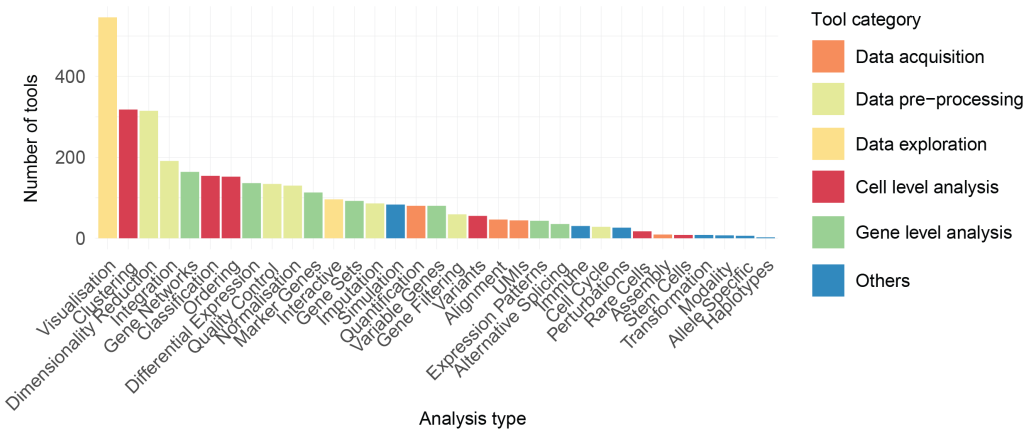


Figure 1.4: Number of single-cell tools in each analysis type, coloured by the broad category.

1.2.4 *Data interpretation*

The significant increase in sampling resolution from individual sample containing thousands of cells being the unit to individual cell being the unit causes a significant increase in data complexity. The complexity of the data is both an opportunity and also a challenge to data interpretation. As aforementioned, over 1000 computational tools have been developed to unravel the data from multiple aspects and derive useful insights that are unattainable by bulk RNA-seq.

However, current single-cell analytical tools mostly focus on the characterization of individual cells or genes (Figure 1.4). There is a lack of defined framework for summarising the cellular profiles of individual cells into a patient profile to enable the interpretable of data at individual level (Figure 1.5). This has important implications for precision medicine analysis, in which individuals are the units of interest. While the original expression matrix in the format of genes by cells can be used as input to infer the transcriptomics change for a particular individual, the ability to represent an individual with other layers of information (e.g. interaction of genes and pathways) could uncover additional insights. Novel methods that construct molecular representations of individuals for downstream exploration are therefore of urging need for the effective usage of single-cell data for precision medicine.

We address this challenge in Chapter 4 by developing a novel framework for creating biologically relevant learning features across multiple feature types and enabling a multi-view representation of individuals for downstream modelling and interpretation of disease outcomes. In total, we constructed a total of 17 different feature types across six distinct feature categories of i) cell type proportions, ii) cell type specific gene expressions, iii) cell type specific pathway expressions, iv) cell type specific cell-cell interaction (CCI) scores, v) overall aggregated gene expressions and vi) spatial metrics. Using a collection of single-cell patient datasets across multiple diseases and cell types, we demonstrate that different feature types are useful for predicting the disease outcomes in different datasets and even for the same groups of patients in different treatment stages, thereby showing the importance of representing individuals using a broad collection of biological features.

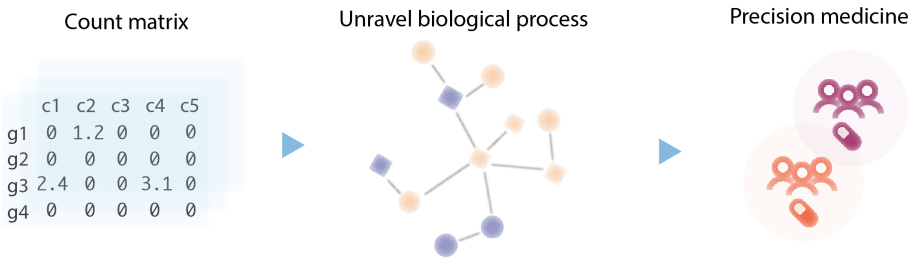


Figure 1.5: Data interpretation as a challenge in the effective utilisation of single-cell data for precision medicine.

1.3 THESIS OUTLINE AND CONTRIBUTIONS

This thesis is dedicated to exploring the various challenges and opportunities in precision medicine research with a focus on the field of single-cell data sequencing. The rest of the thesis is organised into five chapters (Figure 1.6), each of which addresses a specific challenge in the single-cell analysis workflow. Chapter 2 explores ensemble deep learning methods that have achieved success in the bioinformatics field, including the single-cell field. An understanding of the state-of-the-art approaches can foster new applications utilising such strategies

for single-cell data, for example, to address the dimensionality and sparsity challenge of the data characteristics. To assess the performance of methodologies requires realistic simulation data containing group truth, therefore Chapter 3 then establishes a comparison framework for single-cell data simulation studies. Chapter 4 develops a patient-level analysis strategy from the single-cell data to enable precision medicine applications. Chapter 5 presents a case study that utilizes the concepts discussed in the preceding chapters and further illustrates how these ideas come together to address a precision medicine problem. The details of each chapter are discussed below:

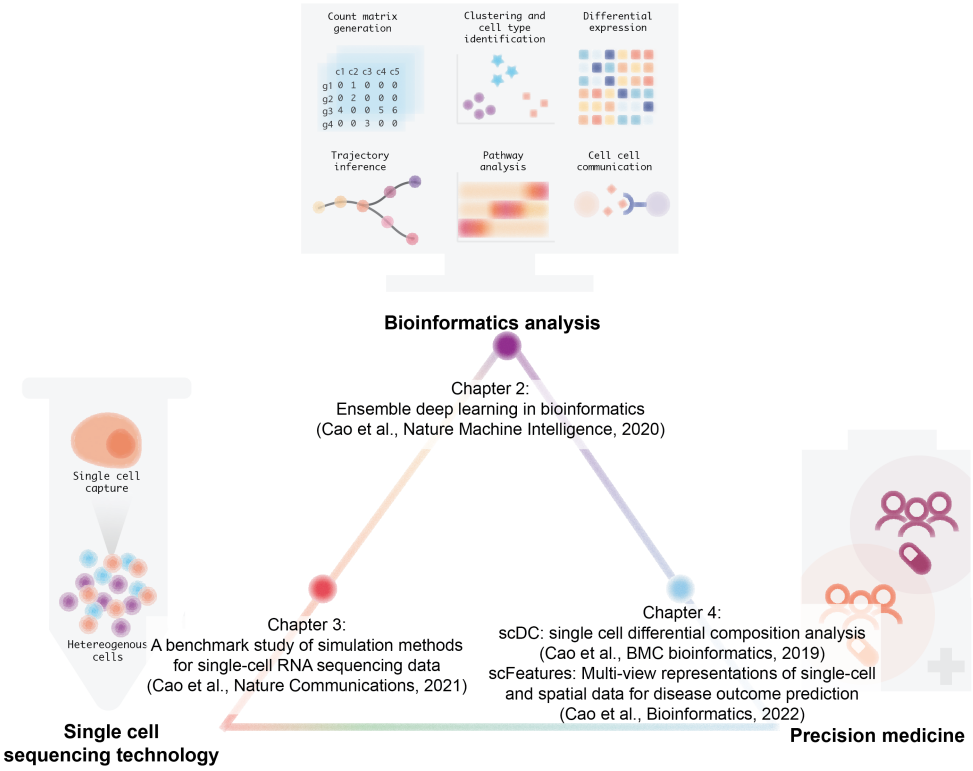


Figure 1.6: Schematic representation of the workflow relating to this thesis, from single-cell data collection by sequencing technology, to examples of bioinformatics analysis for interpreting single-cell data and ultimately, the goal of aiding precision medicine. The key studies in each chapter of the thesis and their approximate positions in the analysis workflow are also shown.

Chapter 2. Ensemble deep learning in bioinformatics. This chapter examines the current application of ensemble deep learning in a range of bioinformatics

applications including disease outcome analysis. Deep learning is powered by the ability to handle and improve model training from large-scale data. Ensemble learning is known to increase the stability of model output. We envisage this chapter can inspire novel applications that utilise the synergistic power brought by the combination of these two machine learning techniques to address both the issue of stability and scalability caused by the unique characteristic of single-cell data and to promote the usage of single-cell data for precision medicine research.

Chapter 3. A benchmark study of simulation methods for single-cell RNA sequencing data. This chapter curates a collection of single-cell benchmarking datasets that can be used by any evaluation of single-cell tool in general, as well as develops a novel evaluation framework, SimBench, for evaluating scRNA-seq data simulation tools. We utilise SimBench to evaluate current simulation tools, thereby addressing the lack of such study in the current literature. In the era where single-cell methods are growing at an incredible rate, the set of recommendations provided in our study allows method developers and users to efficiently identify the strengths and limitations of current methods and select the most appropriate methodology for their data analysis. Both the benchmarking datasets and evaluation framework have been made publicly available as R packages to the research community and we envisage this work will promote the future development of methods for precision medicine.

Chapter 4. scFeatures: Multi-view representations of single-cell and spatial data for disease outcome prediction. This chapter develops a novel method, scFeatures, for constructing molecular profiles of individuals and enabling the downstream exploration of precision medicine applications such as disease outcome prediction. As the success of modelling and interpretation of diseases requires biologically relevant learning features from the data, we generate the feature vectors based on a broad range of analytical approaches in literature from cell type specific gene expression to measures of cell-cell (ligand receptor co-expression) interaction.

Chapter 5. Case study of precision bioinformatics on COVID-19 single-cell data. In this chapter, we utilise the work presented in Chapter 2,3 and 4 and conduct a comparison study on COVID-19 patient severity prediction. We use the patient representation generated by scFeatures as the input and evaluate the combined impact of the choice of learning framework ranging from machine learning to deep learning and choice of ensemble strategy in severity prediction.

Chapter 6. Conclusion and future work. The final chapter summarises the contribution of this thesis and discusses future directions building upon the works discussed.

In summary, by exploring ensemble deep learning strategies, single-cell data simulation and patient-focused analytical strategies, this thesis demonstrates how these approaches can be used to overcome specific technical challenges in using single-cell data in precision medicine research. The publicly available frameworks and methods developed in this thesis will not only aid the current research community but will also aid in aspiring future methodologies.

2

ENSEMBLE DEEP LEARNING IN BIOINFORMATICS

Single-cell sequencing technology enables the profiling of individual cells and leads to a plethora of novel applications for precision medicine. However, the characteristics of extreme sparsity and ultra-high resolution of single-cell data bring data analysis challenges in terms of model stability and scalability and hinder the effective utilisation of single-cell data for precision medicine.

In this chapter, we examine the recent emergence of ensemble deep learning frameworks in the broader bioinformatics field, where synergistic improvements in model scalability, stability and accuracy are achieved through combining the two learning techniques of ensemble learning and deep learning. We survey recent key developments in ensemble deep learning and how their contributions have benefited a wide range of bioinformatics research from basic sequence analysis to systems biology, including disease analysis. The result has been published as a review article as Cao *et al.* (2020). To the best of our knowledge, this is the first review article on the topic of ensemble deep learning for bioinformatics applications.

While the application of ensemble deep learning in bioinformatics is diverse and multifaceted, this chapter identifies and discusses the common challenges and opportunities in the context of bioinformatics research. We hope this work will bring together the broader community of machine learning researchers, bioinformaticians, and biologists to foster future research and development in ensemble deep learning and inspire novel precision medicine applications using single-cell data that achieves both model stability and model scalability.

2.1 INTRODUCTION

Bioinformatics, an interdisciplinary field of research, is at the centre of modern molecular biology where computational methods are developed and utilised to transform biological data into knowledge and translate them for biomedical applications. Among the various computational methods utilised in bioinformatics research, machine learning, a branch of artificial intelligence characterised by data-driven model building, has been the key enabling computational technology (Larranaga *et al.*, 2006). At the forefront of machine learning, ensemble learning and deep learning have *independently* made a significant impact on the field of bioinformatics through their widespread applications from basic nucleotide and protein sequence analysis to systems biology (Eraslan *et al.*, 2019; Camacho *et al.*, 2018).

Until recently, ensemble and deep learning models have largely been treated as independent methodologies in bioinformatics applications. The fast-growing synergy between these two popular techniques, however, has attracted a new wave of development and application of next-generation machine learning methods referred to as *ensemble deep learning* (Figure 2.1a). The root of ensemble deep learning can be traced back two decades where ensembles of neural networks were found to reduce generalisation error (Hansen and Salamon, 1990). However, the recent resurgence of ensemble of deep learning models has brought about new ideas, algorithms, frameworks, and architectures that significantly enrich the old paradigm. Through its novel application to a wide range of biological and biomedical research, ensemble deep learning is unleashing its power in dealing with key challenges including small sample size, high- dimensionality, imbalanced class distribution, and noisy and heterogeneous data generated from diverse cellular and biological systems using an array of high-throughput omics technologies. These computational, methodological and technological undertakings and breakthroughs together are leading a phenomenal transformation of bioinformatics.

Both ensemble learning and deep learning methods have been extensively studied and reviewed in the context of bioinformatics applications (Yang *et al.*, 2010b; Min *et al.*, 2017). However, the emergence of ensemble deep learning and its application in bioinformatics has yet to be documented. With the aim of providing a reference point to foster research in the increasingly popular field of ensemble deep learning and its application to various challenges in bioinformatics, in this review, we revisit the foundation of ensemble and deep learning and summarise and categorise the latest developments in ensemble deep learning. This is followed by a survey of ensemble deep learning applications in bioinformatics. We then discuss the remaining challenges and opportunities which we hope will inspire future research and development across multiple disciplines.

2.2 BASICS OF ENSEMBLE AND DEEP LEARNING

Ensemble learning refers to a class of strategies where instead of building a single model, multiple 'base' models are combined to perform tasks such as supervised and unsupervised learning (Dietterich, 2000). Classic ensemble methods for supervised learning fall into three categories including bagging-, boosting- and stacking-based methods. In bagging (Breiman, 1996), individual base models are trained on subsets of data sampled randomly with replacement (Figure 2.1b). In boosting (Schapire *et al.*, 1998), models are trained sequentially (Figure 2.1c), where subsequent models focus on previous misclassified samples. In stacking, a meta-learner is trained to optimally combine the predictions made by base models (Wolpert, 1992). Like supervised ensemble learning, conventional unsupervised ensemble learning such as ensemble clustering (Vega-Pons and Ruiz-Shulcloper, 2011) also relies on the generation and integration of base models (Figure 2.1d). While their variants, including more advanced methods reviewed in the next section, have also been used in ensemble learning, a guiding principle in designing ensemble methods has been 'many heads are better than one' (Altman and Krzywinski, 2017).

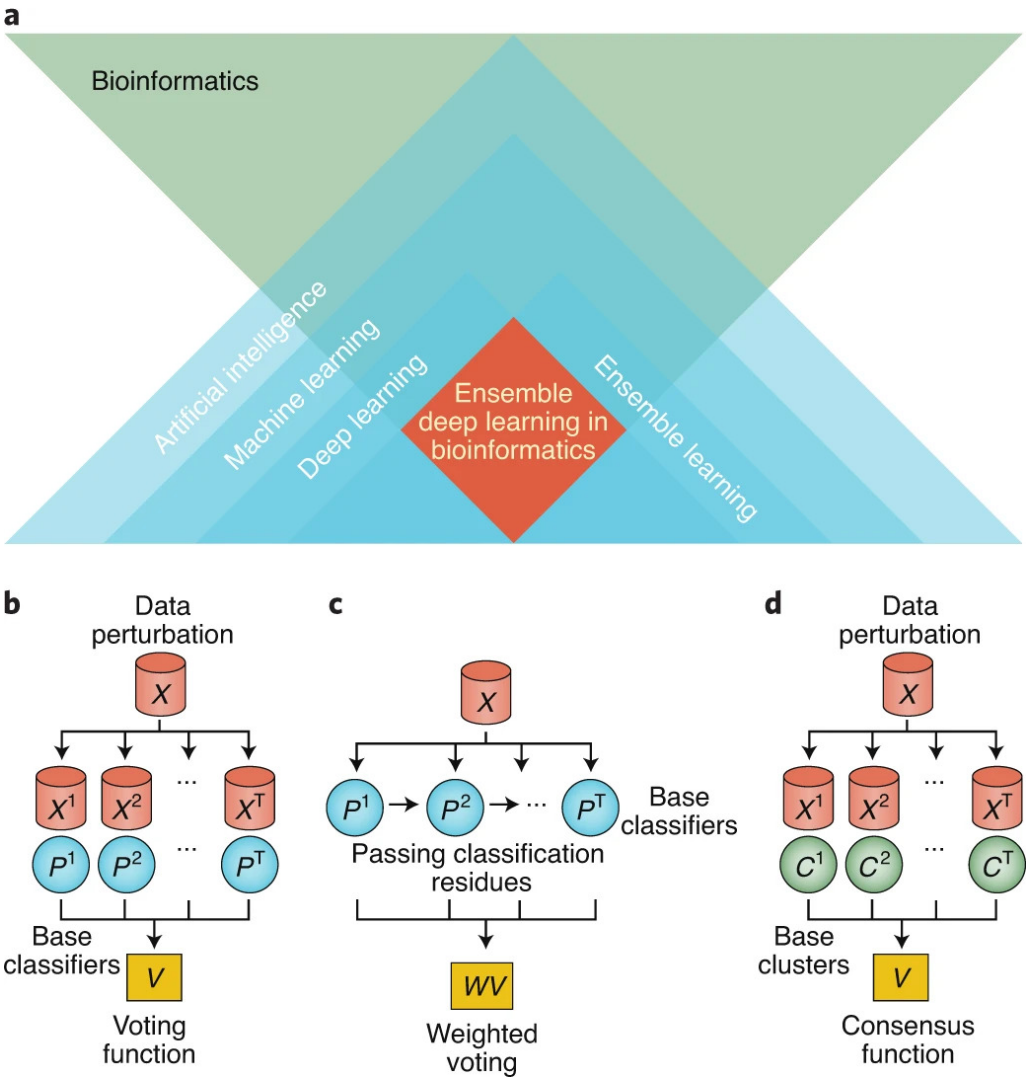


Figure 2.1: The focus of this review and classic ensemble methods. (a) Relationships of artificial intelligence, machine learning, deep learning, ensemble learning, and bioinformatics. The red square denotes the focal point of this review. Classic ensemble learning frameworks including (b) bagging and its variants; (c) boosting and its variants; and (d) ensemble clustering based on data perturbation. For all panels, X represents the data input, either original data or perturbed data, P represents the probability of classification outcome and C represents the clustering outcome.

Deep learning, a branch of machine learning, is rooted in artificial neural networks (ANNs) (Schmidhuber, 2015). The most fundamental architecture of deep learning models is the densely-connected neural network (DNN), consisting of a series of layers of neurons; each of these is connected to all neurons in

the previous layer (Rumelhart *et al.*, 1986). More sophisticated models extend on the basic architectures. In convolutional neural networks (CNNs) (Krizhevsky *et al.*, 2012), each layer comprises a series of filters which 'slide over' the output of the previous layer to extract local features across different parts of the input. In recurrent neural networks (RNNs) (Williams and Zipser, 1989), circuits are created to feed the output of a layer back into the same layer along with new input, allowing the model to act on dependencies between up- and downstream values in a sequence. Variants of RNNs have been proposed to enable more effective learning in long-term dependency tasks, with the two most common ones being long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho *et al.*, 2014). In residual neural networks (ResNet) (He *et al.*, 2016), shortcuts between upstream and downstream layers are introduced to improve the effectiveness of backpropagation in networks with many hidden layers. In autoencoders (Baldi, 2012), networks are constructed with an encoder and a decoder which together learn a more efficient latent space representation of the original higher-dimensional data. Although the difference between traditional neural networks and deep learning may seem elusive, the latter is increasingly defined by their unique architectures and ability to learn complex data representations that are beyond the capacity of classic models (LeCun *et al.*, 2015).

2.3 ENSEMBLE DEEP LEARNING: THE SYNERGY

Deep learning is well known for its power to approximate almost any function and increasingly demonstrates predictive accuracy that surpasses human experts. However, deep learning models are not without shortcomings: they often exhibit high variance and may fall into local loss minima during training. Indeed, empirical results of ensemble methods that combine output of multiple deep learning models have shown to achieve better generalisability than a single model (Ju *et al.*, 2018). In addition to simple ensemble approaches such as averaging output from individual models, combining heterogeneous models enables multifaceted abstraction of data and may lead to better learning

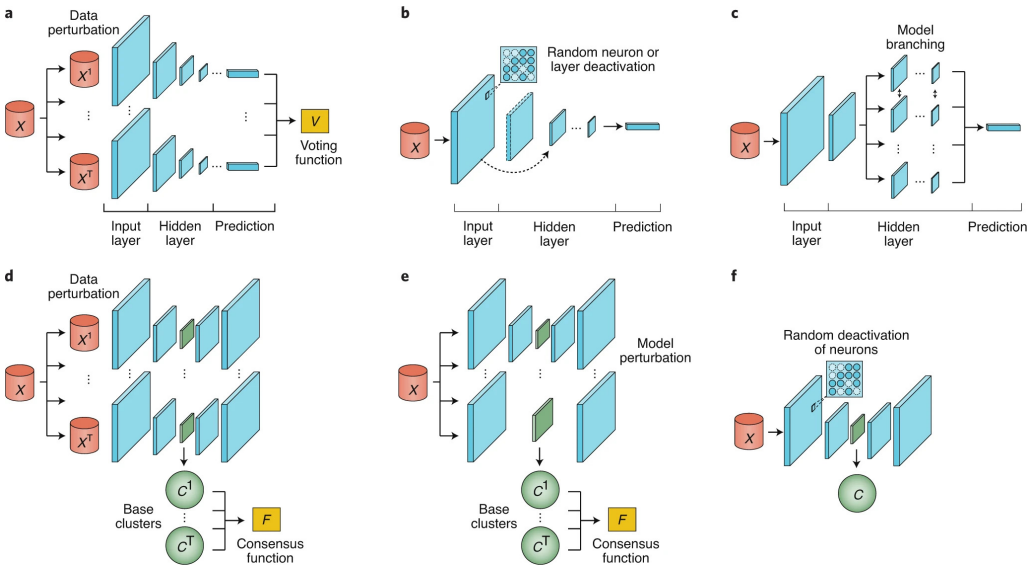


Figure 2.2: Typical ensemble deep learning frameworks in supervised and unsupervised learning. (a) Ensemble across multiple models. Each neural network is trained separately on the dataset, usually perturbed to allow the network to learn from diverse training samples. (b) Ensemble within a single model. Common strategies for creating intrinsic variants of the network include randomly deactivating and bypassing layers (indicated by the curved arrow) and randomly deactivating neurons (indicated by the close-up). (c) Ensemble by model branching. Common strategies include sharing lower layers and branching out to learn different higher level features with or without weight sharing. (d) Unsupervised ensemble by data perturbation. Each autoencoder is trained with a perturbed dataset such as bootstrapping. The latent representations are extracted for clustering and combined through a consensus function. (e) Model perturbation based unsupervised ensemble. Multiple autoencoders each with a different model architecture can be used to learn diverse representation of the original data. (f) Unsupervised ensemble within a single model. Similar to the supervised case, random deactivation of neurons can be used to create intrinsic variants of the network. For all panels, X represents the data input and C represents the clustering outcome.

outcomes (Lee *et al.*, 2015). In this section, we categorise and summarise the most representative ensemble deep learning strategies for both supervised and unsupervised tasks.

2.3.1 *Supervised ensemble deep learning*

2.3.1.1 *Ensemble across multiple models*

The aggregation of multiple and often independent deep learning models is the most straightforward application of ensemble deep learning to classification (Figure 2.2a). As diversity of individual networks is an essential characteristic of a good ensemble model (Granitto *et al.*, 2005), a variety of strategies exist to promote diversity of base networks. One approach is to encourage negative correlation in the classification error of base models (Liu and Yao, 1999). The key motivation behind promoting negative correlation among base models is to encourage complementary learning of the training data to achieve better generalisability of the ensemble. An alternative approach to increase base model diversity is through multiple choice learning in which each network is 'specialised' on a particular subset of data during the training step (Lee *et al.*, 2016).

An issue associated with training and storing multiple models is the computational and storage demand involved. To address this, methods that perform knowledge distillation have become increasingly popular (Hinton *et al.*, 2015). One such implementation is based on the concept of a teacher-student network framework where the teacher networks are selected from a pool of pre-trained networks and the student network distills knowledge of multiple teachers into a single and often simpler network (Shen *et al.*, 2019; Parisotto *et al.*, 2016). The testing phase is storage and computationally efficient, as the samples only need to pass through a single student network.

2.3.1.2 *Ensemble within a single model*

Ensemble strategies described above require training of multiple models. Deep learning models are often computationally costly to train and may take days or even weeks depending on the scale of the dataset and model. Effort has been made to develop 'implicit ensembles' where a single neural network could achieve an effect similar to integrating multiple network models. To this end, a

group of techniques focuses on random deactivation of neurons and layers during the training process of a single model. This leads to an implicit ensemble of networks with different architectures (Figure 2.2b). For example, the random deactivation of neurons, termed *dropout*, originally proposed as a regularisation strategy (Srivastava *et al.*, 2014) for addressing model overfitting is now widely known as an implicit ensemble strategy (Baldi and Sadowski, 2013; Hara *et al.*, 2016). This has inspired followup works on random deactivation of ResBlocks in ResNets (Huang *et al.*, 2016) and the combined random deactivation of neurons and layers (Singh *et al.*, 2016). Besides random deactivation-based methods, alternative strategies have also been explored. One popular approach is the Snapshot ensemble technique, where the key idea is to save multiple versions of a single model during the training process for forming an ensemble (Huang *et al.*, 2017). In a Snapshot ensemble, a cyclic learning rate scheduler is utilised where the learning rate is abruptly changed every few epochs to perturb the network and thus may lead to diversity in the snapshots of the model.

2.3.1.3 *Ensemble with model branching*

Single-model ensemble approaches greatly reduce training cost compared to ensembles of multiple models. However, such a reduction in computational demand comes potentially at a cost in base model diversity. Since the information captured by the lower layers of neural networks is likely to be similar across models, a group of techniques has emerged with a focus on sharing lower layers followed by 'branching' of additional layers (Han *et al.*, 2017). These model branching approaches introduce diversity while also enjoying the reduction of time and computation of training multiple models (Figure 2.2c). Besides reducing computational cost, model branching has also been adapted to address other challenges in training an ensemble. For example, gradient can be propagated over a shorter path in a branching network, mitigating the vanishing gradient problem (Wang *et al.*, 2018). In the knowledge distillation framework, each branch acts as a student model, ensembled to form a teacher model on the fly to reduce the computationally intensive process of pre-training the teacher model (Zhu *et al.*, 2018). The key commonality between these model branch-

ing network ensembles is that by sharing information, the base networks avoid parameter search from scratch and can converge faster.

2.3.2 *Unsupervised ensemble deep learning*

2.3.2.1 *Ensemble across multiple models*

Most unsupervised ensemble deep learning methods employ autoencoders, a popular unsupervised network architecture. Similar to supervised approach, unsupervised ensemble methods can be categorised into those that generate and combine multiple models through data and model perturbation and those that achieve implicit ensemble within a single model.

For methods based on data perturbation, strategies akin to bagging in supervised learning are widely used (Figure 2.2d). For example, Geddes *et al.* used random feature projection of the input data to train a set of autoencoders to create a cluster ensemble (Geddes *et al.*, 2019). Training a series of unsupervised networks with different hyperparameters is a common ensemble strategy for methods based on model perturbation (Figure 2.2e). An example extending this approach is to use different activation functions and a weighting scheme to improve model accuracy Shao *et al.* (2018). An alternative to data and model perturbation is to use multi-view clustering when such data are available. Representative examples include multi-view representation learning using deep canonically correlated autoencoders (Wang *et al.*, 2015) and multi-view spectral clustering where multiple embedding networks were used to represent the original data from different feature sets (Huang *et al.*, 2019).

2.3.2.2 *Ensemble within a single model*

The power of autoencoders in data dimension reduction has motivated research around creating better data representations that are robust to noise in the input data. For example, a denoising autoencoder architecture was introduced in Vincent *et al.* (2008), where values of a random subset of neurons are masked (i.e. changed to zero) during each training epoch, forcing the network to over-

come noise introduced to the data. The concept of randomly masking neurons in denoising autoencoders is an analogue to the dropout method used in the supervised approach, and hence can be considered as an implicit ensemble within a single model, or 'pseudo-ensemble' (Bachman *et al.*, 2014), for unsupervised deep learning (Figure 2.2f). In this line of research, a recent study exploits the flexibility of the dropout algorithm and embeds it in a more advanced variational autoencoder architecture (Antelmi *et al.*, 2019). The proposed algorithm employs a novel strategy to learn the dropout parameter, thus alleviating the need for manual tuning. Another extension in this direction is the 'stacked' denoising autoencoders that uses multiple layers of denoising autoencoders for improving data representation (Vincent *et al.*, 2010). The data representation learned from such 'stacked' denoising autoencoders led to significantly improved classification accuracy than using raw input data.

2.3.3 *Theoretical advances for ensemble deep learning*

While early works on the bias-variance trade-off framework have laid the theoretical foundation for neural network ensembles (Geman *et al.*, 1992), recent research on ensemble deep learning mostly relies on empirical experiments due to the increasingly specialised ensemble methodologies and complex neural network architectures. Nevertheless, efforts have been made to advance the theoretical foundation of this fast-growing field (Bengio, 2009). Studies have shown the existence of multiple local minima in training neural networks, where some enjoy better generalisability than others (Keskar *et al.*, 2017). This has inspired ensemble techniques such as Snapshot methods that take advantage of the diversity of multiple local minimums (Huang *et al.*, 2017). Theoretical justification for dropout as a form of averaging has been discussed in Baldi and Sadowski (2013), where the expectation of the gradient with dropout was shown to be the gradient of the regularised ensemble error. A recent mathematical framework provided a new perspective of dropout by relating it to a form of data augmentation (Zhao *et al.*, 2019).

2.4 BIOINFORMATICS APPLICATIONS OF ENSEMBLE DEEP LEARNING

This section categorises representative works in different areas of bioinformatics applications (Table 1) and identifies their benefits such as improving model accuracy, reproducibility, interpretability, and model inference.

Table 1: Categorisation of recent ensemble deep learning methods in bioinformatics application.

Type of learning	Ensemble technique	Deep learning architecture	Sequence analysis	Genome analysis	Gene expression	Structural bio-informatics	Proteomics	Systems biology	Multi- omics	Bioimage informatics
Supervised	Multiple models	DNN			Grewal *et al.* (2019); Xiao *et al.* (2018); West *et al.* (2018)		(Demichev *et al.*, 2020)	Zhang *et al.* (2019a)	(Sharifi-Noghabi *et al.*, 2019)	Yuan *et al.* (2018)
		CNN	Zhang *et al.* (2018b)	Hu *et al.* (2018)		Zacharaki (2017)		Hu *et al.* (2019b); Hu *et al.* (2019a)		
		CNN + RNN	(Bartoszewicz *et al.*, 2020); Zhang *et al.* (2017)	(Angermueller *et al.*, 2017)		Li and Yu (2016); Torrisi *et al.* (2019)	Zohora *et al.* (2019)	Karimi *et al.* (2019)	(Arefeen *et al.*, 2019)	
		CNN + RNN + ResNet	He *et al.* (2019)			Singh *et al.* (2019); Zhang *et al.* (2018a); (Singh *et al.*, 2018)				
		Others	Cao *et al.* (2018)	Karim *et al.* (2019)						Codella *et al.* (2017)
	Within single model	CNN + RNN		Karim *et al.* (2019)						
	Model branching	CNN								(Song *et al.*, 2015); Rasti *et al.* (2017)
		CNN + ResNet								(Lu *et al.*, 2020)
Unsupervised	Multiple models	Autoencoder			Geddes *et al.* (2019); Tan *et al.* (2017)				(Gala *et al.*, 2019); (Zhang *et al.*, 2019b)	
		Others							(Liang *et al.*, 2014)	
	Within single model	Autoencoder		Karim *et al.* (2019)						

2.4.1 *Sequence analysis*

Biological sequence analysis represents one of the fundamental applications of computational methods in molecular biology. RNN and its variants (e.g. LSTM

and GRU) are well-suited to sequential data. For example, an LSTM/CNN multi-model was trained to extract distinct features to predict *pathogenic potential* of DNA sequences (Bartoszewicz *et al.*, 2020). Compared to DNA sequences, RNA sequences offer an additional layer of information where instructions encoded in genes are transcribed. While traditional methods rely on various manually curated RNA sequence features, ensemble deep learning enables automatic learning from raw data. One example is in predicting *localisation of long non-coding RNAs*, where multiple sub-networks were used to integrate distinct feature sets to maximise model performance (Cao *et al.*, 2018). In another work, a CNN/RNN ensemble was used to integrate features and raw sequence data to predict different types of *translation initiation sites* (Zhang *et al.*, 2017), overcoming the generalisability issue of traditional methods that can only predict a specific type of translational initiation sites.

Following transcription, messenger RNAs (mRNAs) are further translated into proteins that carry out various functions. Similar to RNA sequence analysis, methods relying on ensembles of multiple sub-networks were used to integrate information from multiple features sets to predict *DNA binding sites* (Zhang *et al.*, 2018b) and *post-translational modification (PTM) sites* He *et al.* (2019) on protein sequences. The study on PTM site prediction has further demonstrated that features learned by ensemble models are 'transferable' for predicting different types of PTMs, a key property for tackling the issue of small sample size in training data.

2.4.2 *Genome analysis*

Whilst sequence analysis has led to many biological discoveries, it alone cannot capture the full repertoire of information encoded in the genome. Additional layers of genetic information including structural variants (Feuk *et al.*, 2006) (e.g. copy number variations [CNVs]) and epigenetic modifications (Portela and Esteller, 2010) of the genome bring important insight to the understanding of biological systems, populations, and complex diseases.

A number of ensemble deep learning methods have been developed on this front, such as classifying *cancer types* using CNV data and a Snapshot ensemble model comprising CNNs, LSTMs, and convolutional autoencoders (Karim *et al.*, 2019). The use of supervised CNN and LSTM models allows both global and local sequential features to be captured, and further integration with unsupervised convolutional autoencoders enables unsupervised pre-training, an effective component for handling small sample size (Erhan *et al.*, 2010). Beyond combining different network architectures, studies have also integrated different genomic data modalities to capture distinct and complementary information. In one study, DNA sequences and their neighbouring CpG states were used as input into two sub-networks of an ensemble to explore their relationship in predicting *DNA methylation states* (Angermueller *et al.*, 2017). This has led to the identification of sequence motifs related to DNA methylation and the effect of their mutation on CpG methylation. In another study, an ensemble network that takes input data either from DNA sequences alone or with the addition of epigenetic information extracted from chromatin immunoprecipitation (ChIP) and deoxyribonuclease (DNase) sequencing were used to predict *human immunodeficiency virus type 1 (HIV-1) integration sites* (Hu *et al.*, 2018). The ensemble network, comprised of CNNs with attention layers (Bahdanau *et al.*, 2014), enabled the discovery of DNA sequence motifs that are important for HIV-1 integration.

2.4.3 *Gene expression*

Gene expression data including microarray, RNA-sequencing (RNA-seq) and, recently, single-cell RNA-seq (scRNA-seq) (Yang and Speed, 2002; Ozsolak and Milos, 2011; Kolodziejczyk *et al.*, 2015a), has been studied extensively to better understand complex diseases and to identify biomarkers that can guide therapeutic decision-making. A recent study on *cancer type classification* demonstrated how ensemble deep learning can serve as a potential strategy to address the key challenge of reproducibility in biomarker research (Grewal *et al.*, 2019). The use of a DNN ensemble in this work allowed the derivation of important

genes through consensus ranking across multiple models, resulting in a robust set of biomarkers. Due to the difficulty of obtaining patient samples, especially for rare diseases and cancer types, another common challenge in analysing gene expression data from cancers and diseases is the small sample size. The use of ensemble learning to mitigate this issue is exemplified by Xiao *et al.* (2018), where the authors applied a multi-model approach to generate initial predictions from RNA-seq gene expression profiles of cancer samples and integrated these predictions using a DNN to produce the final ensemble prediction.

In addition to its role in medical research, ensemble deep learning has been used in a wide range of applications to improve *understanding of basic biological mechanisms* from gene expression data. An example is the use of a DNN ensemble to explore the embryonic to fetal transition process, a defining stage where cells lose the potential for regeneration (West *et al.*, 2018). A benefit of training multiple networks is that the prediction scores from each network can be further used to generate an integrative score to determine the transition state of a sample between embryonic and adult state, a strategy that is not possible with a single model. The utility of unsupervised ensemble deep learning has also been demonstrated on the extraction of *biological pathway signatures* (Tan *et al.*, 2017). By integrating signatures across 100 autoencoders through consensus clustering, the ensemble model detected more biological pathways with higher significance than did a single model. Unsupervised deep learning ensembles have also been applied to *cell type identification* in single cell research. In (Geddes *et al.*, 2019), an ensemble of autoencoders was used to generate a diverse set of latent representations of scRNA-seq data for subsequent analysis.

2.4.4 *Structural bioinformatics*

Proteins are the key products of genes and their functions and mechanisms are largely governed by protein structures encoded in amino acid sequences. Therefore, modelling and characterising proteins from their primary amino acid sequences to secondary and tertiary structures is essential for understand-

ing and predicting their functions (Lee *et al.*, 2007). RNN and its architectural variants are specifically designed to capture long- and short-range interactions between sequences, and are hence well-suited to decoding the relationship between amino acid sequences and the protein structures they encode. Extending on the use of a single RNN, the ensemble of variants of RNNs with CNNs is a common hybrid architecture in recent applications that seeks to combine the power of RNN in analysing sequential data and CNN on extracting local features (Li and Yu, 2016; Torrisi *et al.*, 2019). The replacement of CNN with ResNet (Singh *et al.*, 2019) as well as the addition of residual connections between GRU and CNN (Zhang *et al.*, 2018a) were also explored to facilitate feature propagation for improved modelling of long-range dependencies between amino acids. In these works, ensemble deep learning not only improved generalisability on independent datasets but also led to the discovery of novel features associated with protein structures.

Besides predicting protein structures, many studies have focused on directly *predicting protein functions*. An example of ensemble deep learning application in this domain is illustrated by the work of Zacharaki Zacharaki (2017), who used an ensemble of CNNs for protein enzymatic function prediction. Specifically, the ensemble is a fusion of two CNNs trained separately on protein properties and amino acid features for extracting complementary information. In another example, Singh *et al.* Singh *et al.* (2018) built an ensemble deep learning model to identify residue conformation crucial to protein folding and function. While the dataset used for model training has an extreme class imbalance (1.4:1000), the ensemble model, consisting of ResNet and LSTM modules, yielded robust performance on independent test sets without manual generation of a balanced dataset.

2.4.5 *Proteomics*

While protein structure and function prediction are essential tasks for characterising individual proteins, technological advances in quantitative mass spec-

trometry (MS) have now enabled global profiling of the entire proteome in cells, tissues, and species (Walther and Mann, 2010). Computational analysis of such large volume datasets is transforming our understanding of proteome dynamics in complex systems and diseases (Cox and Mann, 2011).

Ensemble deep learning has been used as a key technique for addressing various aspects of proteomics data analysis. The work of Zohora *et al.* Zohora *et al.* (2019) exemplifies the application of ensemble deep learning to *peptide identification* from liquid chromatography-MS (LC-MS) map, a critical step for identifying and quantifying protein abundance. Specifically, a hybrid network architecture comprising both CNN and RNN modules was designed to detect sequential features along the axes during the scan of an MS map. The final model, an ensemble of multiple networks with different parameters, was shown to achieve state-of-the-art results for protein quantification. Another study proposed an ensemble of DNNs for learning from data-independent acquisition (DIA) MS data (Demichev *et al.*, 2020). Whilst conventional MS runs select only a few significant peptides based on their signal levels (i.e. data-dependent acquisition [DDA]) for subsequent quantification, the DIA approach fragments every single peptide for improved proteome coverage. However, the DIA approach may lead to an increase in co-eluted peptides and therefore higher interference in the data. The ensemble framework was able to quantify the amount of interference between multiple peptides mapped to the same point, thereby removing interference and improving peptide identification confidence and quantification accuracy.

2.4.6 *Systems biology*

Systems biology aims to map interactions of molecule species, regulatory relationships and mechanisms to understand complex biological systems as a whole (Kitano, 2002). One key aspect of systems biology is the understanding of what and how biological molecules interact. In recent times, ensemble deep learning has been applied on this front to *predict interactions* among different

biological molecules and entities. The application of an interpretable ensemble of CNN models for predicting binding affinity between peptides and major histocompatibility complex (MHC) is an example of ensemble deep learning in this domain Hu *et al.* (2019b) and has significant implication in clinics. The model demonstrated good generalisability across 30 independent datasets and uncovered binding motifs with literature support. In predicting protein-protein interactions, an ensemble of DNNs trained on *S. cerevisiae* achieved more accurate results than other machine learning methods (Zhang *et al.*, 2019a). Subsequently, the model was applied to other datasets generated from different organisms and the relative accuracy on each dataset was shown to be a good indicator of the evolutionary relationships of those organisms.

Systems biology also extends to the interaction between biological molecules and chemical compounds. In particular, the study of protein and chemical compound interaction in *drug development* has seen a growing number of ensemble deep learning applications. For example, Karimi *et al.* proposed an ensemble model that comprised various network modules for compound-protein affinity prediction (Karimi *et al.*, 2019). To overcome the limited availability of labelled datasets, the model exploited abundant unlabelled compound and protein data through unsupervised pre-training. This was followed by interaction prediction on labelled data using CNN and RNN modules in the ensemble. In another work on predicting drug and target protein interactions, a CNN-based ensemble model was used to score the likelihood of interaction of randomly selected drug-protein pairs Hu *et al.* (2019a). The trained model revealed that drugs with similar structures bind to similar target proteins, suggesting potential similarity in the effects of these drugs.

2.4.7 *Multi-omics*

Multi-omics analysis is a topic closely related to systems biology where integrative methods are used to understand biological regulation by combining an array of omics data. There is a growing interest in multi-omic studies as it is

increasingly recognised that a single type of omics data does not capture the entire landscape of the complex biological networks Yang *et al.* (2019b).

Many conventional machine learning methods have been proposed to utilise the complementary information present across multiple modalities of omics data (Kim *et al.*, 2020; Ramazzotti *et al.*, 2018). Most conventional approaches, however, do not account for the relationships among different omics layers. To this end, Liang *et al.* proposed to use an ensemble of deep belief networks to encode gene expression, miRNA expression, and DNA methylation data into multiple layers of hidden variables for integrative clustering (Liang *et al.*, 2014), thereby actively exploring regulation across different omics layers. Ensembles of different deep learning architectures have also been utilised to take advantage of the unique characteristics of different data types. Using an ensemble of CNNs and LSTMs, both genomic sequences and their secondary structures can now be integrated for alternative polyadenylation site prediction on pre-mRNAs (Arefeen *et al.*, 2019). This addressed the gap where existing models overlooked RNA secondary structures, despite these being important features to the polyadenylation process. Another application in multi-omics was the use of a novel ensemble of autoencoders wherein a coupling cost was used to encourage the base autoencoders to learn from each other (Gala *et al.*, 2019). This unsupervised model allowed the integration of two vastly different data types—single cell transcriptomics and electrophysiological profiles, and to identify common and unique cell types across datasets.

High dimensionality and heterogeneity are both issues associated with the large number of molecular features in multi-omics datasets. The application of autoencoders is popular in dealing with these challenges. In one instance, an ensemble of autoencoders was used to extract lower dimension and integrate over 450,000 features in pan-cancer classification (Zhang *et al.*, 2019b). Stacking multiple deep learning models, each handling a different modality of omics data (Sharifi-Noghabi *et al.*, 2019), is another approach that avoids feature concatenation which might otherwise exacerbate the issue of high dimensionality in datasets potentially containing tens of thousands of features.

2.4.8 *Bioimage informatics*

Traditionally, analysis of bioimages is often performed manually by field experts. With the growing number of computer vision applications demonstrating their superior performance over human experts, automatic analysis has become an increasing focus in bioinformatics studies.

A primary application of ensemble deep learning in bioimage informatics is the *detection of diseases such as cancers* in patient images. For instance, to improve classification of glioma from magnetic resonance images, Lu *et al.* embedded a branching module into ResNet for integrating multi-scale information obtained from different receptive fields of the original ResNet (Lu *et al.*, 2020). Codella *et al.* proposed an ensemble model that combined network architectures including ResNet, CNN and U-Net, to segment and classify skin lesions from dermoscopic images (Codella *et al.*, 2017). It is noteworthy that the proposed model achieved a segmentation result with 95% accuracy, surpassing that of human experts who exhibit an accuracy of around 91%. To segment cervical cell images, Song *et al.* performed multi-resolution extraction and colour space transformation of the images to generate diverse feature sets, leading to enhanced segmentation accuracy (Song *et al.*, 2015).

Besides improving classification and segmentation accuracy, ensemble deep learning methods have also been explored in addressing various other challenges in bioimage analysis. For example, an ensemble network with knowledge distillation and a branching strategy was used to reduce the number of parameters in the model and therefore lower the likelihood of overfitting on small datasets (Rasti *et al.*, 2017). To deal with the problem of class imbalance, Yuan *et al.* (2018) introduced an iterative regularisation approach which, for a given iteration, penalises misclassification of samples that were correctly classified in previous iterations. This method alleviated the problem of bias in favour of majority classes and preserved correctly classified minority examples.

2.5 CHALLENGES AND OPPORTUNITIES

The applications reviewed above reveal various challenges and opportunities surrounding ensemble deep learning in bioinformatics research. As the focus of this thesis is on the analysis of next-generation omics sequencing data, particularly single-cell data, below we highlight several key areas in which ensemble deep learning is likely to have an increasingly significant impact in omics data research.

2.5.1 *Small sample size*

Deep learning is known for its exceptional performance on data with large sample size. While modern omics technologies have enabled the profiling of tens of thousands of molecular species and biological events in a single experiment, the number of samples available is usually small owing to the cost in time and labour. Hence, bioinformatics applications are often confronted with the issue of limited sample size, causing unstable predictions and thus low reproducibility in results.

Fortunately, one essential property of ensemble methods is stability. Leveraging this key property, a number of ensemble deep learning methods were proposed to specifically address small sample size challenges, opening up the opportunity to utilise deep learning in bioinformatics. While the most popular approach so far has been using pre-trained models, more specialised methods have also been explored. Examples include extracting intermediate features learned by the network to generate additional output for integration and thus stabilising the ensemble prediction (Xie *et al.*, 2013); and encouraging cooperation among individual models through a pairwise loss, thereby reducing the variance caused by small sample size (Dvornik *et al.*, 2019). These methods represent promising strategies that can be explored in future lines of research.

2.5.2 *High-dimensionality and class imbalance*

Omics data are well-known for their high-dimensionality, as biological features (e.g. genes, proteins) frequently outnumber samples. This is further exacerbated by the issue of small sample size mentioned above. The problem, widely known as the 'curse of dimensionality', has been identified as one of the main causes of overfitting in deep learning models due to the large number of parameters that needs to be fitted (Bzdok *et al.*, 2019). While deep learning models seem to be particularly susceptible to the high-dimensionality of omics data, the combination of deep learning with ensemble methods such as model averaging (Geddes *et al.*, 2019) and the implicit ensemble through dropout (Srivastava *et al.*, 2014) has been demonstrated to be an effective approach for handling this issue.

Imbalanced class distribution is another common issue in many bioinformatics applications (Yang *et al.*, 2014) where, for example, a biological event of interest is only present in a small proportion of the data. Ensemble deep learning is found to be an effective remedy for dealing with this challenge. Bioinformatics applications reviewed include the use of bootstrap sampling– and random sampling–based ensemble deep learning for dealing with class imbalance in DNA and protein sequence analyses (Zhang *et al.*, 2017, 2018b). Due to the increasing use of high-throughput technologies, ensemble deep learning strategies that are capable of dealing with these challenges will remain an active research direction in bioinformatics.

2.5.3 *Data noise and heterogeneity*

Biological systems are inherently heterogeneous and noisy. This is further confounded by technical noise from various sources including experimental protocol and omics platform. A key characteristic of ensemble methods is their robustness to data noise (Yang *et al.*, 2019a), which can facilitate the reproducible extraction of biological signals from noisy and heterogeneous data. The application of methods such as denoising autoencoders also strengthens model

robustness (Vincent *et al.*, 2008). The integration of ensemble and deep learning methods therefore provides an opportunity to address noise and heterogeneity in biological data.

The development of multi-omics technologies further contributed to heterogeneity within datasets in that different molecular species measured across omics platforms must be combined and analysed integratively to understand biological systems holistically. Ensemble deep learning methods such as multi-model approaches reviewed previously have been demonstrated to be highly effective in combining different omics data for joint inference (Liang *et al.*, 2014) and classification (Arefeen *et al.*, 2019). Given these intrinsic properties of data generated from biological systems, we expect ensemble deep learning methods to play an increasingly important role in omics data analysis and in integrating large-scale multi-omics data.

2.5.4 *Model interpretability*

A common criticism of deep learning models is their lack of interpretability. Besides building accurate model, gaining insight from the model is also critical in bioinformatics applications, since having an interpretable model of a biological system may lead to testable hypotheses that can be validated through experiments.

Several studies reviewed in previous sections have already made notable progress in this direction. For example, attention layers in ensemble networks were used to identify motifs of HIV integration sites (Hu *et al.*, 2018) and drug binding sites (Karimi *et al.*, 2019). The stability and reproducibility offered by ensemble methods such as in feature selection (Abeel *et al.*, 2010) are also making a significant impact in biomarker discovery (Pusztai *et al.*, 2013). This is evident from the application of ensemble deep learning methods to identifying molecular markers for the diagnosis of primary and metastatic cancers (Grewal *et al.*, 2019) and to provide insights into normal development and cancers (West *et al.*,

2018). As we move from predictive to preventive biomedical research, models that offer biological insight into data will become increasingly desirable.

2.5.5 *Choice of network architecture*

The choice of network architecture is crucial for achieving optimal performance in a specific domain and application. For example, many studies choose to employ variants of the RNN such as the LSTM, which is suitable for learning sequential information in biological sequences (Zhang *et al.*, 2017; Torrisi *et al.*, 2019). DNN and CNN architectures, on the other hand, are shown to be suitable for biological applications that handle high-dimensional input (Grewal *et al.*, 2019; Hu *et al.*, 2018).

The use of multi-model ensembles makes it possible to exploit the power of hybrid architectures or to combine heterogeneous data types in multi-omics. Examples reviewed include the ResNet/RNN hybrid used to capture the relationship between each layer of features in RNA secondary structure prediction (Singh *et al.*, 2019) and the CNN/LSTM hybrid to learn both RNA sequences and secondary structures for joint prediction of alternative polyadenylation sites on pre-mRNAs (Arefeen *et al.*, 2019). While these studies demonstrate the importance and the application of specialised network architectures in bioinformatics, the exponential growth of new network architectures proposed in computer science literature will likely lead to many more novel applications in bioinformatics in the coming years.

2.5.6 *Computational expense*

Deep learning models typically contain large numbers of parameters and the computational burden of generating an ensemble of multiple deep learning models could be extremely high especially when working with large-scale omics data. Nevertheless, recent developments in ensemble deep learning have made use of the modularity of deep learning architectures and provided a panel of

ensemble strategies and algorithms to enable more efficient model fitting with a significant reduction in training time. The improvement of computer hardware and technological advances in computing methods such as distributed and federated deep learning (Dean *et al.*, 2012; Smith *et al.*, 2017) also facilitate the application and deployment of ensemble deep learning on large-scale omics data. Given that the size and complexity of biological data are only expected to soar as technology progresses, the development of more efficient ensemble deep learning algorithms and architectures will be another crucial direction in both machine learning and bioinformatics research.

2.6 FUTURE OUTLOOK

While the ensemble of neural networks has existed long before the deep learning era, the recent development of ensemble deep learning has significantly enriched the field with novel architectures and ensemble strategies that greatly improve model accuracy, reliability, and efficiency. These innovations together with properties such as robustness to small sample size, high-dimensionality, and data noise, have transformed ensemble deep learning into a new force leading to remarkable and widespread breakthroughs across different fields of bioinformatics applications. Nonetheless, many of the advanced ensemble techniques that harness the power of recent deep learning architectures remain under-explored in their application to bioinformatics. In addition, the development and application of models that enable interpretation of biological systems are still in their infancy. We hope this review has sparked thoughts on ensemble deep learning across multiple disciplines and will inspire future research and application embracing the myriad of ensemble deep learning strategies to revolutionise biological research and especially single-cell research.

3

A BENCHMARK STUDY OF SIMULATION METHODS FOR SINGLE-CELL RNA SEQUENCING DATA

Simulation data is frequently used to aid methodological development, particularly for analysis of patient disease datasets when the ground truth is unattainable. With the rapid increase in the number of single-cell methods for addressing data analysis challenges, a number of single-cell data simulation tools have been published for meeting the various needs of simulation data. Ensuring the quality of simulation data, particularly their ability to reflect real experimental data is of crucial importance, as this can directly impact the development of computational methods that use simulation data as part of the evaluation measure. However, while numerous scRNA-seq data simulation methods have been proposed, a systematic evaluation of these methods is currently lacking.

In this chapter, we address the above challenge by developing SimBench, a comprehensive evaluation framework for scRNA-seq simulation methods and applying it to benchmark 12 simulation methods using 35 scRNA-seq experimental datasets. The simulation methods are evaluated on multiple aspects of criteria, including the ability to capture a panel of data properties, the ability to maintain biological signals, as well as scalability and applicability. SimBench has been published as Cao *et al.* (2021). To the best of our knowledge, this is the first systematic benchmark study on scRNA-seq simulation methods.

This chapter demonstrates that SimBench uncovers performance differences among the methods and highlights the varying difficulties in simulating data characteristics. These results, together with the framework and datasets made publicly available, will guide simulation methods selection and their future development. Furthermore, the collection of single-cell benchmarking datasets

and the design behind the evaluation framework is not only applicable to the specific task of evaluating single-cell simulation methods, but can also be applied for future evaluation studies.

SimBench is designed as a living benchmark. The SimBench benchmark framework is available as an R package at `https://github.com/SydneyBioX/SimBench`. A Shiny web application for interactively exploring the results is available at `http://shiny.maths.usyd.edu.au/SimBench/`. As the website can be updated beyond the publication of study, new simulation methods can be incorporated when they become available so that our comparative study will stay up-to-date and will support future method development. The availability of a GitHub site also enables the broader community to contribute via creating GitHub pulls.

Following the publication of SimBench, we have since updated the living benchmark to include an additional six scRNA-seq data simulation methods and these can be viewed at the Shiny web application.

3.1 INTRODUCTION

Single-cell RNA-sequencing (scRNA-seq) is a powerful technique for profiling the transcriptomes at the single-cell resolution and has gained considerable popularity since its emergence in the last decade (Kolodziejczyk *et al.*, 2015b). To effectively utilise scRNA-seq data to address biological questions (Luecken and Theis, 2019), the development of computational tools for analysing such data is critical and has grown exponentially with the increasing availability of scRNA-seq datasets. Evaluation of their performance with credible ground truth has thus become a key task for assessing the quality and robustness of the growing array of computational resources. While there exist certain control strategies such as spike-ins with known sequence and quantity, data that offer ground truth while reflecting the complex structures of a variety of experimental designs are either difficult or impossible to generate. Thus, in silico simulation methods for creating scRNA-seq datasets with desired structure and

ground truth (e.g. number of cell groups) are an effective and practical strategy for evaluating computational tools designed for scRNA-seq data analysis.

To date, numerous scRNA-seq data simulation methods have been developed. The majority of these methods employ a two-step process of using statistical models to estimate the characteristics of real experimental single-cell data and using the learnt information as a template to generate simulation data. The distinctive difference between them is the choice of underlying statistical framework. Early methods often employ negative binomial (NB) (Vieth *et al.*, 2017; Zappia *et al.*, 2017; Korthauer *et al.*, 2016) as it has been the typical choice for modelling gene expression count of RNA-seq (Anders and Huber, 2010). Its variant, zero-inflated NB (ZINB) model takes account of excessive zeros in the count data and is chosen by other studies to better model the sparsity in single-cell data (Risso *et al.*, 2018; Van den Berge *et al.*, 2018). In more recent years, alternative models have been proposed with the aim to increase modelling flexibility including gamma-normal mixture model (Li and Li, 2019), beta-Poisson (Zhang *et al.*, 2019c), gamma-multivariate hypergeometric (Baruzzo *et al.*, 2020) and the mixture of zero-inflated Poisson and log-normal Poisson distributions (Su *et al.*, 2020). Other studies argued that parametric models with strong distributional assumption are often not appropriate to scRNA-seq data given its variability and proposed the use of a semi-parametric approach as the simulation framework (Assefa *et al.*, 2020). Similarly, a recent deep learning-based approach (Marouf *et al.*, 2020) leverages the power of neural networks to infer underlying data distribution and avoid prior assumptions.

A common challenge of simulation methods is the ability to generate data that faithfully reflect experimental data (Lähnemann *et al.*, 2020). Given that simulation datasets are widely used for the evaluation and comparison of computational methods (Vieth *et al.*, 2019), deviations of simulated data from properties of experimental data can greatly affect the validity and generalisability of the evaluation results. With the increasing number of scRNA-seq data simulation tools and the reliance on them to guide other method development as well as choosing the most appropriate data analytics strategy, a thorough assessment

of all currently available scRNA-seq simulation methods is crucial and timely, especially when such an evaluation study is still lacking in the literature.

In this chapter, we present a comprehensive evaluation framework, SimBench, for single-cell simulation benchmarking. Considering that realistic simulation datasets are intended to reflect experimental datasets in all data moments including both cell-wise and gene-wise properties, as well as their higher-order interactions, it is important to determine how well simulation methods represent all these values. To this end, we systematically compare the performance of 12 simulation methods across multiple sets of criteria, including accuracy of estimates for 13 data properties, the ability to retain biological signals and to achieve computation scalability, as well as their applicability. To ensure robustness of the results, we collect 35 datasets across a range of sequencing protocols and cell types. Moreover, we implement measure based on kernel density estimation (Duong *et al.*, 2012) in the evaluation framework to enable the large-scale quantification and comparison of similarities between simulated and experimental data across univariate and multivariate distributions, and thus, avoid visual-based criteria which are often used in other studies (Assefa *et al.*, 2020; Li and Li, 2019; Baruzzo *et al.*, 2020). To assist development of new methods, we study potential factors affecting the simulation results and identify common strength and weakness of current simulation methods. Finally, we summarise the result into recommendation to the users, and highlight potential areas requiring future research.

3.2 SIMBENCH FRAMEWORK

3.2.1 *Dataset collection*

A total of 35 publicly available datasets was used for this benchmark study. For all datasets, the cell type labels are either publicly available or obtained from the authors upon request (Chen *et al.*, 2018). Details of each dataset including their accession code are included in the Table A1. The datasets contain a range of

sequencing protocols including both unique molecular identifiers (UMIs) and read-based protocols, multiple tissue types and conditions, and from human and mouse origin.

The raw (unnormalised) count matrix was obtained from each study and quality control was performed by removing potentially low-quality cells or empty droplets that expressed less than one percent of UMIs. For methods that require normalised count, we converted the raw count into log2 counts per million reads (CPM), with addition of pseudocount of 1 to avoid calculating log of zero.

Note the Tabula Muris dataset was only used to benchmark speed and scalability of methods. Properties estimation was evaluated on all other datasets. For evaluating biological signals, 25 datasets containing multiple cell types or conditions as specified by Table A1 were used.

3.2.2 *Selection and implementation of simulation methods*

An extensive literature review was conducted and a total of 12 published single-cell simulation methods with implementation available in R and Python was found. The details of each method, including the version of the code used in this benchmark study and its publication are outlined in Table 1 and Table A2. Table A3 detailed the execution strategy of each method for data property estimation and biological signals and is dependent on the input requirement and the documentation of each method. Where possible, default setting or suggested setting from documentation is followed.

To ensure the simulated data is not simply a memorisation of the original data, we randomly split each dataset into 50% training and 50% testing (referred to as the real data in this study). The training data was used as input to estimate model parameters and generate simulated data. The real data was used as the reference to evaluate the quality of the simulated data, by comparing the similarity between the simulated data and the real data. The same training

and testing subset was used for all methods to avoid the data splitting process being a confounding factor in evaluation.

All methods were executed using a research server with dual Intel(R) Xeon(R) Gold 6148 Processor (40 total cores, 768GB total memory). For methods that support parallel computation, we used 8 cores and stopped the methods if the simulation was not completed within 3h. For methods that run on a single core, we stopped the methods if not completed within 8h.

3.2.3 *Evaluation of data property estimation*

3.2.3.1 *Data properties measured in this study*

We adapted the implementation from countsimQC (v1.6.0) (Soneson and Robinson, 2018b), which is an R package developed to evaluate the similarities between two RNA-seq datasets, either bulk or single-cell and evaluated a total of 13 data properties across univariate and bivariate distribution. They are described in detail below:

- Library size: total counts per cell.
- TMM: weighted trimmed mean of M-values normalisation factor (Robinson and Oshlack, 2010).
- Effective library size: library size multiplied by TMM.
- Scaled variance: z-score standardisation of the variance of gene expression in terms of log2 CPM.
- Mean expression: mean of gene expression in terms of log2 CPM.
- Variance expression: variance of gene expression in terms of log2 CPM.
- Fraction zero cell: fraction of zeros per cell.
- Fraction zero gene: fraction of zeros per gene.
- Cell correlation: Spearman correlation between cells.

- Gene correlation: Spearman correlation between genes.
- Mean vs variance: the relationship between mean and variance of gene expression.
- Mean vs fraction zero: the relationship between mean expression and the proportion of zero per gene.
- Library size vs fraction zero: the relationship between library size and the proportion of zero per gene.

Note that properties relating to library size, including TMM and effective library size can only be calculated using unnormalised count matrix and could not be obtained from methods that generate normalised count. As a result, these scores were shown as a blank space in all relevant figures.

3.2.3.2 *Evaluation measures*

In this study, we used a non-parametric measure termed kernel density based global two-sample comparison test (KDE test) (Duong *et al.*, 2012) to compare the data properties between simulated and real data. The discrepancy between two distributions is calculated based on the difference between the probability density functions, either univariate or multivariate, which are estimated via kernel smoothing.

The null hypothesis of the KDE test is that the two kernel density estimates are the same. An integrated squared error (ISE) serves as the measure of discrepancy and is subsequently used to calculate the final test statistic under the null hypothesis. The ISE is calculated as:

$$T = \int [f_1(x) - f_2(x)]^2 \, dx \tag{1}$$

where f_1 and f_2 are the kernel density estimates of sample 1 and sample 2, respectively. The implementation from the R package ks (v1.10.7) was used for the KDE test performed in this study.

We used the test statistic from the KDE test as the measure to quantify the extent of similarity between simulated and real distributions. We applied a transformation rule by scaling the absolute value of the test statistic to [0,1] and then taking 1 minus the value as shown in the equation below:

$$x_{\text{transformed}} = \frac{|x| - |x_{\text{minimum}}|}{|x_{\text{maximum}}| - |x_{\text{minimum}}|} \tag{2}$$

where x is the raw value before transformation. The transformation is applied on the KDE scores obtained from all methods across all datasets, thus the xminimum and xmaximum are defined based on those values. The purpose of the transformation is to follow the principle of, the higher the value the better and enable easier interpretation.

To assess the validity of the KDE statistic and compare it against other measures, for example, the well-established KS test for univariate distribution, we utilised the measures implemented in countsimQC package. It includes the implementation of the following six measures: Average silhouette width, average local silhouette width, nearest neighbour (NN) rejection fraction, K-S statistics, scaled area between empirical cumulative distribution functions (eCDFs) and Runs statistics. For ease of comparing between the six measures and with the KDE test, we applied transformation rules where applicable such that the outputs from all measures are within the range of 0–1, where value closer to 1 indicates greater similarity. Similarly, the transformation is calculated from all methods across all datasets.

The measures and their transformation rules are:

1. Average silhouette width

For each feature, the Euclidean distances to all other features were calculated. The feature was either gene or cell, depending on the data properties evaluated. A silhouette width $s(i)$ was then calculated using the following formula:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{3}$$

where $b(i)$ is the mean distance between feature i and all other features in the simulation data, $a(i)$ is the mean distance between feature i and all other features in the original dataset.

$s(i)$ of all features is then averaged to obtain the average silhouette width. The range of silhouette width is [-1, 1]. A positive value close to 1 means the data point from the simulation data is similar to the original dataset. Value close to 0 means the data point is close to the decision boundary between the original and simulated. A negative value means the data point from the original dataset is more similar to the simulation data. The same transformation as described in Eq. (2) was applied.

2. Average local silhouette width

Similar to the average local silhouette width. The difference is that instead of calculating the distance with all the features, only the k NNs were used in the calculation. Default setting of k of 5 was used. The same transformation as described in Eq. (2) was applied.

3. NN rejection fraction

First, for each feature the k NNs were found using Euclidean distance. A chi-square test was then performed with the null hypothesis being the composition of k NNs belonging to original and simulation data is similar to the true composition of real and simulation data. The NN rejection fraction was calculated as the fraction of features for which the test was rejected at a significance level of 5%.

The output is the range of [0,1], where a higher value indicates greater dissimilarity between the features from real and simulation data. The value was thus transformed by taking 1 minus the value.

4. Kolmogorov-Smirnov (K-S) statistic

The K-S measure is based on K-S statistic obtained from performing K-S test, which measures the absolute max distance between the eCDFs of simulated and real dataset. The K-S statistics is in range [0, Inf]. The K-S measure was obtained by log-transformation followed by the transformation rule defined previously.

5. Scaled area between empirical cumulative distribution functions (eCDFs)

The difference between the eCDFs of the properties in simulated and real dataset. The absolute value of the difference was then scaled such that the difference between the largest and smallest value becomes 1. The area under the curve was calculated using the Trapezoidal Rule. The final value is in the range of [0,1], where a value closer to 1 indicates greater differences between the data properties distributions of the real and simulation data. The value was then reversed by taking 1 minus the value such that it follows the general pattern of higher value being better.

6. Runs statistics

The Runs statistics is the statistic from a one-sided Wald-Wolfowitz runs test. The values from the simulated and real dataset were ordered and a runs test was performed. The null hypothesis is that the sequence is a random sequence with no clear pattern of values from simulated or real dataset next to each other in position.

3.2.4 *Methods comparison through multi-step score aggregation*

In order to summarise the results from multiple datasets and multiple criteria, we implemented the following multi-step procedure to aggregate the KDE scores.

First, we aggregated the KDE scores within each dataset. For most methods, each cell type in a dataset containing multiple cell types was simulated and evaluated separately for the reason mentioned in the previous section. This resulted in multiple KDE scores for a single dataset, one for each cell type. To

aggregate the scores into a single score for a dataset, we calculated the weighted sum by using the cell type proportion as weight, defined as the follows:

$$\sum_{i=1}^{n} (x_i * w_i) \tag{4}$$

where n is the number of cell types in the simulated or original datasets, x_i is the evaluation score of the ith cell type and w_i is the cell type proportion of the ith cell type.

Since each method was evaluated using multiple datasets, we then summarised the performance of each method across all datasets by taking the median score. This resulted in a single score for each method on each criterion, which then enabled us to readily rank each method by comparing the score. Cases where a method was not able to produce result on particular dataset were not considered in the scoring process. The reasons for failing to simulate a data include not completing the simulation in the given time limit, error arising in the simulation methods during the simulation process, and special cases in which the simulation method is limited to an input dataset containing two or more cell types and cannot generate result on datasets containing a single cell type. The breakdown of the number of datasets successfully simulated and the number of failed cases are reported in detail in Figure A1.

Finally, the overall rank of each method was computed by firstly calculating its rank for each criterion and then taking the mean rank across all criteria.

3.2.5 *Evaluation of biological signals*

The five categories of biological signals evaluated in this study were adapted from29 and their descriptions are detailed below.

1. DE (limma)

This is the typical differentially expressed genes. Limma (Ritchie *et al.*, 2015b) was performed to obtain the log fold change associated with each gene. We selected genes with log2 fold change > 1.

2. DE (DEsingle)

This finds the differentially expressed genes using a DE detection method DEsingle (Miao *et al.*, 2018) that is specifically designed for scRNA-seq data.

3. DV

DV stands for differentially variable genes. Bartlett's test for differential variability was performed to obtain the P-value associated with each gene.

4. DD

DD stands for differentially distributed genes. K-S test was performed to obtain the P-value associated with each gene.

5. DP

DP is defined as differential proportion genes. We considered genes with log2 expression greater than 1 as being expressed and otherwise as non-expressed. A chi-square test was then performed to compare the proportion of expression of each gene between two cell types.

6. BD

BD is defined as bimodally distributed genes. Bimodality index defined using the below formula was calculated for each gene:

$$\mathrm{BI} = \frac{|m_1 - m_2|}{s\sqrt{p(1-p)}} \tag{5}$$

where m1 and m2 are the mean expression of genes in the two cell types, respectively, s is the standard deviation and p is the proportion of cells in the first cell type.

For the first five categories, genes with P-value < 0.1 (Benjamini-Hochberg adjusted) were selected. This higher threshold was used instead of the typical threshold of 0.05 to result in a higher proportion of biological signals, as larger value would enable clearer differentiation of methods' performance. For the last category, we used bimodality index (Wang *et al.*, 2009) >0.03 as the cut-off to yield a reasonable proportion of BD genes (Figure A6).

To quantify the performance of each method, we used SMAPE (Armstrong, 1978):

$$\text{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{\frac{A_t+F_t}{2}} \tag{6}$$

where F_t is the proportion of biological signals in simulated data and A_t is the proportion in the corresponding real data, n is the number of data points, one from each dataset evaluated. The proportion was calculated as the number of biological signal genes divided by the total number of genes in a given dataset.

3.2.6 *Evaluation of scalability*

To reduce potential confounding effect, we measured scalability using the Tabula Muris dataset only. The dataset was subset to the two largest cell types and random samples of the cells without replacement were taken to generate datasets containing 50, 100, 250, 500, 750, 1000, 1250, 1500, 2500, 3000, 4000, 6000 and 8000 cells with equal proportion of the two cell types.

Running time of each method was measured using the Sys.time function built-in R and the time.time function built-in Python. Tasks that did not finish within the given time limit are considered as no result generated. To record the maximal memory for R methods we used the function Rprofmem in the built-in utils Package in R. For Python methods we used the psutil package and measured the maximal Resident Set Size. All measurements were repeated three times and the average was reported.

In the majority of methods, simulation was performed in a two-step process. In the first step, a range of properties is estimated from a given dataset. This set of properties are then used in the second step of generating the simulation data. For these methods, the time and memory usage of the two steps was recorded separately and shown in Figure A1. For other methods where the two processes were completed in one single function, we measured the time and memory usage of this single step and used a dashed line to indicate these methods in Figure A1.

In order to compare and rank the methods as shown in Figure 3.2, we summed the time and memory of the methods that use two-step procedure and displayed the total time and memory usage, such that their results became comparable with methods that involve one single step. Some methods did not complete the simulation within the given time, and the time and memory usage were unable to be recorded as the result. These timed out simulations would bias the result when ranking the methods based on the total time and memory usage. To account for this case, we assigned these simulation jobs a total time usage as the time limit and a memory usage as the memory of the previous simulation task. For example, a method that failed to simulate 8000 cells within the time limit of 8h was assigned 8h as the total time usage, and a memory usage as the memory recorded when simulating the previous job of 6000 cells.

3.2.7 *Evaluation of impact of data characteristics*

We selected a subset of datasets to examine the impact of the number of cells and sequencing technologies. Briefly, each dataset was split into 50% training and 50% testing. Transformed KDE score was then calculated from the raw score obtained from all methods across the selected datasets, resulting in values ranging between 0 and 1.

3.2.7.1 *Impact of number of cells*

To assess the impact of the number of cells on the accuracy of data property estimation, we used the Tabula Muris dataset subset to the two largest cell types and sampled to create datasets of 100, 200, 500, 1000, 1500, 2000, 2500, 3000, 5000, 6000, 8000, 12,000 and 16,000 cells. Each dataset was split into 50% training and 50% testing as previously described.

Linear regression was fitted using the lm function in the built-in stats package in R for each of the 13 data properties. This resulted in a total of 13 regression models with the formula defined as:

$$y = \beta_0 + \beta_1 x_1 \tag{7}$$

The response variable y was the KDE score corresponding to the data property and the exploratory variables x1 was the number of cells measured in 1000.

3.2.7.2 *Impact of the sequencing protocols*

To assess the impact of the sequencing protocols while avoiding potential batch effect, we utilised two sets of datasets from the same study (Ding *et al.*, 2020) that sequenced the same tissue type using multiple protocols. It contains human PBMC data generated using the following six protocols, 10x Genomics, CEL-seq2, Drop-seq, inDrops, Seq-Well and Smart-seq2 and mouse cortex cells using the following four protocols of sci-RNA-seq, 10x Genomics, DroNc-seq and Smart-seq2.

ANOVA was fitted using the built-in stats package in R to examine whether there was significant change in mean KDE score across the above datasets of different sequencing technologies for each simulation method. P-values were displayed on the figures.

3.2.8 *Data availability*

All datasets used in this study are publicly available. Details on each dataset including accession numbers and source websites are listed in A3. Curated version of the datasets is available as a Bioconductor package under the name SimBenchData (`https://bioconductor.org/packages/devel/data/experiment/html/SimBenchData.html`).

3.2.9 *Code availability*

The benchmark framework is available as an R package at `https://github.com/SydneyBioX/SimBench34`. A Shiny application for interactively exploring the results is available at `http://shiny.maths.usyd.edu.au/`.

3.3 RESULTS

3.3.1 *A comprehensive benchmark of scRNA-seq simulation methods on four key sets of evaluation criteria using diverse datasets and comparison measure*

Our SimBench framework evaluates 12 recently published simulation methods specifically designed for single-cell data (Figure 3.1a, Table 1 and Table A1). To ensure robustness and generalisability of the study results and account for variability across datasets (Figure A2), we curated 35 public scRNA-seq datasets (Figure 3.1b and Figure A3) that include major experimental protocols, tissue types, and organisms. To assess a simulation method's performance on a given dataset, SimBench splits the data into input data and test data (referred to as the real data). Simulation data is generated based on the data properties estimated from the input data and compared with the real data in the evaluation process (Figure 3.1c). Using four key sets of evaluation criteria (Figure 3.1c-d), we systematically compare the single-cell simulation methods' performance for 432

simulation data representing 12 simulation methods applied to 35 scRNA-seq datasets.

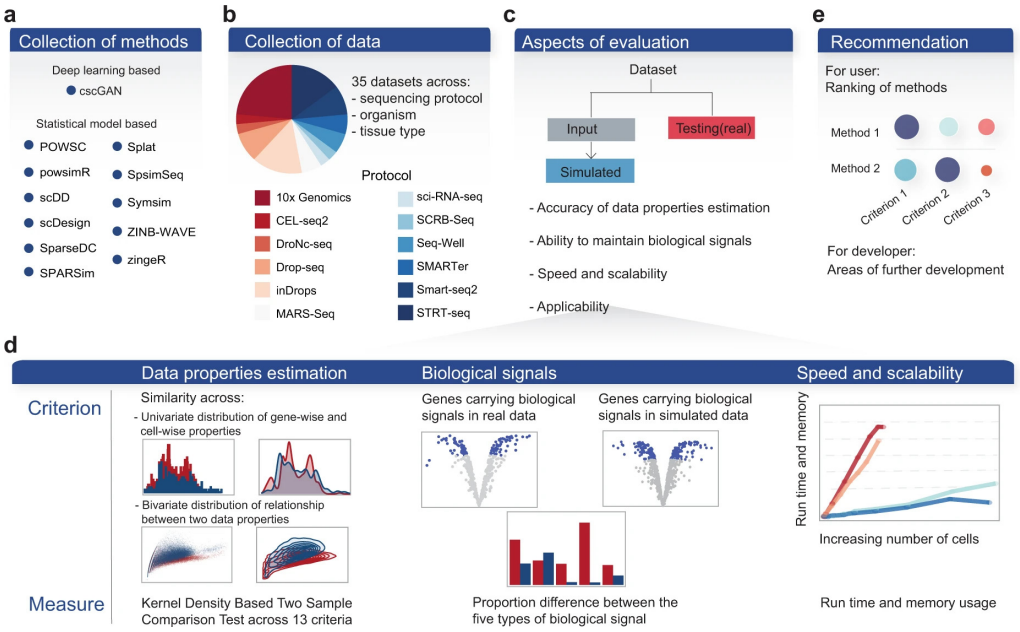


Figure 3.1: Schematic of the benchmarking workflow. (a) A total number of 35 datasets, covering a range of protocols, tissue types, organisms and sample size was used in this benchmark study. (b) We evaluated 12 simulation methods available in the literature to date. (c) Multiple aspects of evaluation were examined in this study, with the primary focuses illustrated in detail in panel (d). (e) Finally, we summarised the result into a set of recommendations for users and identified potential areas of improvement for developers.

The first set of evaluation criteria, termed data property estimation, aims to assess how realistic is a given simulated data. To address this, we first defined the properties for a given dataset with 13 distinct criteria and then developed a comparison process to quantify the similarity between the simulated and real data (Figure A3). The 13 criteria capture both the distributions of genes and cells as well as higher-order interactions, such as mean–variance relationship of genes. We anticipated that not all simulation methods will place emphasis on the same set of data properties and it is thus important to incorporate a wide range of criteria. We then examined a number of statistics for measuring distributional similarity (Soneson and Robinson, 2018c). Figure A4 shows that all statistics show similar performance with mean correlation of 0.7 and we

have chosen to use the kernel density based global two-sample comparison test statistic (Duong *et al.*, 2012) (KDE statistic), in our current study as it is applicable to both univariate and multivariate distributions.

The other three sets of evaluation criteria seek to assess each simulation method's ability to maintain biological signals and computational scalability and its applicability. For biological signals, we measured the proportion of differentially expressed (DE) genes obtained in the simulated data using DE detection methods designed for bulk and single-cell RNA-seq data, as well as four other types of gene signals of differentially variable (DV), differentially distributed (DD), differential proportion (DP) and bimodally distributed (BD) genes (see "SimBench Framework"). A similar proportion to the real data would indicate an accurate estimation of biological signals present in the data. Scalability reflects the ability of simulation methods to efficiently generate large-scale datasets. This is measured through computational run time and memory usage with respect to the number of cells. Applicability examines the practical application of each method in terms of whether it can estimate and simulate multiple cell groups and allow simulation of differential expression patterns. Overall, our framework provides recommendations by taking into account all aspects of evaluation (Figure 3.1e).

3.3.2 *Comparison of simulation methods revealed their relative performance on different evaluation criteria*

Through ranking the 12 methods on the above four sets of evaluation criteria, we found that no method clearly outperformed other methods across all criteria (Figure 3.2). We therefore examined each set of criteria individually in detail below and the variability in methods' performance within and across the four sets of evaluation criteria.

For data property estimation, we observed variability in methods' performance across the 13 criteria. ZINB-WaVE, SPARSim and SymSim are the three methods that performed better than the others across almost all 13 data properties

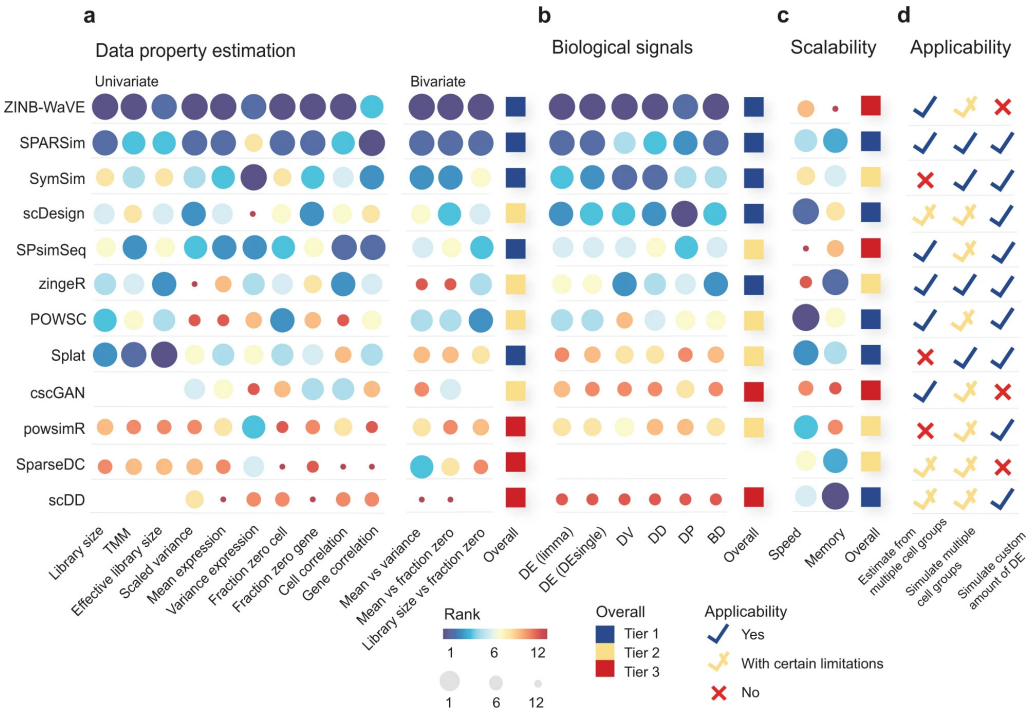


Figure 3.2: Ranking of methods across key aspects of evaluation criteria. The colour and size of the circle denote ranking of methods, where a large blue circle represents the best possible rank of 1. Missing space indicates where a measurement was not able to be obtained, for example, due to the output format being normalised count instead of raw count (see "SimBench Framework"). The ranks within each criterion were summarised into an overall tier rank, with tier 1 being the best tier. (a) Ranking of methods within data property estimation, ranked by median score across multiple datasets. (b) Ranking of methods within biological signals, ranked by median score across multiple datasets. (c) Scalability was ranked by the total computational speed and memory usage required for properties estimation and dataset generation across datasets. (d) Applicability was examined in terms of three criteria, which are explained in more detail in Table 1. The number of datasets used in the entire evaluation process and the success rate of each method on running the datasets is reported in Figure A1.

(Figure 3.2a). For the remaining methods, a greater discrepancy was observed between the 13 criteria, in which the rankings of methods based on each criterion do not show any particular relationship or correlation structure. Overall,

our results highlight the relative strengths and weaknesses of each simulation method on capturing the data properties.

We observed that some methods (e.g. zingeR and scDesign) that were not ranked the highest in data properties estimation performed well in retaining biological signals (Figure 3.2b). scDesign is designed for the purpose of power calculation and sample size estimation, while zingeR is designed to evaluate the DE detection approach in its publication and thus both methods require an accurate simulation and estimation of biological signals, particularly differential expression. It is not unexpected that they ranked highly in this aspect despite not being the most accurate in estimating other data properties.

For computational scalability, the majority of methods showed good performance with runtime of under 2h and memory consumption of under eight gigabytes (GB) (Figure A5) when tested on the downsampled Tabula Muris dataset19 with 50–8000 cells (see "SimBench Framework"). However, some top performing methods, such as SPsimSeq and ZINB-WaVE revealed poor scalability (Figure 3.2c). This highlights the potential trade-off between computational efficiency and complexity of modelling framework. SPsimSeq, for example, involves the estimation of correlation structure using Gaussian-copulas model and scored well in maintaining gene- and cell-wise correlation. Its advantage came at the cost of poor scalability, taking nearly 6h to simulate 5000 cells. Thus, despite the ability to generate realistic scRNA-seq data, the usefulness of such methods may be partially limited if a large-scale simulation dataset is required. In contrast, methods such as SPARSim, which was ranked second in parameter estimation as well as being one of top tier methods in scalability, may better suit needs if a large-scale simulation dataset is required by users.

Lastly, we found that different simulation methods satisfy different numbers of the applicability criteria (Figure 3.2d). This is due, in part, to the fact that not all simulation methods are designed as general purpose simulation tools. For example, powsimR was originally designed as a power analysis tool for differential expression analysis but was included as a simulation tool by a number of simulation studies (Li and Li, 2019; Zhang *et al.*, 2019c) in their performance

comparison with other simulation methods. Being a power analysis tool, its primary usage is to simulate two cell groups from a homogenous cell population with a user-defined amount of differential expression. In contrast, a number of other methods such as SPARSim, SymSim and Splat that are originally intended as general purpose simulation tools are able to simulate multiple cell groups with user-defined differential expression patterns. We have outlined the primary purpose and the limitations of each method on this front in more detail in Table 1, as well as the downstream applications of each method as demonstrated in their respective publications (Table 2), to guide users in making informed decisions on methods that best suited to their needs.

3.3.3 *Impact of data- and experimental-specific characteristics on model estimation*

Aside from comparing the overall performance of methods to guide method selection, it is also necessary to identify specific factors influencing the outcome of simulation methods. Here, we examined the impact of data- and experimental-specific characteristics including cell numbers and sequencing protocols on simulation model estimation.

To explore the general relationship between cell number and accuracy of data property estimation across simulation methods, we evaluated each method on thirteen subsamples of Tabula Muris data with varying numbers of cells but fixed number of cell types (see "SimBench Framework"). Through regression analysis, we found certain data properties such as mean–variance relationships were more accurately estimated under datasets with larger numbers of cells, as shown by the positive regression coefficients (Figure 3.3a and Figure A6). Nevertheless, most other data properties in the simulated data were negatively correlated with the increasing number of cells (e.g. library size, gene correlation). These observations suggest that overall, the increasing cell number may be accompanied by the increasing complexity of data and thus maintaining data properties may become more challenging. Future method development

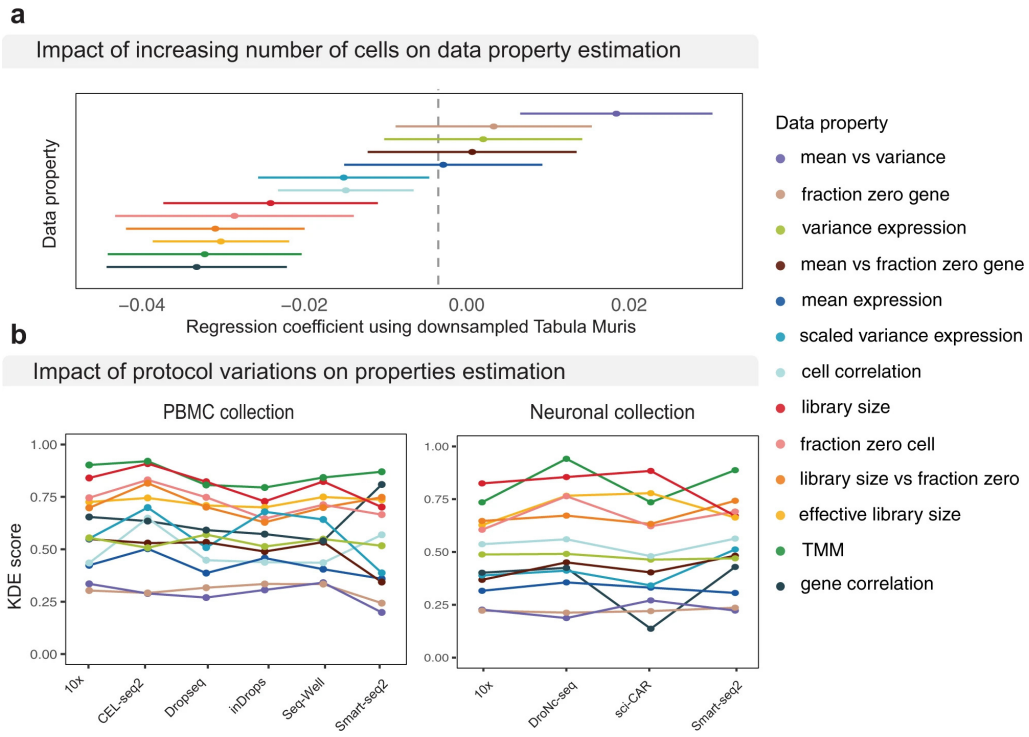


Figure 3.3: Impact of dataset characteristic on method performance. (a) Impact of the number of cells on selected properties (see Figure A6 for all properties). Line shows the trends with increasing cell numbers. Dot indicates where a measurement is taken. (b) Impact of protocols was examined using two collections of datasets (see Figure A7 for individual methods). Boxplots show the individual score of each property for each method.

should consider this factor as an aspect of evaluation when assessing model performance.

To examine the impact of sequencing protocols, we utilised datasets consisting of multiple protocols applied to the same human PBMC and mouse cortex samples from the same study (Ding *et al.*, 2020). Figure 3.3b and Figure A7 reveal no substantial impact was introduced by protocol difference on the overall simulation results, as indicated by the flatness of the line representing the accuracy of each data property across each protocol. Taken together, these results indicate that the choice of reference input being shallow sequencing or deep sequencing has no substantial impact on the overall simulation results. Given that SymSim and powsimR are the only two methods that require specification of input data as either deep or shallow protocols, these results suggest that a

general simulation framework for the two major classes of protocols may be sufficient.

3.3.4 *Comparison across criteria revealed common areas of strength and weakness*

While the key focus of our benchmark framework is assessing methods' performance across multiple criteria, we can further use these results to identify criteria where most methods performed well or were lacking (Figure 3.4a). Comparing across criteria, those that display a large difference between the simulated and real data for most methods are examples of common weakness. This ability to identify common weakness has implications for future method development as it highlights ongoing challenges of simulation methods.

First, we compared the accuracy of maintaining each data property, where a larger KDE score indicates greater similarity between simulated and real data. Figure 3.4b shows data properties relating to the higher-order interactions including mean–variance relationship of genes revealed larger differences between the simulated and real data. In comparison, a number of gene- and cell-wise properties such as fraction of zero per cell had relatively higher KDE scores, suggesting they were more accurately captured by almost all simulation methods. These observations thus highlight the difficulty in incorporating higher-order interactions by current simulation methods in general, and the potential area for method development.

The ability to recapture biological signals was quantified using the metric symmetric mean absolute percentage error (SMAPE), where a score closer to 1 indicates greater similarity between simulated and real data (see "SimBench Framework"). We found differentially distributed (DD) and differential proportion (DP) genes exhibited a greater difference between simulated and real data (Figure 3.4b). We also noted that four out of the 12 methods consistently had very low SMAPE score of between 0 and 0.3, indicating the biological signals in the simulated data were at a very different proportion to that in real data. Upon closer examination, these methods simulated close to zero proportions of

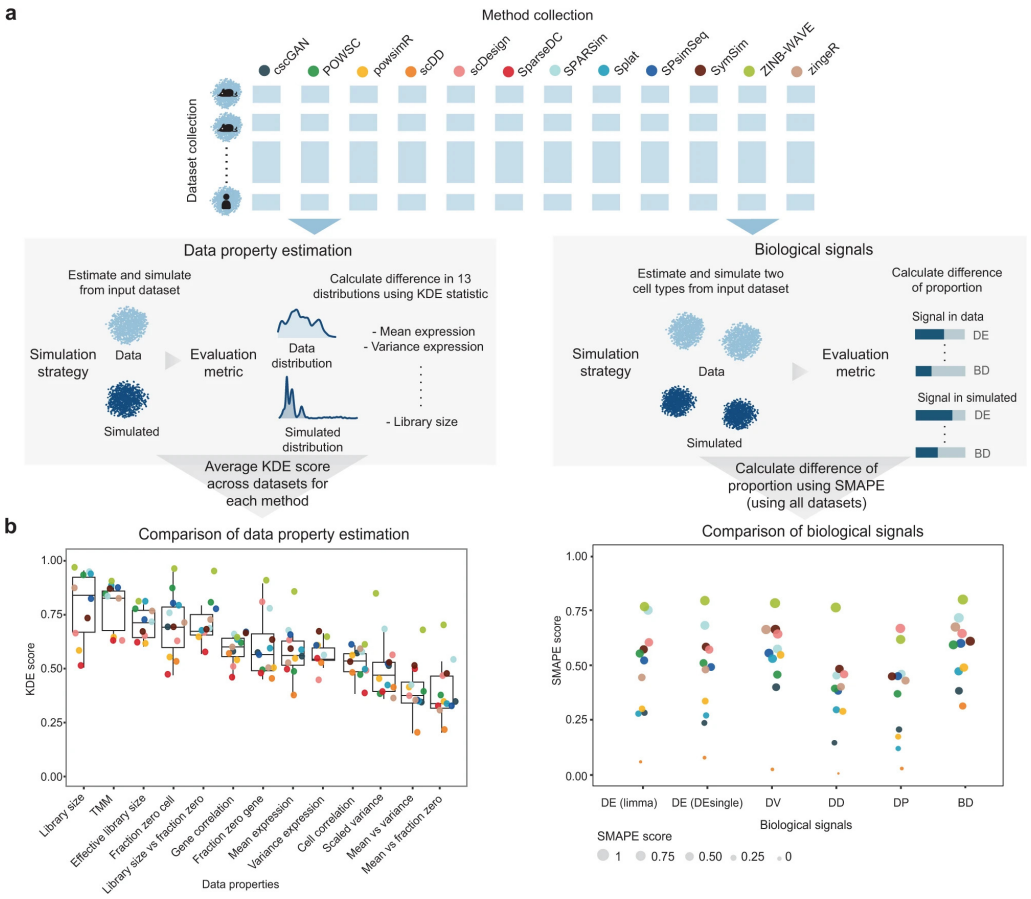


Figure 3.4: Comparison of criteria in data property estimation and in biological signals. (a) Evaluation procedure for data property estimation and biological signals. (b) The evaluation results and the comparison of criteria within the two aspects of evaluation. For data property estimation, the KDE score measures the difference between the distribution of 13 data properties in simulated and in real data. A score close to 1 indicates a greater similarity. Each boxplot shows the distribution of the median KDE score attained by all simulation methods (n = 12), with the KDE score attained by each method shown in individual data point. The box represents quartiles, the line represents the median, the lower and upper whisker represents the bottom 25% and top 25% of the data. Outliers can be seen from the individual data points that are outside the whiskers. For biological signals, the SMAPE score measures the percentage difference between the proportion of biological signals detected in simulated and in real data. A score of 1 indicates no difference in the biological signals detected in real and simulated data and a score of 0 indicates maximal difference.

biological signals irrespective of the true proportion in the real data (Figure A6). Together, these observations point to the need for better strategies to simulate biological signals.

3.4 DISCUSSION

We presented a comprehensive benchmark study assessing the performance of 12 single-cell simulation methods using 35 datasets and a total of 25 criteria across four aspects of interest. Our primary focus was on assessing accuracy of data property estimation and various factors affecting it, ability to maintain biological signals and computational scalability, as well as applicability. Additionally, using these results we also identified common areas of strength and weakness of current simulation tools. Altogether, we highlighted recommendations for method selection and identified areas of improvement for future method development.

We found that various underlying models were used for different simulation methods (Table 1). Each of the five top performing methods in category 1, for instance, uses a different underlying statistical approach (Table 1). As another example, the three methods ZINB-WaVE, zingeR and powsimR differ substantially in detail despite the fact that they are all inspired by representing the observed counts using the NB family. Specifically, zingeR uses NB distribution to fit the mean and dispersion of the count data and model the excess zero using the interaction between gene expression and sequencing depth using additive logistic regression model. powsimR uses the standard ZINB distribution to fit the mean and dispersion of the count data, with the zero inflation modelled using binomial sampling. In ZINB-WaVE, the ZINB distribution is used to fit the mean and dispersion of the count data, as well as the probability that a zero is observed. Additionally, the estimation of mean and zero probability incorporates an additional parameter adapted from the RUV framework (Gagnon-Bartsch and Speed, 2012) to capture unknown cell-level covariates. Therefore, while both powsimR and ZINB-WaVE use ZINB distribution

to fit the count data, the actual model differs. Interestingly, while deep learning methods have dominated various fields and applications, cscGAN, a deep learning-based model, for scRNA-seq data simulation only had moderate performance compared to the other models. This may be due to the large number of cells required for training the deep neural network in cscGAN as was demonstrated in their original study (Marouf *et al.*, 2020).

Based on the experiments conducted, we identified several areas of exploration for future researchers. Maintaining a reasonable amount of biological signal is desirable and was not well captured by a number of methods. We also observed the genes generated by some methods (Table 1) were assigned uninformative names such as gene 1 and exhibit no relationship with genes from the real data. This limited us to assessing the proportion of biological signals in the simulated data, instead of assessing whether the same set of genes carrying biological signals (e.g. marker gene) are maintained in the simulated data. Incorporating the additional functionality of preserving biologically meaningful genes is likely to increase the usability of future simulation tools. Furthermore, we noted that several simulation studies only assessed their methods based on a number of gene- and cell-wise properties and did not examine higher-order interactions. Those studies are thus limited in the ability to uncover limitations in their methods. In comparison, our benchmark framework covered a comprehensive range of criteria and identified relative weakness of maintaining certain higher-order interactions compared to gene- and cell-wise properties.

As expected, we identified that none of the simulation methods assessed in this study could maintain the heterogeneity in cell population that was due to patient variability. This is potentially related to the paradigm used by current simulation techniques, as some methods implicitly require input to be a homogeneous population. For instance, some simulation studies inferred modelling parameters and performed simulation on each cell type separately when the reference input contains multiple cell types. However, experimental datasets with data from multiple samples, for example multiple patients, would be characterised by sample-to-sample variability within a cell type. This cellular het-

erogeneity is an important characteristic of single-cell data and has key applications such as identification of subpopulations. The loss of heterogeneity can thus be a limiting factor, as in some cases the simulation data could be an oversimplified representation of single-cell data. Future research such as phenotype-guided simulation (Sun *et al.*, 2020) can help to extend the use of simulation methods.

Finally, we found there exists various trade-offs between the four aspects of criteria and having a well-rounded approach could be more important than a framework that performs best on one aspect but limiting in the other aspects. For example, as single-cell field advances and datasets with hundreds of thousands of cells become increasingly common, users may be interested in simulating large-scale datasets to test the scalability of their methods. As a result, methods that rank highly on scalability while also performing well on other aspects (e.g. SPARSim, scDesign and Splat) may be more favourable than other methods under these scenarios. We also note that due to the primary intended purpose of each method, not all methods allow users to customise the number of cell groups and the amount of differential expression between groups. Method that offers a well-rounded approach across multiple aspects of interests is therefore a direction of future research.

While we aim to provide a comprehensive assessment of available simulation methods, our study is not without limitations. For example, a few methods were excluded in this study due to their unique properties. SERGIO (Dibaeinia and Sinha, 2020) is able to simulate regulation of genes by transcriptional factors, and therefore requires gene regulatory networks as one of the inputs. Both PROSSTT (Papadopoulos *et al.*, 2019) and dyngen (Cannoodt *et al.*, 2021) are designed to simulate scRNA-seq data with trajectory information and require user-defined lineage trees. Lun (Lun and Marioni, 2017) was originally designed to tackle confounding plate effects in DE analysis and it requires plate information to be specified in the input. These simulation methods may need special considerations and evaluation criteria that could not be captured by the general framework in this study. Although the choice of DE detection meth-

ods could affect the evaluation of the simulation methods, our evaluation using both limma, a DE method originally designed for bulk RNA-seq data, and DEsingle, a DE method specifically designed for scRNA-seq data demonstrate a high agreement of the rankings of simulation methods based on the two DE methods (Figure 3.2b).

In conclusion, we have illustrated the usefulness of our framework by summarising each method's performance across different aspects to assist with method selection for users and identify areas of further improvement for method developers. We advise users to select the method that offers the functionality best suited to their purpose and developers to address the limitations of current methods.

The evaluation framework has been made publicly available as the R package SimBench (`https://github.com/SydneyBioX/SimBench`). SimBench allows any new simulation methods to be readily assessed under our framework. It requires two inputs including the simulated data generated by any simulation method and the real data that was used as the reference input to generate the simulated data. SimBench then runs the evaluation procedure as performed in this study. We also provide all datasets used in this study as a Bioconductor data package SimBenchData (`https://bioconductor.org/packages/devel/data/experiment/html/SimBenchData.html`). Together these two packages enable future simulation methods to be assessed and compared with the methods benchmarked in this study.

Additionally, we provide a Shiny application for interactively exploring the results presented in this study hosted at `http://shiny.maths.usyd.edu.au/`. The application allows users to select datasets of their interest, such as within a certain range of cell numbers, and examine methods performance based on the specified datasets. Furthermore, we will provide updates to the website to include the benchmark results from new simulation methods when they become available so that our comparative study will stay up-to-date and will support future method development.

Table 1: scRNA-seq simulation methods evaluated in this study.

Methods	Year of publication	Approach	Estimate from multiple cell groups	Simulate multiple cell groups	Customise DE expressiona	Assign gene name to generated data	Primary purpose as general simulation?
scDD (Korthauer *et al.*, 2016)	2016	Dirichlet process mixture of normals	Restricted to two groups	Restricted to two groups	Yes	No	No, used for generating differentially distributed genes defined in the scDD study and evaluating the scDD framework
Splat (Zappia *et al.*, 2017)	2017	Gamma distribution for modelling mean expression; Poisson distribution for modelling count	No, requires a homogenous population (e.g. one cell type)	Yes, can simulate any number of groups	Yes	No	Yes
powsimR (Vieth *et al.*, 2017)	2017	Negative binomial or zero-inflated negative binomial model	No, requires a homogenous population (i.e. one cell type)	Restricted to two groups	Yes	Yes	No, power analysis tool for single-cell and bulk RNA-seq
SparseDC (Lin *et al.*, 2020b)	2017	Optimisation framework	Restricted to two conditions with multiple cell groups within each condition	Restricted to two conditions with multiple cell groups within each condition	Yes	No	No, used for generating the simulation data for assessing the performance of the SparseDC clustering method
zingeR (Van den Berge *et al.*, 2018)	2018	Negative binomial model with additive logistic regression to account for zeros	Yes, can estimate from any number of groups	Yes, can simulate any number of groups	Yes	No	No, used for generating simulation data for assessing the performance of the zingeR DE method
ZINB-WaVE (Risso *et al.*, 2018)	2018	Zero-inflated negative binomial model	Yes, can estimate from any number of groups	Restricted to the groups in the input data	No	No	No, dimension reduction method for scRNA-seq
SymSim (Zhang *et al.*, 2019c)	2019	Kinetic model using Markov chain Monte Carlo	No, requires a homogenous population (i.e. one cell type)	Yes, can simulate any number of groups	Yes	No	Yes
scDesign (Li and Li, 2019)	2019b	Gamma-normal mixture model	Restricted to one and two groups	Restricted to one and two groups	Yes	No	No, power analysis tool for scRNA-seq
SPARSim (Baruzzo *et al.*, 2020)	2020	Gamma distribution for modelling expression; multivariate hypergeometric distribution for modelling technical variability	Yes, can estimate from any number of groups	Yes, can simulate any number of groups	Yes	Yes	Yes
SPsimSeq (Assefa *et al.*, 2020)	2020	Estimation of probability distribution uses fast log-linear model-based density estimation method; Gaussian-copulas for modelling gene–gene correlation	Yes, can estimate from any number of groups	Restricted to the groups in the input data	Yes	Yes	Yes
POWSC (Su *et al.*, 2020)	2020	Mixture of zero-inflated Poisson for modelling inactive transcription; log-normal Poisson for modelling the active transcription	Yes, can estimate from any number of groups	Restricted to the groups in the input data	Yes	No	No, power analysis tool for scRNA-seq
cscGAN (Marouf *et al.*, 2020)	2020	Generative adversarial network with Wasserstein distance	Yes, can estimate from any number of groups	Restricted to the groups in the input data	No	Yes	Yes

Table 2: scRNA-seq simulation methods and their downstream applications as demonstrated in their publication. We included both the benchmarked and non-benchmarked methods for comprehensiveness.

	DE gene detection	DS gene detection	Batch effect removal	Data imputation	Trajectory inference	RNA velocity	Gene regulatory network
SERGIO					✓	✓	✓
Splat	✓		✓		✓		
SymSim	✓		✓		✓		
PROSSTT					✓	✓	
ESCO	✓			✓	✓		
POWSC	✓						
SPARSim	✓		✓				
SPsimSeq	✓		✓				
muscat	✓	✓					
SCRIP	✓						
BASiCs	✓						
scDesign2	✓						
scDesign	✓						
ZINB-WaVE	✓						
hierarchicell	✓						
powsimR	✓						
dyngen	✓		✓		✓	✓	✓
scDD	✓						
SparseDC	✓						
cscGAN					✓		

4

SCFEATURES: MULTI-VIEW REPRESENTATIONS OF SINGLE-CELL AND SPATIAL DATA FOR DISEASE OUTCOME PREDICTION

Precision medicine is concerned with the analysis of the differences amongst patients, in which the patient is the unit of interest. With the recent surge of large-cohort scale single-cell research, it is of critical importance that analytical methods can fully utilize the comprehensive characterization of cellular systems that single-cell technologies produce to provide insights into samples from individuals. However, currently there is little consensus of the best ways to compress information from the complex data structures of these technologies to summary statistics that represent each sample (e.g. individuals).

In this chapter, we contribute to the field by developing scFeatures, an approach that creates interpretable cellular and molecular representations of single-cell and spatial data at the sample level (Cao *et al.*, 2022b). scFeatures generates features across six categories representing different molecular views of cellular characteristics. These include i) cell type proportions, ii) cell type specific gene expressions, iii) cell type specific pathway expressions, iv) cell type specific cell-cell interaction (CCI) scores, v) overall aggregated gene expressions and vi) spatial metrics.

This chapter demonstrates that the different types of features constructed enable a more comprehensive multi-view representation of the expression data. In addition, summarising a broad collection of features at the sample level is both important for understanding underlying disease mechanisms in different experimental studies and for accurately classifying disease status of individuals.

scFeatures is publicly available as an R package at `https://github.com/SydneyBioX/scFeatures`.

4.1 INTRODUCTION

Recent single-cell or near single-cell resolution omics technologies such as spatial transcriptomics enable the discovery of cell- and cell type specific knowledge and have transformed our understanding of biological systems, including diseases (Longo *et al.*, 2021). Key to the exploration of such data is the ability to untangle and extract useful information from their high feature dimensions (Yang *et al.*, 2021) and uncover hidden insights. A plethora of computational methods has been developed on this front, with the main focus on individual cell analysis (Stegle *et al.*, 2015), such as cell type identity (Abdelaal *et al.*, 2019; Kim *et al.*, 2021) and pseudotime ordering within a lineage (Saelens *et al.*, 2019). While these tools enable characterisation of individual cells, there is a lack of tools that allow for the representation of individual samples based on their cellular characteristics and the investigation of how these cellular properties are driving disease outcomes. With the recent surge of multi-condition and multi-sample single-cell studies on large sample cohort (Lin *et al.*, 2020a), the next frontier of research is on representing and characterising cellular properties at the sample (e.g. individual patient) level for linking such information with the disease outcome.

Creating a representation of each sample from the collection of sequenced cells is a crucial step for subsequent analysis as successful modelling and interpretation of disease outcome requires biologically relevant learning features from the data. While using the original expression matrix as the input to various models could inform the change in transcriptomics level across disease conditions, the ability to represent the data with other layers of information is critical for uncovering additional insights given the complex and nonlinear relationships among the feature dimensions (e.g. interaction of genes, gene networks and pathways). The single-cell field has a wealth of tools for data exploration (Wu and Zhang,

2020) which enables exploration of biology underlying the individuals. Most current tools are not specifically designed to derive a set of features that can be used to represent an individual. Yet, with careful adaptation, a number of approaches can be used to construct novel molecular representations of individual samples. Cell-cell interactions tools (Armingol *et al.*, 2020), for example, calculate cell type specific signalling scores between pairs of ligand and receptor molecules. The interaction scores can be used to represent the intercellular communications of cells and cell types in a sample. Another example is gene set enrichment analysis (Maleki *et al.*, 2020) which infers the pathway enrichment score of individual cells. By summarising the scores across cell types, a cell type specific representation of the pathway enrichment of each sample can be constructed. Our previous publication (Cao *et al.*, 2019) is another motivating example that demonstrates cell type proportion can be used to represent samples for distinguishing between disease conditions.

To this end, we develop scFeatures, a tool that generates a large collection of interpretable molecular representations for individual samples in single-cell omics data, which can be readily used by any machine learning algorithms to perform disease outcome prediction and drive biological discovery. Together, scFeatures generates features across six categories representing different molecular views of cellular characteristics. These include i) cell type proportions, ii) cell type specific gene expressions, iii) cell type specific pathway expressions, iv) cell type specific cell-cell interaction (CCI) scores, v) overall aggregated gene expressions and vi) spatial metrics. The different types of features constructed thereby enables a multi-view of our data and enables a more comprehensive representation of the expression data. Based on the generated features, scFeatures produces an HTML report containing visual summaries of features most associated with conditions. In a collection of 17 published single-cell RNA-seq, single-cell spatial proteomics and spatial transcriptomics datasets, scFeatures reveal different feature classes are useful for predicting the disease outcomes in different datasets. Furthermore, through examining the selected features in two case studies, scFeatures uncovers cell types important to ulcerative colitis and stratified individuals with distinct survival outcomes in a triple negative

breast cancer dataset. Together, these results demonstrate that scFeature enables a data-driven generation (or feature engineering) and facilitate unbiased identification of feature classes most perturbed by the disease conditions.

4.2 SCFEATURES FRAMEWORK

4.2.1 *Data collection and processing*

4.2.1.1 *scRNA-seq*

To demonstrate scFeatures on scRNA-seq data, we collected data from four published studies and curated a total of 15 datasets from the studies. The data are described in detail below:

Six Ulcerative Colitis datasets: The UC data Smillie *et al.* (2019) sequenced healthy control, inflamed and non-inflamed colon biopsies from multiple individuals. The data was retrieved from Single Cell Portal with accession ID SCP259. We subset the data into epithelial, stromal cells and immune subset according to the original publication, resulting in the following 6 datasets:

- UC healthy vs non-inflamed (Epi)
- UC healthy vs non-inflamed (Fib)
- UC healthy vs non-inflamed (Imm)
- UC inflamed vs non-inflamed (Epi)
- UC inflamed vs non-inflamed (Fib)
- UC inflamed vs non-inflamed (Imm)

where Epi stands for epithelial, Fib stands for stromal and Imm stands for immune subsets. Inflamed, non-inflamed and healthy are conditions of interest.

Six Lung datasets: The lung data (Adams *et al.*, 2020) sequenced healthy control, idiopathic pulmonary fibrosis (IPF), and chronic obstructive pulmonary disease (COPD) biopsies from multiple individuals. The data was retrieved from Gene

Expression Omnibus (GEO) with accession ID GSE136831. We subset the data into epithelial, stromal cells and immune subset according to the original publication, resulting in the following datasets:

- Lung healthy vs IPF (Epi)
- Lung healthy vs IPF (Fib)
- Lung healthy vs IPF (Imm)
- Lung healthy vs COPD (Epi)
- Lung healthy vs COPD (Fib)
- Lung healthy vs COPD (Imm)

where healthy, IPF and COPD are conditions of interest.

Two melanoma data (Sade-Feldman *et al.*, 2019) sequenced immune cells from tumour biopsies of melanoma patients prior to and after treatment with immune checkpoint therapy. The data was retrieved from GEO with accession ID GSE120575. We subset the data into pre-treatment and post-treatment datasets. The conditions of interest in both datasets are non-responding and responding.

The COVID dataset (Schulte-Schrepping *et al.*, 2020a) sequenced peripheral blood mononuclear cells (PBMC) from COVID-19 individuals. The data was retrieved from European Genome-phenome Archive (EGA) with accession ID EGAS00001004571. We subset the original data into mild and severe individuals and consider the mild and severe disease stage as the conditions of interest.

4.2.1.2 *Spatial proteomics*

The triple negative breast cancer dataset (Keren *et al.*, 2019) measured the patient's protein expression using MIBI-TOF (multiplexed ion beam imaging by time of flight) technology. Data was obtained from `https://mibi-share.ionpath.com`.

4.2.1.3 *Spatial transcriptomics*

The amyotrophic lateral sclerosis dataset (Maniatis *et al.*, 2019) sequenced lumbar spinal cord tissue of ALS and control mouse at varying time points using the spatial transcriptomics technology. The data was retrieved from GEO with accession ID GSE120374. We used the subset of data sequenced at the disease onset time point.

4.2.2 *Implementation of feature types*

We generated 17 feature types that can be broadly categorised into six categories: i) cell type proportions, ii) cell type specific gene expressions, iii) cell type specific pathway expressions, iv) cell type specific CCI scores, v) overall aggregated gene expressions and vi) spatial metrics. All feature types except for the overall aggregated gene expressions category have different implementations for scRNA-seq and spatial data to better leverage the characteristics of different data types and the implementation details are described in Table B1.

For spot-based spatial transcriptomics, we performed the following additional processing in order to allow certain feature classes to be applicable. First, since the cell type specific feature categories require cell type information while the spot in spot-based data contains a mixed population of multiple cells, we used Seurat's TransferData function to predict the cell type probability of each spot. A published scRNA-seq data on mouse spinal cord with cell type labels was used as the reference (Sathyamurthy *et al.*, 2018). Then, given that each spot contains an unknown number of cells which vary across each spot, we weighted the contribution of each spot to the generated features by the relative number of cells it contains. We used library size as an estimate of the relative number of cells, motivated by a study that found a high correlation between the number of cells and library size of spots (Saiselet *et al.*, 2020). To calculate the relative number of cells, we binned the log2 transformed total library size of cells into 100 bins, and assigned each spot a relative number of cells ranging between 1 to 100 according to its bin. The cell type probability of each spot together with

the relative number of cells were used in the implementation of feature types for spatial transcriptomics.

4.2.3 *Correlation between features and feature classes*

Given scFeatures constructs a standard matrix of samples by features, we can readily compute the Pearson's correlation between individual features as shown in Figure B2. We subsampled 100 features from feature types that have more than 100 features to avoid the correlation plot being dominated by feature types with greater number of features.

To summarise the correlation between pairs of feature types, the following approach was taken. First, we calculated the Pearson's correlation between all features from a pair of feature types, such as proportion raw and gene mean celltype. This is repeated for each pairwise combination of feature types for each dataset. Then we subsampled 1000 values from the correlation values to reduce the computational burden of plotting. For ease of visual interpretation, the absolute values of the correlation values were taken.

These correlation values were further summarised by taking the average correlation values, followed by hierarchical clustering to cluster the feature types.

4.2.4 *Classification and survival analysis using generated features*

In scFeatures, we provide functionality to perform classification and survival analysis for the convenience of users. The classification function is a wrapper around a classification package classifyR (Strbenac *et al.*, 2015) that was published by our group earlier. By default, we use a random forest model, set the number of folds to three, perform 20 cross-validation and calculate F1-score. classifyR has an in-built feature selection function. We used the default setting that uses the feature selected from the random forest model built on the training set to evaluate on the test set. These were also the settings used to report the classification performance in this study and can be specified by the user.

The only exception being that 100 cross-validation was performed to obtain a more stable feature importance score for the case study on the "UC healthy vs non-inflamed (Fib)" dataset.

For survival analysis, we use a cox proportional-hazards model provided in the rms R package. By default, we set the number of folds to three, perform 20 cross-validation and calculate C-index. Note that as the cox model is not designed to take in a large number of features at once, unlike a typical classification model, we input one feature from the generated feature class at a time for building the cox model. The best C-index is reported as the performance for the feature class.

4.2.5 *Complementarity of the generated features*

To explore the complementarity of the generated features, we compared the classification accuracy of using features from individual feature types with using the combination of features from all feature types. In detail, we used the classification model described above, which is trained on all feature space to derive the feature importance. We then identified the top 8 features from each feature type and combined them into the "combined feature set". This set contains 96 features (8 features x 12 feature types) for the ALS dataset and 104 features (8 features x 13 feature types) for the other 15 datasets. The triple negative breast cancer dataset was excluded from this analysis as Cox proportional-hazards model is not designed to take in a large number of features at once. For fair comparison with the individual feature type, we used the top 100 features from each individual feature type. For feature types with less than 100 features, i.e., "proportion raw" and “proportion logit", we used all features. We used the random forest model, set the number of folds to three, performed 50 cross-validation and recorded the F1 score.

4.2.6 *Feature importance score*

The runTests function in ClassifyR outputs the features selected by the classification model. Since repeated cross-validation was performed, this generated one set of included features for each cross-validation process. Based on all the derived sets, the frequency of inclusion was considered as the feature importance score of each feature.

For the cell type specific feature category, given that each feature is associated with a cell type, it is also of interest to aggregate the feature importance score associated with each cell type. We approached by summing the feature importance score of all features associated with a cell type, then dividing by the number of features constructed for that particular cell type to adjust for the difference in the number of features per cell type. The final score was considered as the feature importance score of each cell type.

4.2.7 *Speed and memory usage*

To benchmark the scalability of the 17 features classes, we used the UC inflamed vs non-inflamed (Imm) dataset and took random samples to construct datasets with 1000, 2000, 3000, 5000, 10000, 20000, 30000, 50000, 70000 and 100000 cells. Each dataset contains the same 15 individuals and the same 15 cell types.

For the purpose of evaluating the features classes designed for spot-based data which require each spot to be associated with a cell type probability vector, we treated each cell as a "spot" and randomly created a cell type probability vector for each cell. Similarly, for the purpose of evaluating the feature classes under the category spatial metrics which require spatial coordinates of each cell, we randomly assigned a pair of x and y-coordinates to each cell. In addition, the cell type probability and number of cells in each spot was randomly generated to represent such data.

Runtime was measured using the built-in Sys.time function in R. Memory was measured by recording the peak resident set size, which measures the peak amount of memory that a process consumes across all cores. All code was run in parallel using 8 cores for three times and the average measurements were taken. All processes were carried out using a research server with dual Intel(R) Xeon(R) Gold 6148 Processor with 40 cores and 768 GB of memory.

4.2.8 *Data availability*

All data used in this study are publicly available. The accession links are reported in previous sections.

4.2.9 *Code availability*

scFeatures is publicly available as an R package at `https://github.com/SydneyBioX/scFeatures`.

4.3 RESULTS

4.3.1 *scFeatures performs multi-view feature engineering for single-cell and spot-based data*

We propose scFeatures, a new multi-view feature engineering framework that creates an interpretable representation of cellular level features for each individual sample from a given single-cell or spot-based expression dataset (Figure 4.1a). To capture the wide range of cellular information for sample classification (e.g., diseased versus healthy individuals) using single-cell data, we implemented an extensive collection of algorithms to extract over 50,000 interpretable features from a given dataset. These features, spanning a total of 17 types, are motivated by established analytical approaches in a broad range single-cell literature and can be broadly grouped into six distinct categories including i) cell

type proportions, ii) cell type specific gene expressions, iii) cell type specific pathway expressions, iv) cell type specific CCI scores, v) overall aggregated gene expressions and vi) spatial metrics (Figure 4.1b). These collections of constructed features can then be used for various downstream analysis such as disease outcome prediction, biomarker selection, survival analysis and enable the identification of interpretable features and feature types associated with disease conditions.

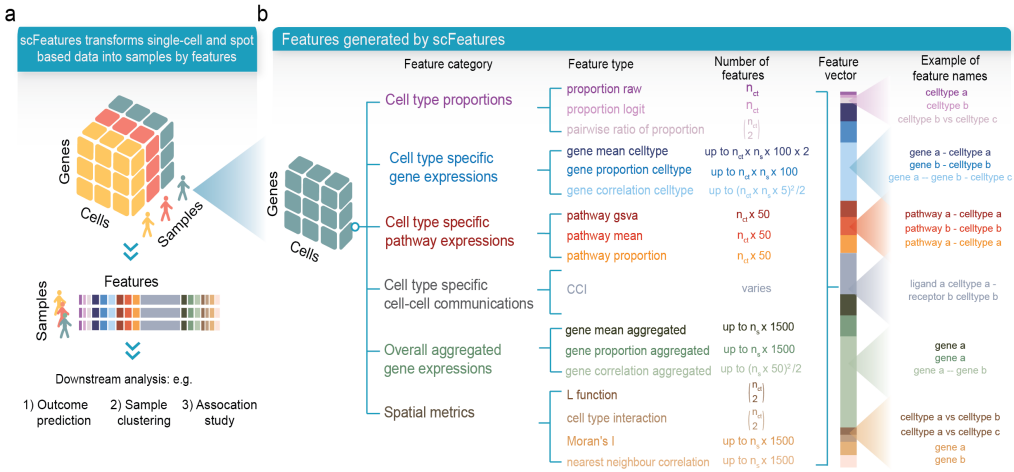


Figure 4.1: Overview of scFeatures. (a) The input for scFeatures is an omics dataset containing multiple samples such as individuals and cell type labels. scFeatures extracts different views of the data, thereby transforming the gene by cell matrix into a vector of features for each sample. (b) scFeatures constructs 17 feature types that can be broadly classified into six categories. Each feature type consists of multiple individual features. For example, for "gene mean celltype", 100 features are generated by default per cell type (n_{ct}) per sample (n_s) (see "scFeatures framework"). Examples of feature names from each feature type are given to illustrate the data format.

The six feature categories represent different "views" of the single-cell information. Specifically, category I captures cell type proportion information in which the proportion of cell types for each sample and the ratio of proportions between two cell types are measured. Category II represents cell type specific gene expression, and examines the expression of sets of genes or proteins in each cell type. We implemented different approaches for representing genes

or proteins measurement, including average expression, proportion of expression and correlation of expressions. In category III, which calculates cell type specific pathway scores, by default the 50 hallmark pathways in the Molecular Signatures Database (MSigDB) (Liberzon *et al.*, 2015; Subramanian *et al.*, 2005) were used to generate various features such as the average expression of each pathway in each cell type. Category IV contains the CCI scores, measuring the probability of ligand-receptor interaction based on the expression values of each sample. Category V is designed to recreate the bulk expression through aggregating the expression across cells or spots depending on the data types. Category VI is designed specifically for spatial data type for capturing spatial information and includes classical metrics for identifying spatial patterns. For all feature categories except category V, the values are summarised at per cell type level, for example, feature x cell type a and feature y cell type b, which then forms the vector of molecular representation containing over 50,000 features for each sample. The implementation details can be found in Table B1.

scFeatures extracts interpretable features from data generated by scRNA-seq, spatial proteomics, and spatial transcriptomics (Table B1). In particular, spatial transcriptomics data, a spot-based technique in which the expression value of each spot is based on a small population of cells, often contains cells from multiple cell types in each spot. We developed several novel ways to adapt the 13 feature types to spot-based data whenever possible; this collection of spatial metrics takes the properties of spot-based technology into consideration and reveals cell type specific features in spot-based data. For example, spot-based data precludes direct application of cell type proportion computation since each spot includes an unknown number of cells while cell type percentage estimation requires individual cell counts for each cell type. To overcome this issue, we estimated the number of cells in each spot using the library size of that location, based on the association between the two values. Table B1 provides more documentation on the implementation details on the adaptation of feature types from single cell RNA-sequencing to spot-based technologies.

4.3.2 *scFeatures generates large collection of diverse features and is scalable to large datasets*

To demonstrate the characteristics of the feature representation, we applied scFeatures to 17 datasets measured using scRNA-seq, spatial proteomics and spatial transcriptomics data (Table B2). For a typical scRNA-seq data, scFeatures generated over 50,000 features (Figure 4.2a). As expected, the number of features generated were mostly associated with the number of cell types in the dataset and not other data characteristics of number of genes and number of cells (Figure B1).

To explore the diversity of the features generated from scFeatures, we first examined the correlation between the features across 17 datasets (Figure 4.2a, Figure B2). By summarising the correlation values between every pairwise combination of feature types (Figure B3), we observed that overall the feature types were poorly correlated, with the median correlation ranging from 0.1 to 0.3 (Figure 4.2b). Hierarchical clustering of the correlations revealed that the higher correlation was observed between certain feature types from the same feature category (Figure 4.2c-d). For example, the "gene mean aggregated" and "gene proportion aggregated" from the aggregated gene expression category had high correlation within each of the feature types and between the feature types pair. This is consistent with our expectation of some degree of co-expression linked with disease conditions.

To further examine the complementarity of the feature types, we compared the performance of individual feature types with the combination of features across feature types (Figure B4). The ability to accurately classify disease outcomes was used as the evaluation metric (see "scFeatures framework"). We found the combination of features in general performed better than most of the individual feature types and achieved the best classification performance in 11 out of the 16 datasets, suggesting the complementarity of the feature types.

We next benchmarked the runtime and memory requirements of the feature types on single-cell scRNA-seq (Figure B5a), spatial proteomics (Figure B5b), as

well as spot-based spatial transcriptomics datasets (Figure B5c) for evaluating both the single-cell RNA-sequencing implementation and the spot-based implementation. All datasets contain 1,000 to 100,000 cells. On the largest datasets with 100,000 cells, the majority of feature types took less than a minute to compute when executed on eight cores, demonstrating that scFeatures is highly scalable to large datasets. As expected, there was some trade-off between processing time and memory. As a result of parallel computation over eight cores, some feature types required more than 10GB of RAM in total; however, users can run on a single core to reduce the memory requirement.

We next benchmarked the runtime and memory requirement of the feature types on single-cell scRNA-seq (Figure B4a), spatial proteomics (Figure B4b), as well as on spot-based spatial transcriptomics datasets (Figure B4c) for evaluating both the single-cell RNA-sequencing implementation and the spot-based implementation. All datasets contain 1,000 to 100,000 cells. On the largest datasets with 100,000 cells, the majority of feature types took less than a minute to compute when executed on eight cores, demonstrating that scFeatures is highly scalable to large datasets. As expected, there was some trade-off between processing time and memory. As a result of parallel computation over eight cores, some feature types required more than 10GB of RAM in total; however, users can run on a single core to decrease the memory required.

4.3.3 *The most informative features classes differ between different datasets*

We hypothesised that distinct feature classes would be informative for different datasets since each dataset comprises samples with varying characteristics and disease outcomes. Several datasets were used where each feature type was evaluated on their ability to predict disease outcomes and the observations are in alignment with our hypothesis. First, we used a lung disease dataset collection where the cells were split into the epithelial, immune and fibroblast subset and the outcome of interest was to classify the individuals into healthy or idiopathic pulmonary fibrosis (IPF). In Figure 4.3a, we visualised the classification

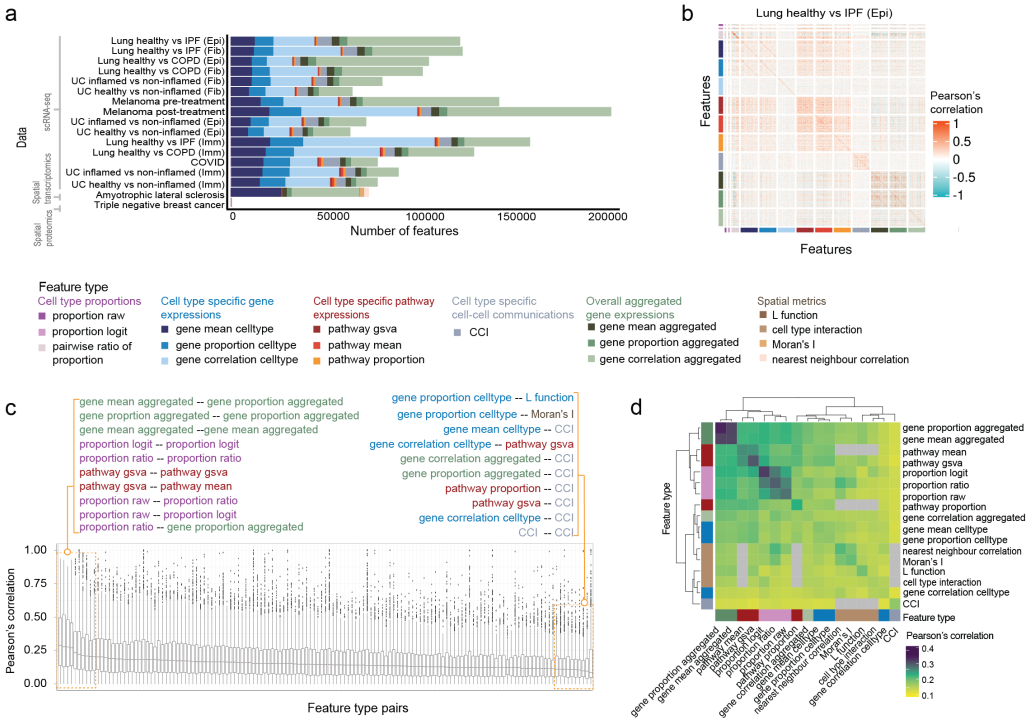


Figure 4.2: Characteristics of the features generated by scFeatures. (a) Compositional barchart showing the number of features generated by scFeatures for each dataset. Datasets are first ordered by data types, and then by the number of cell types. (b) Correlation plot showing Pearson's correlation of features on the "Lung healthy vs IPF (Epi)" dataset as a representative example. The features are colour labelled by feature types for ease of interpretation. (c) Boxplots summarising the correlation between pairs of features across all datasets (see "scFeatures framework"). Texts highlight the 10 most and 10 least correlated feature types pairs, coloured according to their feature category. (d) Hierarchical clustering of the average correlation between feature types. Heatmap is colour labelled by feature category for ease of interpretation.

performance of the feature types on the three subsets and ordered the feature types according to their performance in the epithelial subsets. This reveals that feature types related to cell type proportions (i.e., "proportion ratio", "proportion logit" and "proportion raw") achieved the highest accuracy in the epithelial subset (Figure 4.3a). In contrast, the performance of feature types on the immune and fibroblast subset clearly does not follow the same trend as on the

epithelial subset, demonstrating that different feature types are useful to the three datasets.

Similar observation is also found in the melanoma pre-treatment dataset and melanoma post-treatment dataset where the question of interest is classifying non-responders and responders. Figure 4.3b illustrates that proportion features (i.e., "proportion raw" and "proportion logit") more accurately classified individuals in the post-treatment dataset than in the pre-treatment dataset, while pathway features (i.e, "pathway gsva" and "pathway proportion") provided higher classification accuracy for pre-treated individuals.

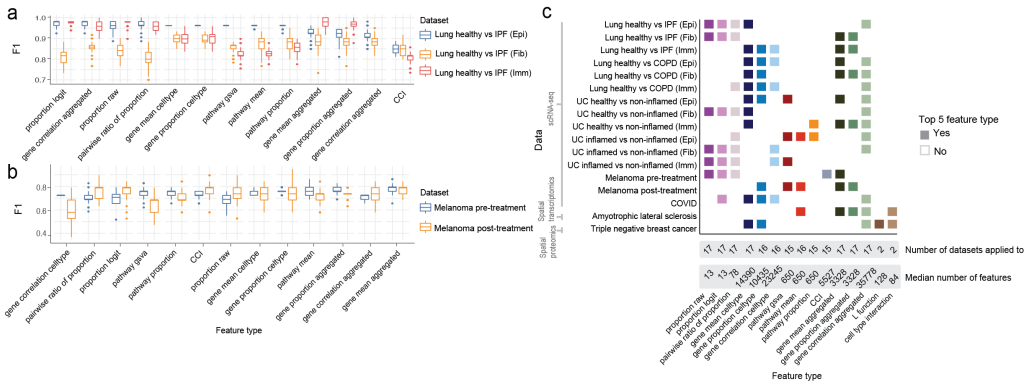


Figure 4.3: Performance of feature types on patient outcomes. (a) shows the epithelial, fibroblast and immune subsets of healthy and IPF individuals, where the outcome of interest is classifying healthy and IPF status. The feature types are ordered by their F1 scores on the epithelial subset. (b) shows pre-treatment and post-treatment melanoma patients, where the outcome of interest is classifying therapy responders and non-responders. The feature types are ordered by the difference of the F1 scores between the two datasets. (c) For each of the 17 datasets, the squares denote the top five feature types with the highest F1 scores.

We then examined across 17 datasets (Figure B6) and highlighted the five informative feature types for each dataset (Figure 4.3c) for a more comprehensive assessment of the performance of the feature types. Across the 17 datasets tested, "gene mean celltype", which examines expression in cell type specific manner, occurred in 10 datasets as the top five informative feature types. This is perhaps not surprising, as it elucidates the power of single-cell technology to profile the

cell type specific expression to uncover changes in response to diseases. Across the spatial datasets, we saw feature types in the spatial feature category appearing as the top five informative feature types, indicating the effectiveness of this category to capture spatial information and the potential of spatial data modality offering complementary information. All together, these findings highlight that different feature types are useful for exploring disease mechanisms in different datasets and even in different subsets of the same dataset, as seen by the pre- and post-treatment melanoma datasets and the lung disease dataset subset by cell types, and argue for the need for a diverse compendium of feature types for such analyses.

4.3.4 *scFeatures provides interpretable insight into disease outcome from scRNA-seq data*

To illustrate that scFeatures provides interpretable features for the understanding of diseases, we applied scFeatures to the "UC healthy vs non - inflamed (Fib)" dataset (Smillie *et al.*, 2019). This scRNA-seq dataset compares fibroblast cells of non-inflamed biopsies from ulcerative colitis (UC) individuals with biopsies from healthy individuals. We focused on the two top performing feature types of "gene mean celltype" and "proportion raw" based on the classification model performance from the previous section (Figure 4.3c) and discovered different sets of cell types were important to the two feature types. In particular, for the feature type based on cell type specific gene expression (denoted by "gene mean celltype"), the fourth-ranked cell type according to feature importance score (see "scFeatures framework") was WNT5B+ 2 (Figure 4.4a). This cell type was ranked as the 11th cell type in terms of the differences in cell type proportion (denoted by "proportion raw") (Figure B7), indicating that while the gene expression was different between disease outcomes, the proportion of cell types was similar. In contrast, glia was ranked first in terms of gene expression and second in terms of cell type proportion. These two feature types offer different perspectives from the same data and reveal distinct collections of cell types where one group is more concerned with changes in expression and the

other collection is more concerned with changes in proportion. It would have been challenging or impossible to accurately disentangle the contributions of cell type percentage and cell specific gene expression in classical bulk gene expression data. These observations not only highlight the necessity of single-cell research, but also emphasize the importance of evaluating various feature types, as generated by scFeatures.

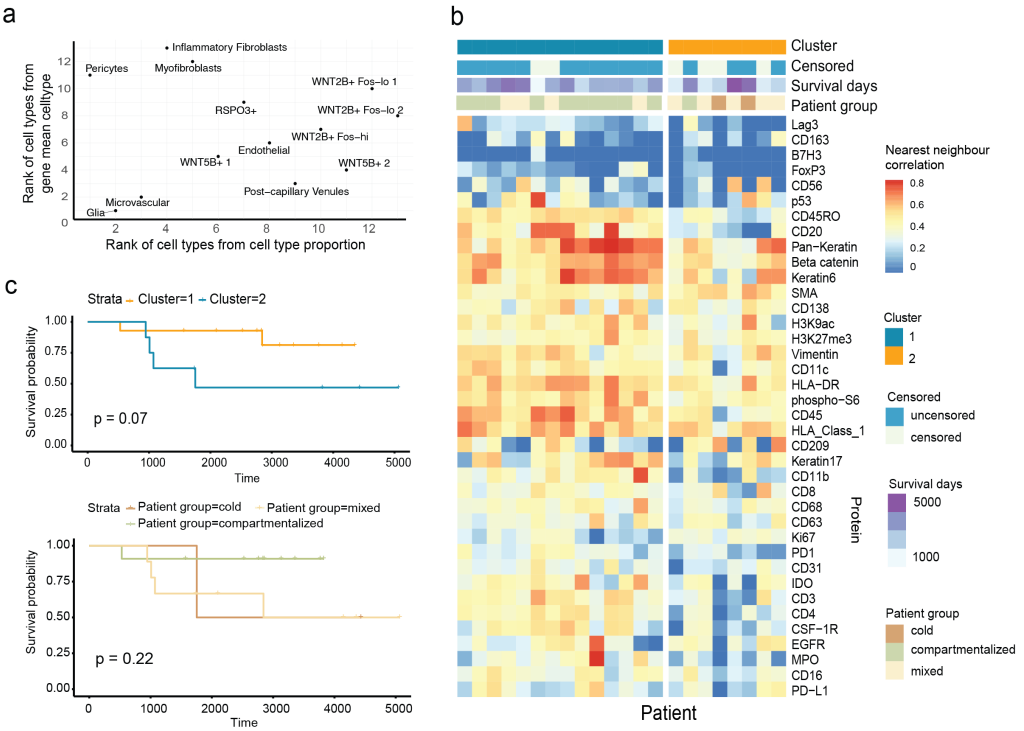


Figure 4.4: Selected features generated on the "UC healthy vs non - inflamed (Fib)" dataset and the "triple negative breast cancer" datasets. (a) Scatterplot of cell type rank for the feature type "cell type proportion" and "gene mean celltype". (b) Heatmap showing the clustering result using the nearest neighbour correlation. (c) Kaplan-Meier plot of individuals stratified by the clustering output (top) and stratified by patient groups defined in the original study (bottom).

4.3.5 *scFeatures uncovers data features associated with survival outcome from spatial proteomics*

To demonstrate the utility of scFeatures at extracting spatial information, we applied scFeatures to a spatial proteomics dataset of tumours from triple negative breast cancer individuals. The question of interest is classifying tumours based on cellular organisation into distinct types that are associated with patient survival. The original study defined three tumour groups based on mixing scores, where a "cold group" is identified by low immune infiltrate, a "compartmentalised group" is identified by compartments formed by almost entirely of either tumour or immune cells, and a "mixed group" is when there is no clear boundary separating the tumour and immune cells.

The nearest neighbour correlation is a feature type in scFeatures that was created primarily to capture spatial co-expression patterns. It computes the correlation of a cell's protein expression with that of its nearest neighbour. Therefore, spatial organisation of cells, such as whether tumour cells are next to immune cells would affect the correlation of protein expression of cells with neighbouring cells. To construct this feature type, we used scFeatures on selected "triple negative breast cancer" samples from the dataset and clustered the resulting features (Figure 4.4b). Survival analysis using the Kaplan-Meier Curve revealed differences between survival outcomes of individuals from the two clusters (P-value of 0.07, Figure 4.4c), compared to the patient group defined in the original study with P-value of 0.22. This suggests that the new patient subgroup found by scFeatures has greater association with the survival outcomes and demonstrates the ability of the spatial feature category at representing spatial organisations and uncovering novel patterns in the data.

4.3.6 *scFeatures automatically generates an HTML file that report features most associated with conditions to facilitate interpretable discoveries*

One of the commonly investigated questions by researchers is what features are most associated with disease conditions. scFeatures implemented a function that takes generated features as input and automatically performs a series of association studies for each feature type, producing an HTML report as the output. An example of a comphrensivie HTML report can be found in our Github (`https://github.com/SydneyBioX/scFeatures`). The HTML report includes a variety of visual summaries to aid the downstream interpretations of features (Supplementary notes B3). Here, we used the "COVID" dataset to identify featuress associated with disease severity and illustrate a selected panel of visual summaries (Figure 4.5). The composition plot visualised the features from "cell type proportion raw" (Figure 4.5a) and revealed that many cell types underwent drastic change between mild and severe conditions. The pathway enrichment plots (Figure 4.5b) summarised that, in the rare cell type plasmablasts, genes associated with severe condition were enriched in immune pathways. Heatmap is used to visualise difference between conditions that can be expressed numerically. The heatmap on feature type "CCI" revealed that the cell cell interactions in most pairs of cell types increased in severe patients compared to mild patients (Figure 4.5c). Overall, the association study and visual summaries provided by the HTML facilitates a more focused exploration of features for further analysis.

4.4 DISCUSSION

In summary, scFeatures creates a multi-view molecular representation of individuals by generating over tens of thousands of interpretable features based on single-cell and spot-based spatial data. The innovation and motivation of scFeatures lie in the generation of various literature motivated and biologically relevant feature vectors for phenotype disease modelling and disease prediction. We have designed 17 feature types across six categories based on a broad range

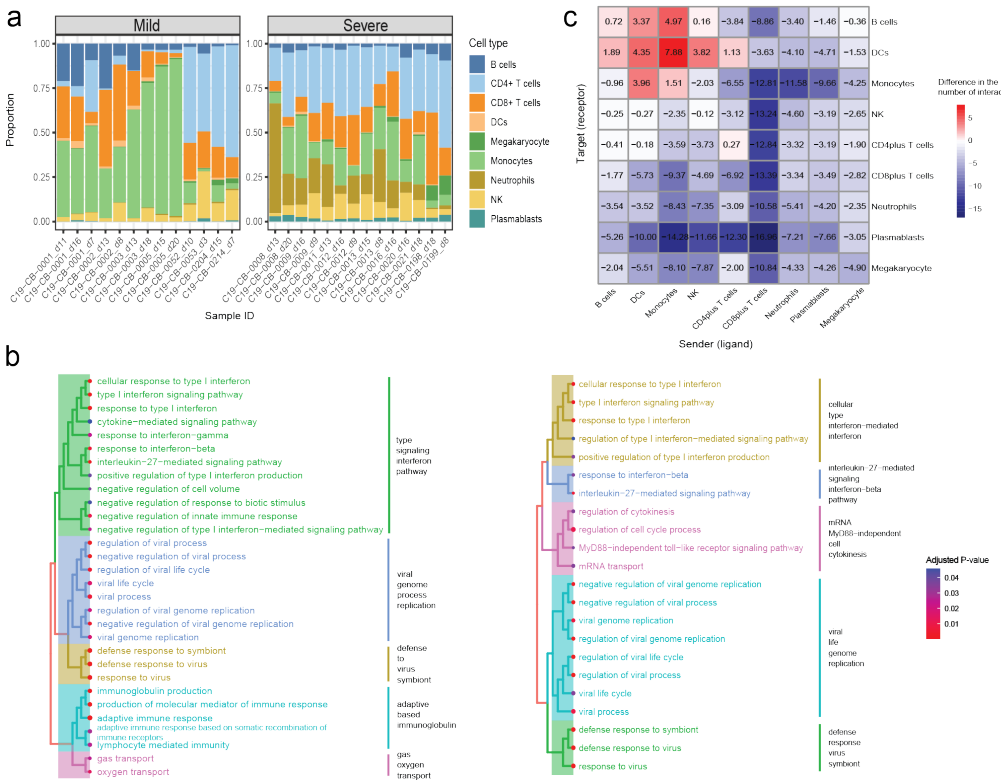


Figure 4.5: Selected visualisation summaries from the HTML report. The "COVID" dataset containing mild and severe COVID-19 patients was used to show a subset of visualisation summaries from the association analysis report. (a) Composition plot of the "cell type proportion raw" features in mild and severe patients. (b) Enriched pathways of the top 200 features associated with the plasmoblasts of severe patients. Pathway enrichment in the left plot was calculated based on features from "cell type specific mean expression". Pathway enrichment in the right plot was calculated based on features from "cell type specific mean proportion". Similar pathway terms were grouped by hierarchical clustering. (c) Heatmap shows the difference in the number of CCI features between mild and severe patients. Positive number indicates more interactions in the mild patients and negative number indicates more interactions in the severe patients.

of analytical approaches in literature from cell type specific gene expression to measures of cell-cell (ligand-receptor co-expression) interaction and demonstrated that the feature types are diverse with low correlation amongst them. We illustrated scFeatures on scRNA-seq data from ulcerative colitis and discovered a number of features linked with disease characteristics. scFeatures is also

able to extract spatial features from a triple negative breast cancer proteomics data, resulting in the stratification of tumours that are more strongly related to survival outcomes than the original study's subgroups. Through the automatic report generation that highlight features most associated with disease, scFeatures support ease of feature exploration.

The features vector generated by scFeatures can be used for a broader set of downstream applications and is not limited to the ones illustrated in the case studies. For example, given the features vectors are generated at the sample level, this provides the opportunity for the exploration of differential patient response to diseases due to heterogeneity between individuals. Even for individuals recorded as responders to treatment, the extent of response and the change at omics level varies between individuals. The feature vector can be subjected to latent class analysis, which has typically been applied on single-cell level for exploring cellular diversity (Cheng *et al.*, 2019; Buettner *et al.*, 2017), to enable detection of sub-populations in the cohort, as well as the biology driving patient heterogeneity. Given that scFeatures creates a representation for each patient, this also enables the integrative analysis of patients across multiple datasets to increase the power of analysis and to expand the range of questions that can be asked. Batch correction methods, such as scMerge (Lin *et al.*, 2019) and Harmony (Korsunsky *et al.*, 2019), may be needed in this case to remove the unwanted technical variation due to datasets.

The multiple feature representations generated by scFeatures can be considered as multiple views of the data and as such leads naturally to multi-view learning. This is one of the many collections of methods that perform integration across multiple features classes to enhance model performance. There exists a number of approaches for data integration (Li *et al.*, 2018), from the simple concatenation of features from all feature types into a single vector as the input, to incorporating and optimising the procedure within the model training process. While current multi-view learning in bioinformatics typically refers to the use of multiple omics obtained from the same sample (Nguyen and Wang, 2020) ,

we envisage the generation of multiple features types by scFeatures opens new opportunities for multi-view learning for single omic type.

scFeatures is currently designed to perform feature engineering for single-cell RNA-seq, spatial proteomics and spatial transcriptomics data, but the framework is not limited to these platforms. Taking chromatin accessibility as an example, a commonly used analysis strategy is assigning genes based on nearby peaks, thereby converting the peak matrix to a matrix of gene activity scores similar to gene expressions (Baek and Lee, 2020). Using this approach, all feature classes designed for scRNA-seq are then applicable to chromatin accessibility data. In future, we plan to extend scFeatures to other single-cell omics such as single-cell DNA methylation, single-cell chromatin accessibility and single-cell genomics, leveraging the common analytical approach in these omics and constructing specific feature classes. For chromatin accessibility, the co-accessibility between pairs of peaks, which is used to predict cis-regulatory interactions, can be constructed and stored as a vector for each sample. The correlation values between transcription factors (TF) motifs can be readily constructed as another class of feature representation vector, and can be used to identify the modules of TF motifs affected in disease state.

With the recent surge of cohort based single-cell studies and the number of tools for characterising individual cells, there is an increased demand for defining samples in a study based on their cellular characterization to guide better understanding of disease and health. Here, we present scFeatures, a tool that provides a multi-view extraction of molecular features from single-cell and spot-based spatial data to characterise cellular features of each individual. scFeatures efficiently extracts collections of interpretable features from large-scale data and derives biological insights in both scRNA-seq and spatial data. We envision that scFeatures, a public R package available at `https://github.com/SydneyBioX/scFeatures`, will facilitate better understanding of single-cell data from a sample (i.e. patient) perspective and the signatures underlying disease conditions from different angles.

Table 1: List of features generated by scFeatures.

Application	Data characteristic and representative technology	Feature category	Cell type specific proportions			Cell type specific gene expressions			Cell type specific pathway expressions			Cell-cell interaction scores	Overall aggregated gene expressions			Spatial metrics			
		Feature type	Proportion raw	Proportion logit	Ratio	Mean	Proportion	Correlation	Mean	GSVA	Proportion	CCI	Mean	Proportion	Correlation	L function	Cell type interaction	Moran's I	Nearest neighbour correlation
scRNA-seq data	Single-cell based Eg. 10x, Smart-Seq		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Spatial proteomics data	Single-cell based Eg. MIBI-TOF, IMC, CODEX		✓	✓	✓	✓	✓	✓					✓	✓	✓	✓	✓	✓	✓
Single cell transcriptome data	Spot based Eg. ST, Visium, MALDI		✓	✓	✓	✓			✓				✓	✓	✓	✓	✓	✓	✓

5

TOWARDS A BENCHMARKING STUDY OF ENSEMBLE DEEP LEARNING AND SCFEATURES WITH COVID-19 DATASETS

5.1 INTRODUCTION

Since the outbreak of COVID-19 at the start of 2020, the world has been thrown into chaos. Two and a half years later, at the time of this thesis, new strains are still fast-evolving, impacting the global population (Cao *et al.*, 2022a). Many research efforts globally in all disciplines have been initiated to respond to this threat and on this front, single-cell technology is one of the technologies actively used to understand the molecular mechanism behind the differential response of COVID-19 (Garg *et al.*, 2021).

One of the challenges associated with the exploration of COVID-19 single-cell data is the analytical choices for such large-scale multi-sample and multi-condition cohort studies. Currently, there is little consensus on the best ways to compress information from the complex single-cell data structures to summary statistics that represent each sample and to apply machine learning techniques for downstream analysis. In the previous chapters of the thesis, we developed a number of frameworks towards precision medicine in the single-cell field. In this chapter, we further use these frameworks to illustrate a framework towards benchmarking data workflow for handling multi-sample and multi-condition cohort data using a collection of single-cell RNA-seq COVID-19 datasets. In particular, we construct molecular representations of each patient sample using scFeatures developed in Chapter 4 and implement different learning models from classical machine learning to the more recent deep learning model, as

well as different ensemble strategies surveyed in Chapter 2. Finally, through a comparison framework (Chapter 3), we evaluate the combined impact of these key data analytical steps (i.e., model choices, ensemble learning strategy and integration strategy when using multiple datasets as the input) in disease outcome prediction for COVID-19 patients.

5.2 DESIGNING A COMPARISON FRAMEWORK

As highlighted in detail in Chapter 3, any benchmarking or comparison study involves three key elements: (1) a collection of datasets that are used for the evaluation, (2) evaluation strategy for comparing the methods and (3) evaluation metric that quantify the performance. These are described in detail in the following subsections.

5.2.1 *Evaluation datasets collection*

As the globe has been heavily impacted by COVID-19 for nearly three years, COVID-19 data is perhaps one of the largest collections of multi-sample multi-condition single-cell datasets. Therefore, to examine the data analytics strategy for cohort analysis, we used five publicly available COVID-19 datasets containing individuals with mild and severe disease progression (Figure 5.1a). All datasets sequenced the peripheral blood mononuclear cells (PBMC) or whole blood using scRNA-seq technology. The details of the datasets such as the number of individuals are provided in Table 1.

5.2.2 *Evaluation strategies*

The comparison study aims to examine the impact of various analytical strategies in performing patient disease outcome prediction. The evaluation is composed of three key steps, (1) generating features to represent patients, (2) using features as input into learning models and (3) comparing single-view and multi-

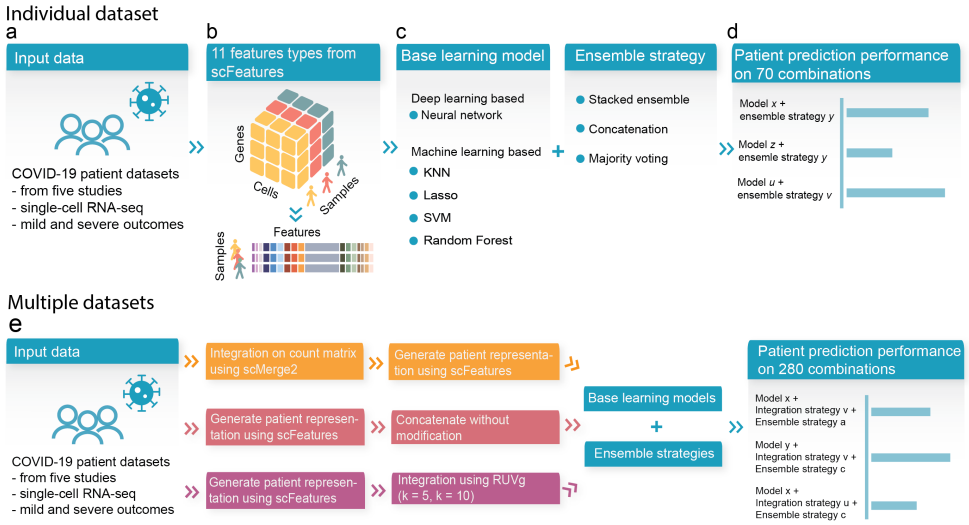


Figure 5.1: Schematic of the benchmark workflow. (a) Five COVID-19 scRNA-seq datasets containing mild and severe outcome patients were used in this benchmark study. (b) We used scFeatures to generate 11 types of molecular representations of each individuals (i.e., the patients). (c) We implemented five models containing both deep learning and machine learning, as well as three ensemble strategies. (d) The analytical strategies resulted in a total of 70 combinations for evaluating patient outcome prediction in each individual dataset. (e) We also evaluated the performance of analytical strategies on the combined dataset. To combine the dataset, we implemented three integration strategies. We used the same base learning models and ensemble strategies as shown in c. This resulted in another 280 combinations.

Table 1: Additional details of the scRNA-seq simulation methods evaluated in this study.

Dataset name	Reference	Accession ID	Number of mild individuals	Number of severe individuals	Number of mild and severe individuals	Number of cells in mild and severe individuals
Combat	Ahern *et al.* (2022)	EGAS00001005493	30	61	91	524,557
Ren	Ren *et al.* (2021)	GSE158055	68	87	155	872,663
Schulte-schrepping	Schulte-Schrepping *et al.* (2020b)	EGAS00001004571	44	51	95	212,023
Stephenson	Stephenson *et al.* (2021)	E-MTAB-10026	58	32	90	493,685
Wilk	Wilk *et al.* (2021)	GSE174072	23	19	42	112, 589
Total			223	250	473	2,215,517

view features. In step 1, we use our recently developed feature engineering tool, scFeatures, to generate multi-view molecular representation of each individual that can be used as input into downstream analytical models (Figure 5.1b). In step 2, we survey and implement multiple learning platforms from classical machine learning to modern deep learning methods (Figure 5.1c). In step 3, we examine the difference in performance between using single-view feature space versus multi-view feature space via implementing multiple ensemble strategies. In summary, we examine a total of 70 workflow combinations from 11 feature representations, five base models and three ensemble strategies (Table 2). Each of these steps is described in more detail in the below subsections.

Table 2: Summary of the evaluation strategies.

Analytical step	Analytical choice
Feature representation	(1) Proportion ratio, (2) Proportion raw, (3) Proportion logit, (4) Gene mean celltype, (5) Gene proportion celltype, (6) Pathway gsva, (7) Pathway mean, (8) Pathway proportion, (9) CCI, (10) Gene mean aggregated, (11) Gene proportion aggregated
Base learning model	(1) KNN, (2) Lasso, (3) Random Forest, (4) SVM, (5) Neural network
Ensemble strategy	(1) Concatenation, (2) Majority voting, (3) Stacking
Level of integration	(A) Cell level integration, (B) Individual level integration with no normalisation, (C1) Individual level integration with RUVg adjustment (using K = 5), (C2) Individual level integration with RUVg adjustment (using K = 10)

5.2.2.1 *Feature generation*

We used scFeatures as described in Chapter 4 to generate the molecular representation for each individual in each of the COVID-19 datasets. A total of 11 feature types from five feature categories were generated to reflect different views of the molecular property and were used for downstream analysis. For example, the feature types "proportion ratio", "proportion raw" and "proportion logit" represent cell type proportions in a patient. We also included features representing gene expression, pathway-level information and cell-cell interaction (Table 2). Details regarding each of the feature types were discussed in Chapter 4.

5.2.2.2 *Base model implementation*

To examine the effect of learning models on patient prediction, we implemented a selection of classical machine learning approaches and the more recent deep

learning approach. These models served as the base models to model from the training data set and evaluate on the testing data set.

For **classical machine learning approach**, we included a range of models including KNN, Lasso, Random Forest and SVM with linear kernel using the implementation in the Caret R package (Kuhn, 2008) version 6.0-93. Each feature type was used individually as the input to compare the performance of each feature type. The severity (mild and severe) of the patients' conditions was used as the outcome variable. For Lasso which outputs the prediction in terms of probability instead of discrete outcome, we used 0.5 as the threshold.

For the **deep learning approach**, we implemented a neural network structure containing four fully connected layers. For each feature type, we used the same network structure but varied the number of nodes in the layers depending on the number of features in the feature type. In detail, the input layer had a number of nodes equal to the number of features in the respective feature type. The second layer and third contained different numbers of nodes depending on the feature types. We describe the detailed implementation below:

- All feature types in the category "cell type proportions" contained less than 100 features. For these feature types, we set both the first layer and second layer to 20 nodes.

- All feature types in the category "cell type specific pathway expressions", "overall aggregated gene expressions" and "cell-cell communications" contained less than 1000 features. For these feature types, we set the first layer to 500 nodes and the second layer to 100 nodes to reduce the dimension.

- All feature types in "cell type specific gene expression" contained less than 10000 features. To reduce the dimensions for these feature types, we set the first layer to 1000 nodes and the second layer to 100 nodes.

  These node settings were determined using a basic parameter search. The number of nodes specified above was considered as "baseline". We then explored four settings of: increasing the numbers of nodes in both layers, increasing the number of nodes in one layer, decreasing the number of

nodes both layer, decreasing the number of nodes in one layer. Details on the number of nodes under each setting are outlined in Table C1. Given we observe insignificant differences between the prediction accuracy of these five settings (Figure C2), we chose the "baseline" node settings.

The number of nodes in the output layer was the same for all feature types, with two nodes that output the probability of mild and severe conditions, respectively. The condition with higher probability was considered the predicted condition.

5.2.2.3 *Ensemble strategy*

scFeatures generates multiple feature types for a given patient, representing different biological views. It is therefore of interest to combine the feature types into "multi-view" representation and examine the impact on model performance compared to using each of the feature types individual as "single-view" representation. Here, we employed three types of ensemble strategies to obtain a "multi-view" representation. The implementation of these strategies is described in the following:

- Concatenation involved combining the features across all feature types and using the concatenated result as the input. The implementation was the same for both machine learning and deep learning models.
- In majority voting, we first obtained the predicted outcome from each of the 11 feature types, resulting in 11 predictions of either mild or severe for each patient. Then the outcome with the most votes was considered to be the final predicted outcome for the patient. The implementation was the same for both machine learning and deep learning models.
- Stacked ensemble involved a two step process. First, base learners were trained on the feature space, this was then followed by a meta-learner that was trained to best combine the individual base learners. The implementation was different for machine learning and deep learning models:

- For machine learning models, base learners were trained and evaluated on each of the individual feature types, resulting in 11 predictions for each patient. The predictions were then used as the input to build a logistic regression model. The logistic regression model served as the meta-learner that combined the base learners and produced the final predicted outcome.

- For deep learning models, we implemented a network (Figure 5.1) containing 11 subnetworks that took each of the 11 feature types as input. The subnetwork performed feature extraction for each of the feature types individually. We used the same network structure as the network described in the previous section that was used for extracting features from each feature type individually. The extracted features from each feature type were then concatenated, resulting in a vector of 860 features for each individual. This feature vector was then passed through a subsequent fully connected layer containing 50 nodes, followed by the output layer containing two nodes to produce the final prediction.

5.2.2.4 *Integration strategy*

We examine different levels of integration. integration approaches to examine the optimal choice for predicting patient states when multiple datasets need to be combined and used as a whole in building a prediction model. The approaches are described in the following:

- Cell level integration - this approach refers to integration on count matrix: We used scMerge2 (personal communication) to perform data integration on the scRNA-seq count matrix. We then generated the patient representation using scFeatures on the integrated count matrix and used this as input for learning model.

- Individual level integration with no adjustment: We simply concatenated the patient representation without any adjustment or normalization, and used this as input for learning model.

- Integration on patient representations: We used a well-known batch correction method RUVg (Risso *et al.*, 2014) to correct for the batch effect in the patient representation. As k, the number of unwanted variations is a tunable parameter, we explored two settings of k = 5 (i.e., where the number of batches is equal to the number of datasets) and k = 10 (i.e., to introduce a stronger batch correction). The batch-corrected patient representation was used as input for learning model.

5.2.3 *Evaluation metric*

5.2.3.1 *Accuracy metric*

To quantify the performance of the methods, we recorded the prediction accuracy of the severity outcome (Figure 5.1e). To capture the variability in model performance, all classical machine learning and deep learning models were trained and tested with repeated three folds cross-validation using 20 repeats. To control for the potential impact of "good" or "bad" training/testing set splits, where a "bad" split can result in extreme class imbalance in the modelling phase and affect model performance, we used the same training and testing splitting index across all machine learning and deep learning model to ensure a fair comparison. F1 score was used as the evaluation metric, as not all datasets are balanced.

5.2.3.2 *Aggregation of accuracy metric*

Given the number of results from all analytical combinations, we aggregated the results in order to better quantify and interpret the results. First, we took the median F1 score across the 20 repeated cross-validation. This was then followed by different aggregation strategies depending on whether the input used individual or combined datasets.

For the result section where we dealt with the five datasets individually, we further aggregated the median F1 score across datasets by taking the median. Then, we ranked the feature types across each model choice as well as the

model choice across each feature type to derive the ranking of feature types and the ranking of model choice.

5.2.3.3 *Computational resource metric*

Apart from assessing the performance in terms of accuracy, we also assess the performance in terms of the computational resources. This was measured through running time and memory usage averaged over three repeats. All processes were executed using a research server with dual Intel(R) Xeon(R) Gold 6148 Processor with 40 cores, 768 GB of memory and two NVIDIA GeForce RTX 2080 Ti graphics cards.

Running time of each combination was measured using the Sys.time function built in R and the time.time function built in Python. Memory usage was quantified in terms of CPU memory for combinations involving machine learning models. For combinations involving deep learning models, the memory usage was quantified as the sum of CPU and GPU memory, as the deep learning models were executed on GPU.

5.3 RESULTS AND DISCUSSION

5.3.1 *Ensemble strategy improves model performance*

scFeatures generates a wide range of feature types. To explore whether ensemble learning using the combination of feature types can improve performance on downstream analysis, we implemented three common ensemble strategies and ranked their performance against each of the individual features. Performance was evaluated by their prediction accuracy on five COVID-19 patient outcomes (Table 1).

We observed that consistent across the four machine learning models, majority voting consistently achieved the best performance, better than the other two ensemble strategies, as well as better than all individual features (Figure 5.2). This is followed by concatenation, which was also better than using any of the indi-

vidual features. These results highlight the effectiveness of ensemble learning and also suggest that the feature types meet the important criteria of diversity as discussed in Chapter 2, such that different feature types make different errors and combining them results in improvement in classification model performance. Further examination of the top eight learning model and feature type combinations (Figure 5.3) revealed that seven of the eight combinations involves ensemble learning. Interestingly, the more complicated implementation of ensemble learning called stacked ensemble, in which a meta-learner is trained on the base learners trained from individual feature types, performed worse than using any of the individual feature types except for when deep learning was used.

We then took a closer examination at whether this observation is consistent irrespective of the learning model choice or dataset. We ranked the feature types on each of the five types of models and each of the five datasets. We observed that no individual feature type consistently ranked better or worse than others across all models and datasets (Figure 5.4). Almost all individual feature types had ranks that varied from 1 (the best rank) to 14 (the worst rank). This suggests that different feature types are useful for different models and different datasets, despite them all being COVID-19 datasets with mild and severe individuals. In contrast, majority voting achieved a rank of 1 across multiple models and multiple datasets, again illustrating the power of ensemble strategy.

5.3.2 *Deep Learning performs similarly to classical machine learning*

Ranking the learning methods, we noted that there was no clear difference between deep learning and some of the machine learning models (Figure 5.5a). In particular, both neural network and random forest achieved a median rank of 1.75 out of the five learning methods across the 14 feature types and five datasets, followed closely by SVM with a median rank of 2.5 (Figure 5.5b). Only lasso and KNN were consistently ranked lower than other methods. Within neu-

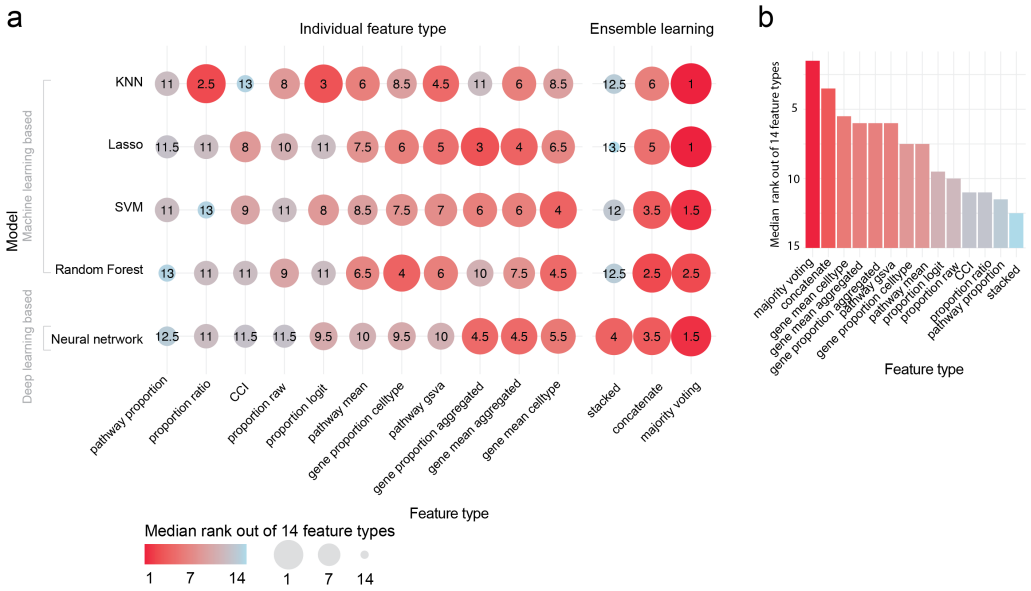


Figure 5.2: Performance of feature types for each model summarised across all datasets. The dotplot shows the relative rank of each feature type to each other for each model, with 1 being the best and 14 being the worst. Ranks are summarised across the five datasets using the median and therefore do not necessarily range from 1 to 14 within each model. Both the colour and circle size denote the median rank of the given feature type.

ral network, random forest and SVM, we then examined the difference between the maximum and minimum F1 score achieved by the three top-performing methods and observed a small median difference of 0.02 (Figure C3). These result all suggest that deep learning do not significantly outperform certain machine learning models in this context.

We then compared the computational resource requirement to see whether the difference in performance came at a cost. Focusing on the feature type "majority voting", we observed that both neural network and random forest took around 4 hours on the largest Ren et al. dataset with 153 patients (Figure **??**, C5). On the other hand, while SVM was ranked after neural network and random forest, it was more computationally efficient, taking less than 1 hour on the Ren et al. dataset. Accounting for the significant difference in computational efficiency and the relatively small difference between model performance, one may consider SVM to be the optimal choice.

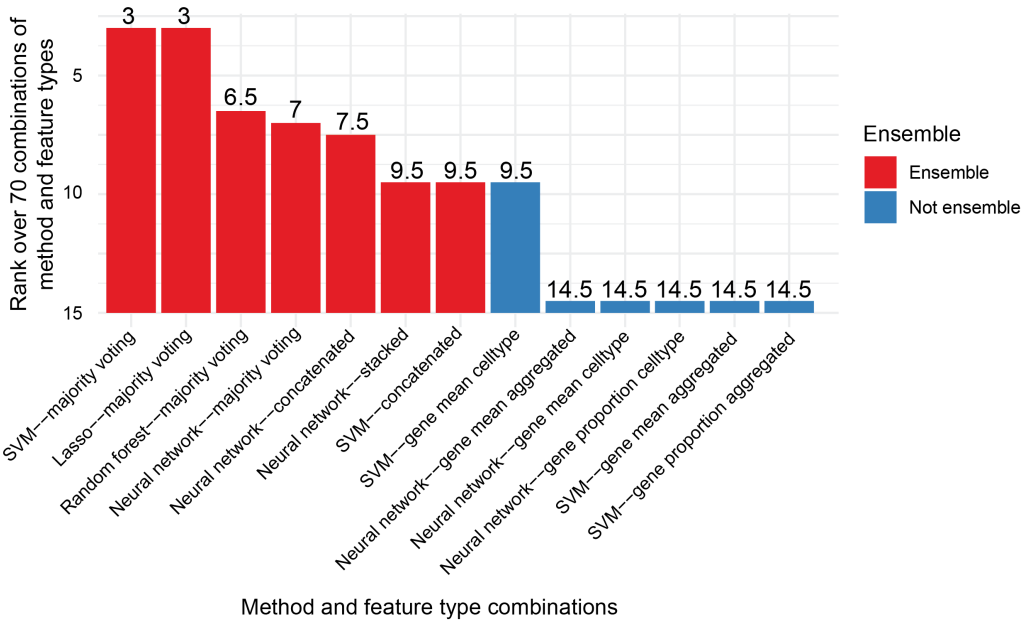


Figure 5.3: Performance of feature types for each model and each dataset. Dotplot shows the rank of each feature type to each other for each model and each dataset. A total of 25 points are shown for each feature type, as each feature type was evaluated on five models and five datasets.

5.3.3 *Normalisation is not necessary when combining multiple datasets as the input*

Using multiple datasets as input data raises a number of questions, such as whether to integrate the raw data or the derived features. Here, we explored three categories of analytical combinations. More specially, different approaches to data integration, including integration on the count matrix, integration on the patient representation with and without normalization. Our results are based on examination of 280 analytical combinations (4 integration types x 14 feature types [11 individual feature types with 3 ensemble feature types] x 5 model choices). Interestingly, there was only a slight difference between integration on the count matrix and concatenation without modification (Figure 5.6), which both achieved high F1 scores. On the other hand, integration on the patient representations achieved lower F1 scores, with the stronger the batch removal setting, the worse the F1 score. This observation is consistent across the choice of method and the type of feature used (Figure C6,C7).

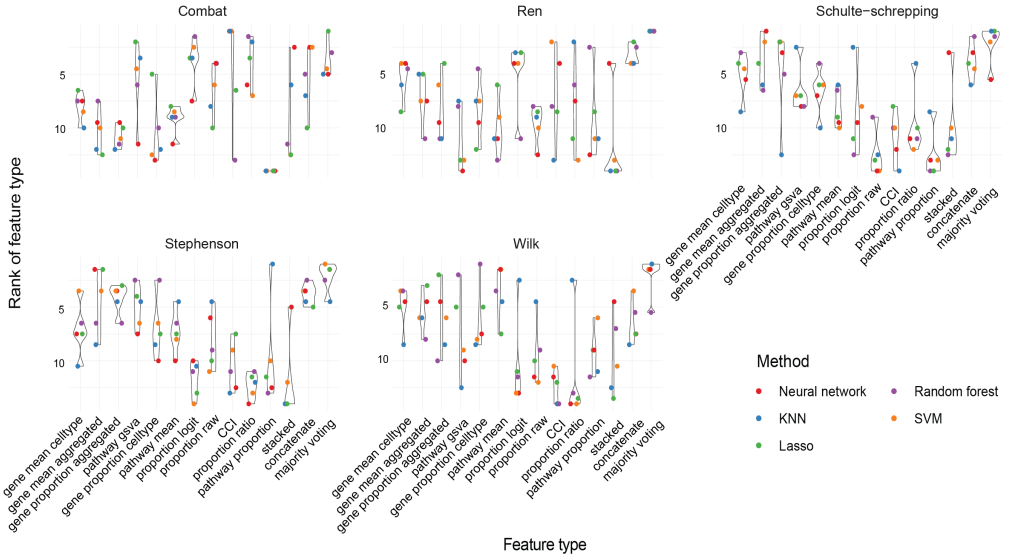


Figure 5.4: Performance of models. (a) shows the relative rank of each model to each other for each feature type with 1 being the best and 5 being the worst. Ranks are summarised across the five datasets using the median and therefore do not necessarily range from 1 to 5 within each feature type. (b) further summarise the ranks of each model across all feature types using the median. Both the colour and circle size denote the median rank of the given model.

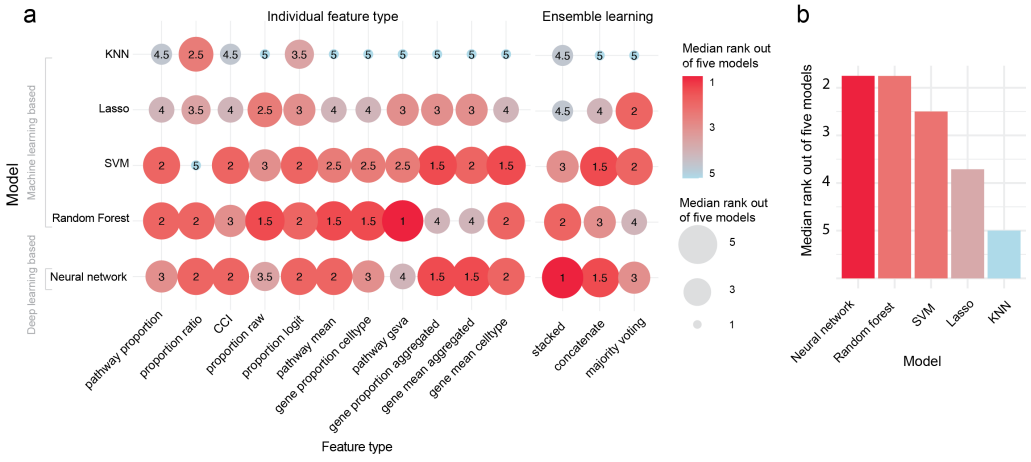


Figure 5.5: Top 13 combinations of model and feature type. Barplot shows the ranks of model and feature type. Given that the ranks are summarised across all five datasets using the median, the values do not necessarily range from 1 to 13.

One of the key strengths of data integration is the ability to examine condition associated features for a subgroup of individuals. Due to the small number of

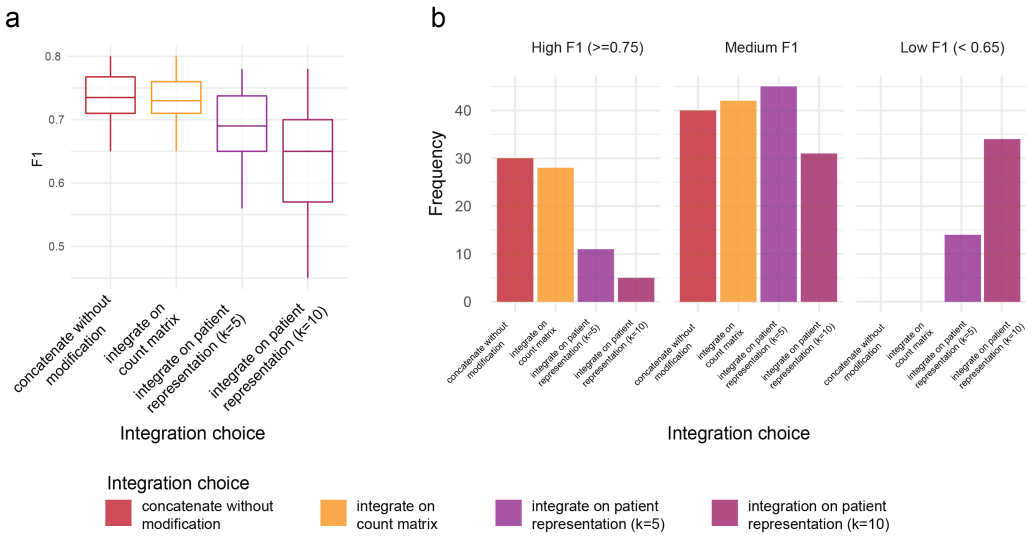


Figure 5.6: Performance of various approaches on combining multiple datasets for building prediction model. a shows the F1 of these 280 analytical combinations, with the x-axis indicating the type of integration choice used in the combinations. b further stratifies the F1 score based on high F1 (defined to be F1 >= 0.75), medium F1 score (defined to be 0.65 < F1 < 0.75) and low F1 score (defined to be F1 <=0.65) and examines the proportion of each integration choice in the set of combinations that fall in the stratification.

individuals that typically fall into the subgroup of interest, this type of research is difficult to conduct using a single dataset. Here, we focused on a subgroup of patients in the 41-50 age group and investigated whether the identification of features are affected by different data integration strategy. First, we compared the rankings of the features obtained according to the feature importance score from the prediction model and found high consistency of the rankings between cell level integration and individual level integration without normalisation (Figure C8a). In comparison, the consistency was much lower between cell level integration and individual level integration with normalisation. Clustering and dimension reduction on the features revealed that in both cases the clustering patterns and sources of variation of the patients were not driven by the dataset source (Figure C8b,c). The lack of batch effect in the generated features therefore potentially suggest that the generated features have self-adjusted in the feature extraction procedure and explains the minimal difference observed be-

tween the feature rankings and suggest for no need of normalisation on cell level or on individual level.

5.4 SUMMARY

In this chapter we illustrate how the various work developed in our thesis enable an effective design for a comparison study for data analysis.We used scFeatures to generate various feature representations for COVID-19 patients and examined the performance of individual feature types and ensemble feature types in classifying COVID-19 severity. By evaluating using multiple datasets and multiple learning methods from classical machine learning to modern deep learning methods, this study demonstrated that all machine learning methods perform similarity, with SVM being a slightly better method when accounting for the computational efficiency. Through implementing different ensemble strategies to incorporate multiple feature types as input into machine learning models, we revealed certain ensemble strategies, in particular majority voting, consistently led to increased performance compared to the non-ensemble strategy of using individual feature types alone. Stacked ensemble for example, often did not achieve better performance compared to using individual feature types. Finally, we suggest that when combining datasets is needed for building a prediction model, prior data integration is not necessary in terms of improving prediction performance. On the other hand, normalisation on the derived patient features decrease prediction performance.

We observed that with the sets of COVID-19 datasets containing 42 to 153 patients, which is a realistic samples size in the current literature, the more complex approaches do not necessarily outperform than simpler approaches. In particular, stacked ensemble can be considered the most complex implementation as it trains additional meta-learner on top of the base models. We observed that while the other two implementations (majority voting and concatenation) both performed better than individual features, stacked ensembled had worse performance compared to using the individual features. We further observed

that when combining datasets is needed as input, there was minimal improvement obtained by cell level integration before generating sample representation. In this case study, although we used five datasets only, it is to be noted that the total number of cells in these datasets reached more than two million. A recent benchmarking study on single-cell integration methods revealed that the majority of existing integration methods took from a few hours to days and even weeks on integrating one million cells (Luecken *et al.*, 2022). Therefore it may not be worth the time and computational resources to perform integration on such large-scale datasets.

This chapter shares a similar concept to Chapter 3, where we developed an evaluation framework SimBench and used it to evaluate the performance of scRNA-seq data simulation methods. Using this concept, we can extend this comparison study into a more comprehensive benchmarking study that incorporates the impact of various other factors. For example, as we curate more COVID-19 datasets, one can further assess the impact of sample sizes. Such systematical benchmarking study would involve using a diverse collection of datasets, the construction of multiple aspects of evaluation criteria and the development of R packages and shiny web application that enable the community to apply the framework to their own data and methods. We envisage the current comparison study and the future study will point direction to an optimised analytical workflow for disease outcome prediction using single-cell data.

6

CONCLUSION

This thesis explores and addresses a number of challenges in the development of bioinformatics approaches for data analysis to facilitate precision medicine in the era of single-cell technology.

Firstly, we discussed the success of deep learning in computer science and highlighted a number of recent bioinformatics applications that benefit from deep learning coupled with ensemble learning (Chapter 2). Our summary of recent ensemble deep learning network structures acts as a resource to inspire future applications of ensemble deep learning for precision medicine that synergistically addresses the challenges of model stability and model scalability.

Secondly, we addressed the lack of comprehensive evaluation studies for scRNA-seq data simulation (Chapter 3). We curated a collection of benchmarking datasets for evaluation of single-cell tools and developed a novel evaluation framework, SimBench, and used these to evaluate current simulation tools. This work enables researchers in precision medicine to efficiently select the method best suited for their research questions and developers to readily identify the areas of limitation of existing methods for precision medicine applications. Moreover, our benchmarking datasets and evaluation framework, both made publicly available to the scientific community, serve to inspire future evaluation studies on precision medicine tools. Importantly, SimBench is available as a living benchmark, since the accompanying web application is actively updated as new tools are published and can also be updated from the research community via GitHub pull requests. After the publication of the study, we have, at the time of writing this thesis, updated the website to include an additional 6 sim-

ulation tools to a total of 19 tools. This living benchmark extends beyond the limitations of traditional publication and stays relevant for future applications.

Thirdly, we developed a novel framework, scFeatures, that creates a molecular representation of individual samples from single-cell and spatial data, enabling personalised medicine in the single-cell era (Chapter 4). Using a comprehensive collection of disease datasets across multiple diseases and conditions, we illustrated the ability of scFeatures to perform a range of precision medicine applications including association studies, supervised classification of disease outcomes and unsupervised clustering of patients into subgroups with distinct survival outcomes. This work represents a novel approach towards defining samples based on their cellular characterization and opens the door for a spectrum of future explorations, such as multi-view learning framework utilising the sample representation to guide better understanding of disease and health.

Finally, using a case study on COVID-19 patients, we demonstrated how the novel works presented in Chapters 2, 3 and 4 are brought together to enable precision medicine. We utilised scFeatures to generate molecular characterisations of patients from COVID-19 single-cell datasets. We implemented ensemble learning techniques as well as deep learning models and observed improvement in prediction accuracy of mild and severe conditions. We also demonstrated the power of using multiple COVID-19 datasets to address precision medicine questions that cannot be easily addressed using single datasets. Future work could incorporate evaluation of the impact of different integration strategies on the prediction power as well as on gaining biological insights. Our evaluation framework SimBench, presented in Chapter 3, can be adapted to the development of this evaluation study.

During the course of my PhD, the rapid development of single-cell technology saw a typical experimental dataset grow from hundreds of cells to hundreds of thousands of cells. During the second and third years of my PhD, when the COVID-19 pandemic struck, thousands of patients were sequenced globally using single-cell technologies in the global effort to understand the molecular mechanism of diseases and assist in the development of therapeutics. As single-

cell technology continues to advance, we envisage the increasing availability of large-scale patient datasets and the increasing application of single-cell technology in the ongoing fight against diseases. The development of computational tools is the key to unleashing the enormous potential behind these patient datasets for precision medicine. The works described in my thesis contribute to the effective utilisation of single-cell data for precision medicine and provide insights for future method development towards the field.

In this thesis we have presented computational methods and frameworks for the effective utilisation of single-cell data for precision medicine. While the last chapter presents a case study specifically on COVID-19, we believe the concepts of deep learning, data simulation and the creation of patient-based summary statistics are general and applicable to other diseases such as cancer, as demonstrated in Chapter 4. This is the same for the wide repertoire of single-cell computational methods that address high-level questions, such as differential expression, data integration and trajectory analysis.

However, we also point out that there also exist distinct differences and challenges in the precision medicine approach for different diseases. For example, the approach to the study of cancer differs from that of infectious diseases such as COVID-19. One major difference is the type of cell population being studied. Cancer often arises from a small subpopulation of cells that have acquired genetic changes and the spread of cancer also often stems from a small subpopulation of cells that develop drug resistance. Cancer studies, therefore, benefit from methodological advance that enables the identification of rare subpopulations and the exploration of heterogeneity of the cells (González-Silva *et al.*, 2020; Lim *et al.*, 2020). In contrast, infectious diseases are caused by viral infection that affects the cells throughout the entire body. Such studies such as with COVID-19 typically focus on understanding the disease mechanisms in known cell types. Another distinct difference is that cancer studies are mostly concerned with the interactions within the patient's own cellular system. Currently, there are a number of cell-cell interaction methods (Armingol *et al.*, 2021) that can be used to address this task. The understanding of Infectious diseases

requires the development of methods for exploring the interaction between pathogen and host, which is a still relatively under-explored area (Penaranda and Hung, 2019). Methods for detecting infected cells and bystander cells (Bost *et al.*, 2020) and the interaction amongst them are in need. In future, as single-cell technology advances, we envisage the single-cell computational field will continue to evolve the address the general and specific challenges in precision medicine.

A

APPENDIX FOR CHAPTER 3

A.1 SUPPLEMENTARY FIGURES

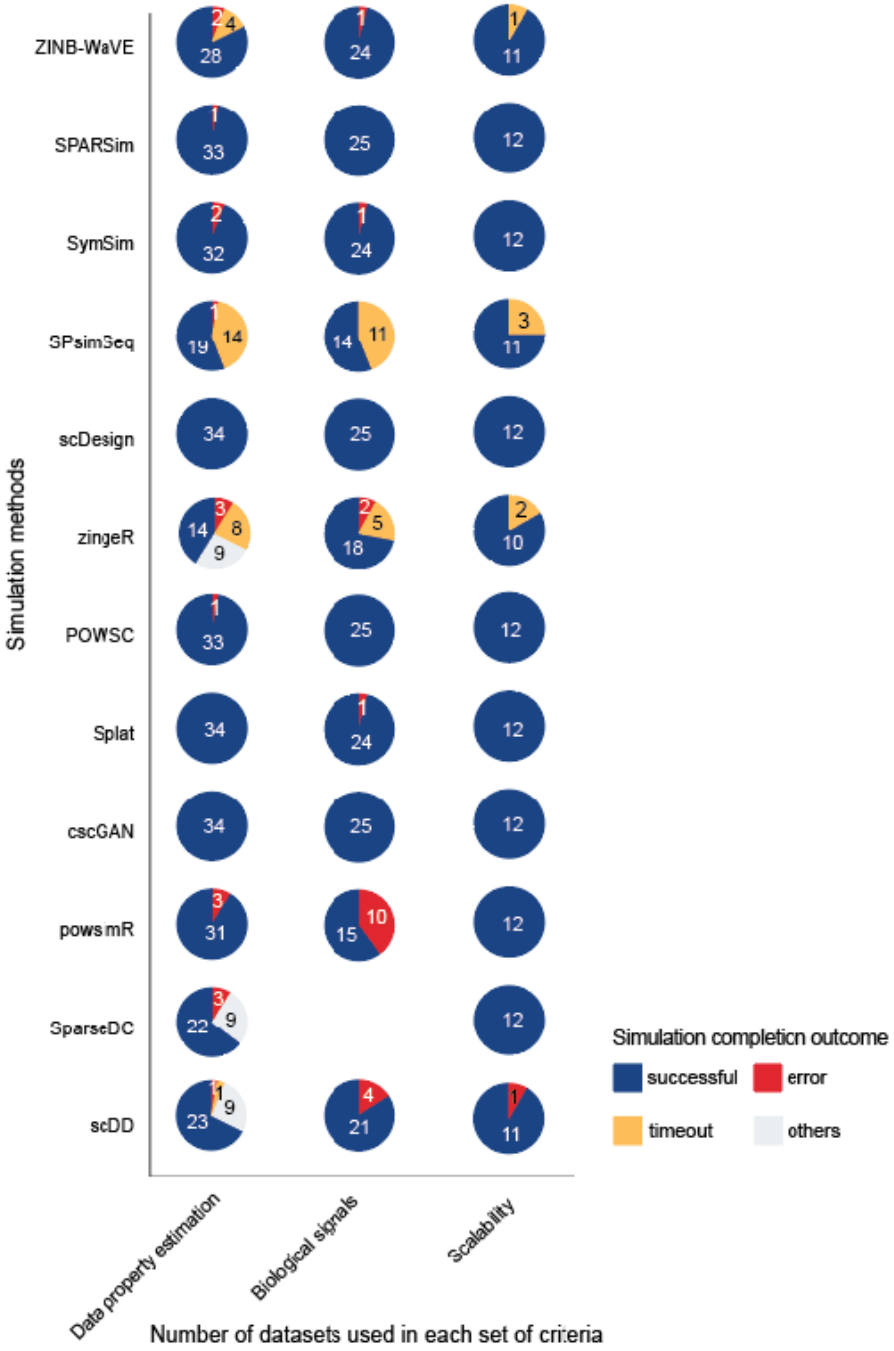


Figure A1: Number of datasets used for evaluation and simulation completion outcome. A total of 34 datasets were used for evaluating the simulation methods on data property estimation, 25 datasets were used for evaluating biological signals and 12 datasets were used for evaluating scalability (see Methods). Pie chart denotes the completion outcome of each simulation method. "Successful" indicates the method produced a simulation dataset within the given time limit. "Error" indicates the method encountered issues during simulation. "Timeout" indicates the method was not able to finish the simulation within the given time limit (see Methods). "Others" indicates a special situation where the methods require the dataset to have two or more cell types and therefore could not be tested on datasets containing a single cell type.

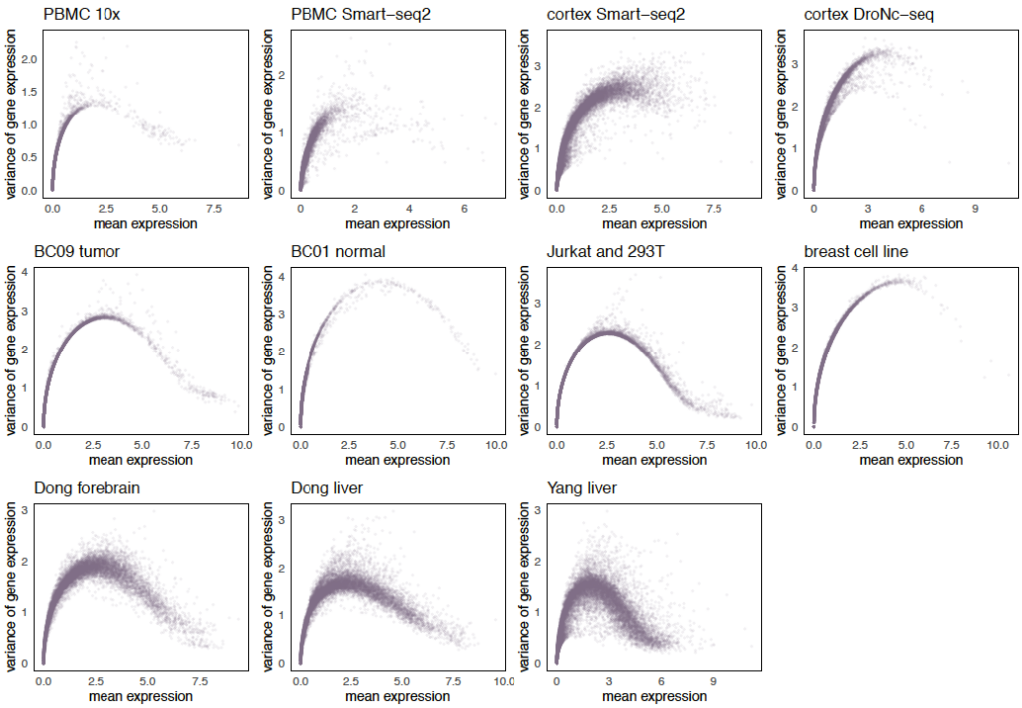


Figure A2: Mean-variance relationship of genes across multiple datasets. We show the property of mean-variance relationship of genes in 11 datasets as an example to demonstrate the variability of data properties across datasets. Top panel shows four datasets of different protocols, the first two from human PBMC samples and the latter two from mouse cortex samples. Middle panel shows two datasets from tissue source and two datasets with cell line source. Bottom panel shows datasets of multiple cell types in mouse sample.

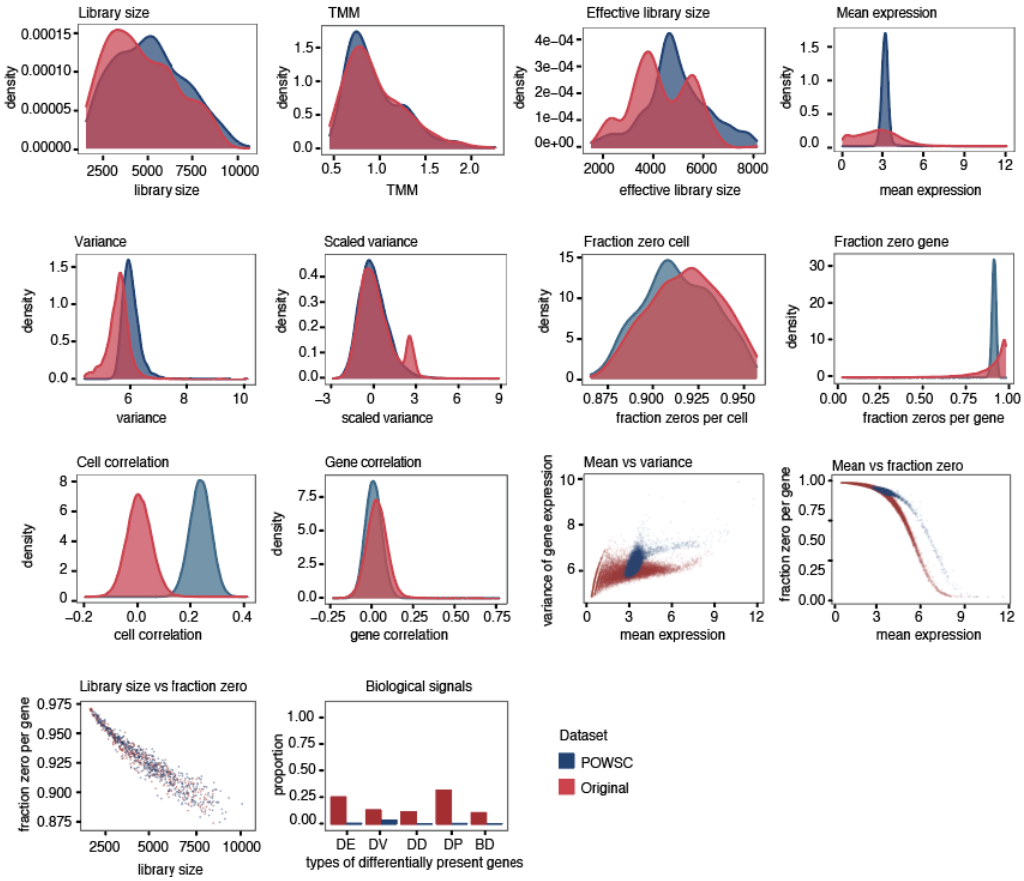


Figure A3: Visual representation of the evaluation criteria in properties estimation and biological signals. As an illustrative example, we compared the simulation data generated by POWSC and the original dataset Soumillon that was used as the reference input. In properties estimation, we compared the concordance of the data characteristics across multiple properties using the KDE statistic. In biological signals, we compared the concordance of the amount of proportion of biological signals in simulated and in real data.

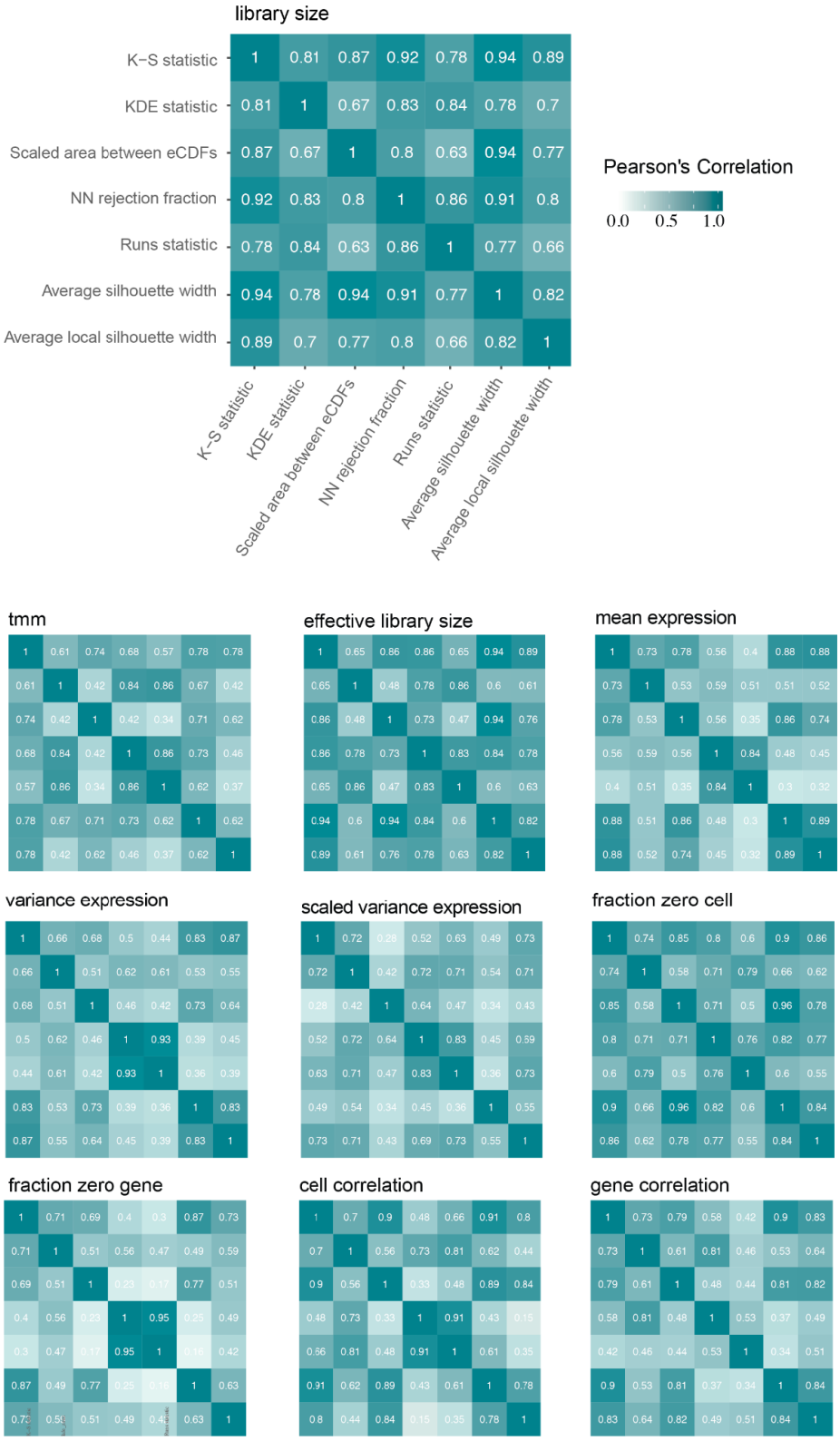


Figure A4: Correlation between seven measures on quantifying similarities for univariate properties. Top panel shows the correlation matrix for the property library size, enlarged for readability of axis labels. Bottom panel shows correlation matrix for the remaining univariate properties. The axis labels are consistent and are not shown for readability of the matrix.

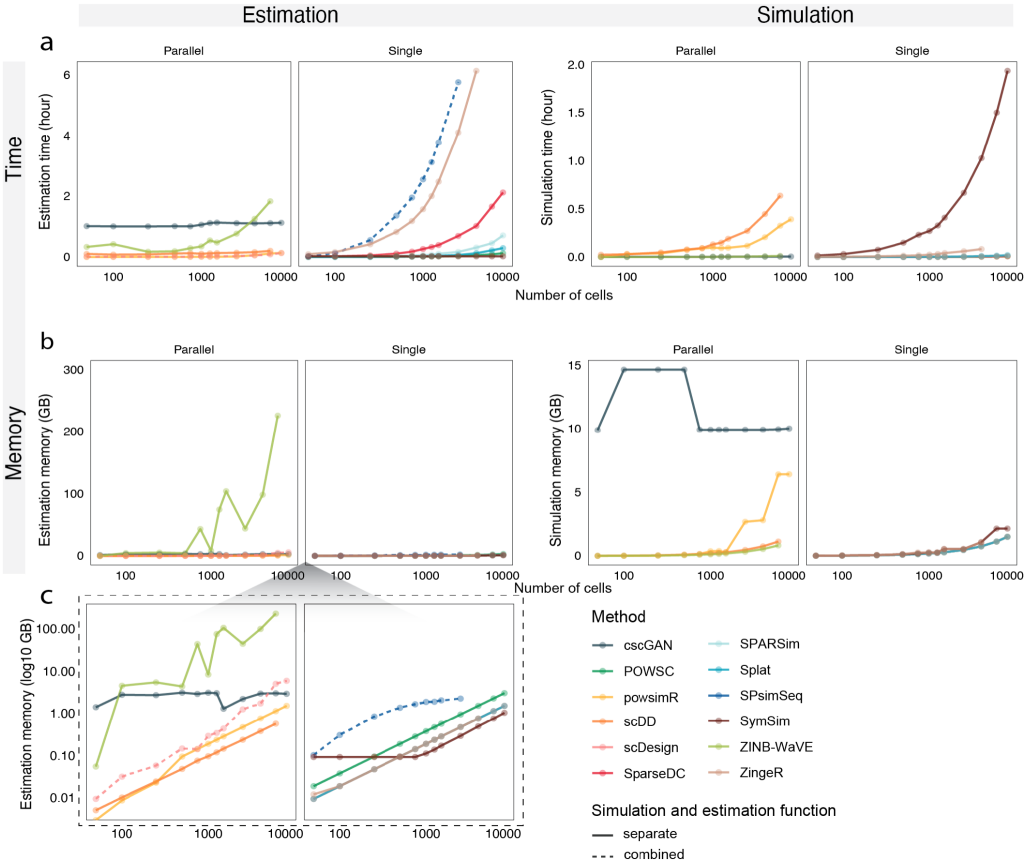


Figure A5: Run time and memory consumption of each method. (a) Runtime of each method. (b) Maximal memory usage of each method. The number of cells is shown in log10 scale. Methods that support parallel computing and those that only support single core are shown separately. Most methods involve a two-step process of properties estimation and dataset simulation. For those methods, we recorded and shown results for the two steps separately under the estimation and simulation panels. A solid line was used to indicate these methods. For methods that perform the two steps together in a single function, we displayed the results under the estimation panel. A dashed line was used to indicate these methods. (c) This shows the same result as in (b), but with the y-axis in log10 scale for enhanced readability.

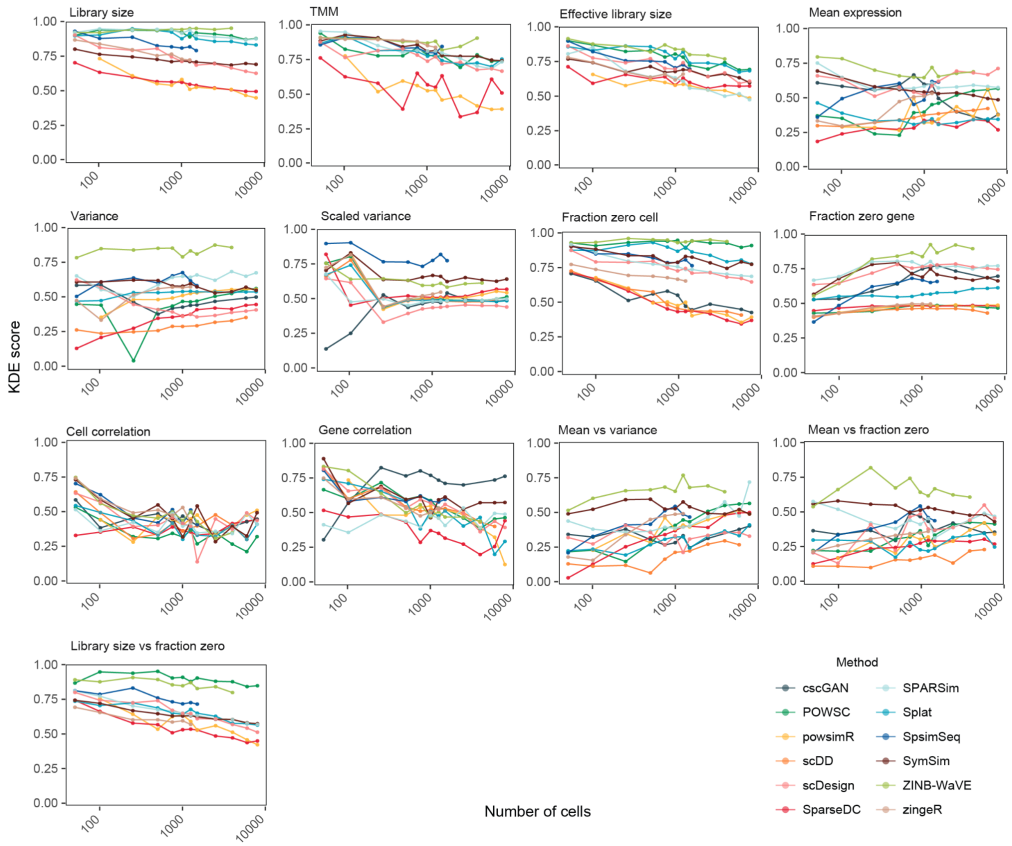


Figure A6: Impact of the number of cells on property estimation. The x-axis shows the number of cells in log10 scale and y-axis shows the score. The line shows the trends with increasing cell numbers. The dot indicates where a measurement is taken. Each measurement was taken three times and the average was shown in the figure.

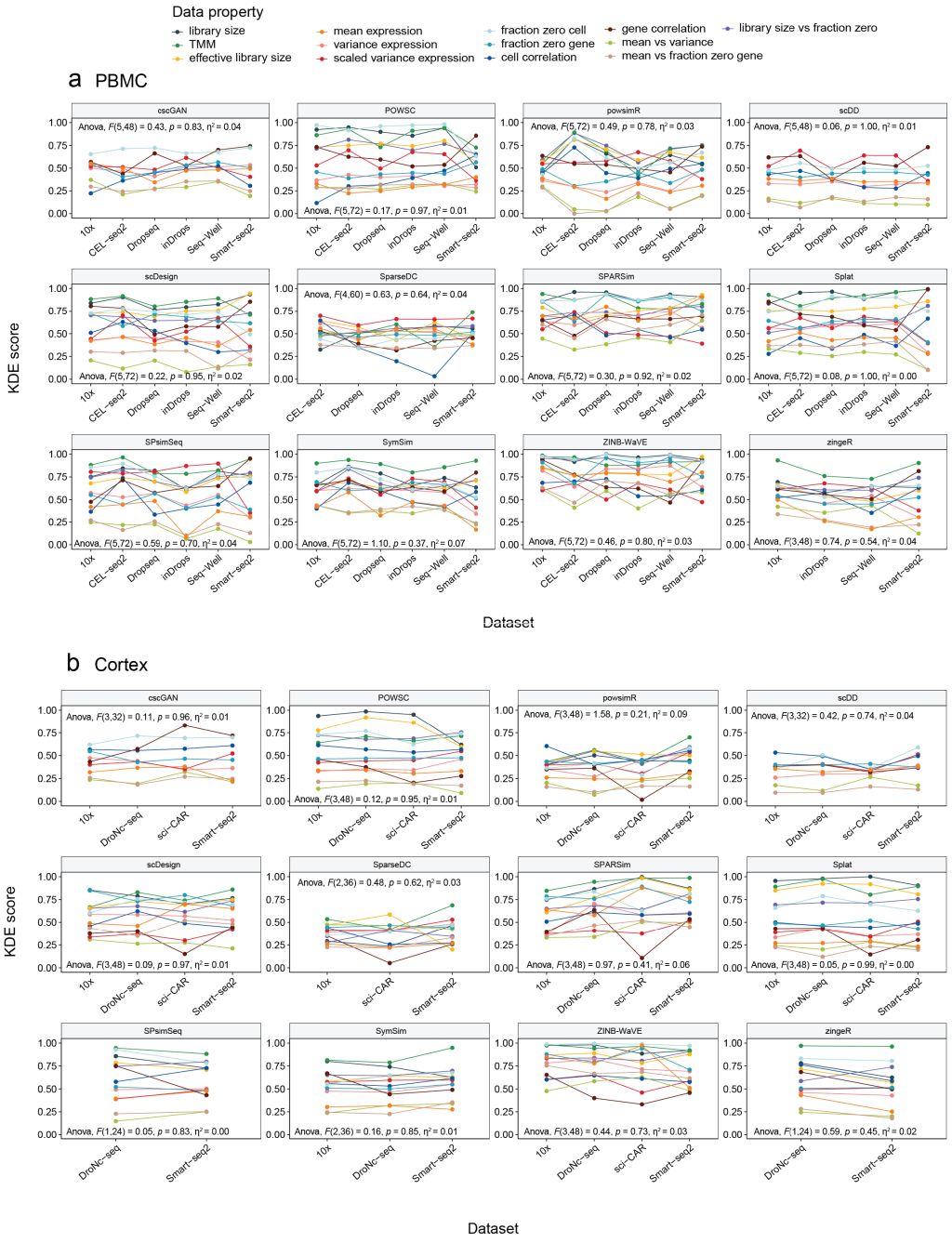


Figure A7: Impact of sequencing protocols on data property estimation. The impact of sequencing protocols on data property estimation using (a) human PBMC data collections and (b) mouse cortex data collections. ANOVA was performed to examine the statistical significance of the change in KDE score due to sequencing protocol effect. The test statistics, effect sizes, degrees of freedom and P-values are shown on each panel.

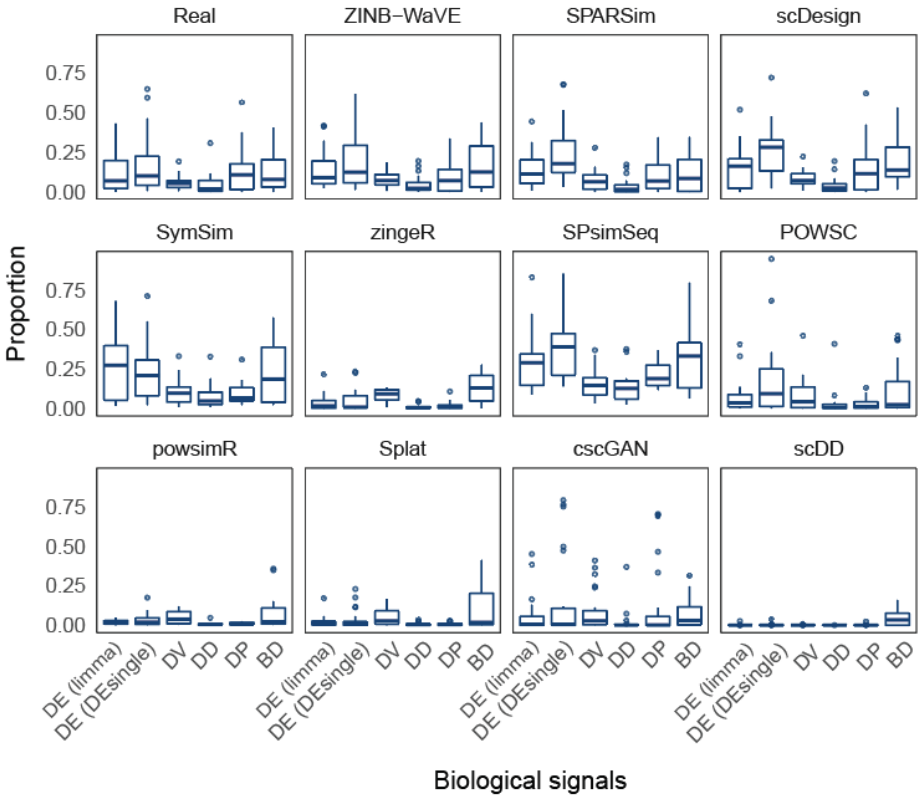


Figure A8: Proportion of biological signals in real and simulated data generated by simulation methods. The boxplots show the distribution of the proportion of biological signals for all datasets examined (n = 25 for SPARSim, scDesign, POWSC and cscGAN, n = 24 for ZINB-WaVE, SymSim and Splat, n = 21 for scDD, n = 18 for zinger, n = 15 for powsimR, n = 14 for SPsimSeq). The proportion of biological signals in the simulated data ideally should be similar to that of the real data. The box represents quartiles, the line represents the median, the lower and upper whisker represents the bottom 25% and top 25% of the data. Outliers are shown as individual data points.

A.2 SUPPLEMENTARY TABLES

Table A1: Details of the datasets used in this study.

Dataset	Accession	Name	Description	Species	Protocol	Number of cells	Multiple cell types or condition ?	Source
1	SCP425	cortex sciRNAseq	Comparison of four protocols using mouse cortex	Mouse	sci-RNA-seq	4912	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP425/single-cell-comparison-cortex-data#study-download
2	SCP425	cortex 10x		Mouse	10x Genomics	5367	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP425/single-cell-comparison-cortex-data#study-download
3	SCP425	cortex DroNc-seq		Mouse	DroNc-seq	2345	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP425/single-cell-comparison-cortex-data#study-download
4	SCP425	cortex Smart-seq2		Mouse	Smart-seq2	644	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP425/single-cell-comparison-cortex-data#study-download
5	SCP424	PBMC 10x	Comparison of six protocols using human PBMC	Human	10x Genomics	3312	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
6	SCP424	PBMC CEL-seq2		Human	CEL-seq2	526	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
7	SCP424	PBMC Drop-seq		Human	Drop-seq	6357	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
8	SCP424	PBMC inDrops		Human	inDrops	4184	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
9	SCP424	PBMC Seq-Well		Human	Seq-Well	2908	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
10	SCP424	PBMC Smart-seq2		Human	Smart-seq2	522	Yes	https://singlecell.broadinstitute.org/single_cell/study/SCP424/single-cell-comparison-pbmc-data#study-summary
11	see source	Tabula Muris	The 10x subset of Tabula Muris	Mouse	10x Genomics	55656	Yes	https://tabula-muris.ds.czbiohub.org/
12	GSE114724	BC09 tumor	Tumor of breast cancer patient ID BC09	Human	10x Genomics	7000	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114724
13	GSE114725	BC02 tumor	Tumor of breast cancer patient ID BC02	Human	inDrops	2437	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114725
14	GSE114725	BC01 blood	Blood of breast cancer patient ID BC01	Human	inDrops	3034	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114725
15	GSE114725	BC02 lymph	Lymph node of breast cancer patient ID BC02	Human	inDrops	6129	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114725
16	GSE114725	BC01 normal	Normal breast tissue of breast cancer patient ID BC01	Human	inDrops	3607	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114725
17	GSE106202	breast cell line	MDA-MB-231 cells cultured in glucose	Human	Drop-seq	785	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE106202
18	GSE102827	light endo	Endothelial smooth muscle of primary visual cortex from mice, exposed to light for 0h, 1h and 4h	Mouse	inDrops	4071	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE102827
19	GSE102827	light micro	Microglia of primary visual cortex from visually stimulated mice, exposed to light for 0h, 1h and 4h	Mouse	inDrops	10158	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE102827
20	GSE92495	Gierahn	Human HEK293 (embryonic kidney cells) cell line	Human	Seq-Well	1453	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE92495
21	see source	293T	293T (adenovirus-immortalized human embryonic kidney cells) cell line	Human	10x Genomics	2885	No	https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/293t
22	see source	Jurkat and 293T	Mixture of Jurkat (human T lymphocyte) and 293T	Human	10x Genomics	6143	Yes	https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/jurkat

23	GSE77288	Tung	Three iPSC (Induced Pluripotent Stem Cells) lines	Human	SMARTer	564	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE77288
24	GSE113660	Chen	Rh41(human alveolar rhabdomyosarcoma) cell line	Human	10x Genomics	6875	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE113660
25	GSE60361	Zeisel	Cortex of mice	Mouse	STRT-seq	3005	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE60361
26	GSE72857	Pual	Bone marrow myeloid progenitors	Mouse	MARS-seq	6144	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE72857
27	GSE63472	retina	Mouse retina	Mouse	Drop-seq	6598	No	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63472
28	GSE87038	Dong forebrain	Forebrain cells of E9.5 to E11.5 mouse embryos	Mouse	Smart-seq2	196	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE87038
29	GSE87038	Dong skin	Skin cells of E9.5 to E11.5 mouse embryos	Mouse	Smart-seq2	196	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE87038
30	GSE87038	Dong intest	Intestine cells of E9.5 to E11.5 mouse embryos	Mouse	Smart-seq2	196	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE87038
31	GSE87038	Dong liver	Liver cells of E9.5 to E11.5 mouse embryos	Mouse	Smart-seq2	196	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE87038
32	GSE90047	Yang liver	Liver cells of E10.5 to E17.5 mouse embryos	Mouse	Smart-seq2	447	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE90047
33	GSE75748	stem cell	Human pluripotent stem cells (hPSCs)	Human	SMARTer	758	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE75748
34	GSE112004	Francesconi	B cell precursors from bone marrow, induced to either transdifferentiate to macrophages or to reprogram into iPSCs	Mouse	MARS-Seq	3833	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE112004
35	GSE53638	Soumillon	Differentiating cells of human adipose-derived stem/stromal cells	Human	SCRB-Seq	2968	Yes	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE53638

Table A2: Additional details of the scRNA-seq simulation methods evaluated in this study.

Methods	Implementation language	Year of publication	Reference (doi)	Software version	Input data (raw/ normalised)	Output data (raw/normalised)
scDD	R	2016	10.1186/s13059-016-1077-y	1.12.0 (implemented in Splatter)	raw	normalised
Splat	R	2017	10.1186/s13059-017-1305-0	1.12.0	raw	raw
powsimR	R	2017	10.1093/bioinformatics/btx435	1.2.3	raw	raw
SparseDC	R	2017	0.1093/nar/gkx1113	0.1.17 (implemented in Splatter)	raw	raw
zingeR	R	2018	10.1186/s13059-018-1406-4	0.1.0	raw	raw
ZINB-WaVE	R	2018	10.1038/s41467-017-02554-5	1.10.0 (implemented in Splatter)	raw	raw
SymSim	R	2019	10.1038/s41467-019-10500-w	0.0.0.9000	raw	raw
scDesign	R	2019	10.1093/bioinformatics/btz321	1.0.0	raw	raw
SPARSim	R	2020	10.1093/bioinformatics/btz752	0.9.5	both raw and normalised	raw
SPsimSeq	R	2020	10.1093/bioinformatics/btaa105	0.99.13	raw	raw
POWSC	R	2020	10.1093/bioinformatics/btaa607	0.1.0	raw	raw
CSCGAN	Python	2020	10.1038/s41467-019-14018-z	GitHub version 379ff6e	raw	normalised

Table A3: Detailed simulation strategy of each method. (*) We used the procedure described in "Evaluation of biological signals" of the Methods section to calculate the proportion of differential expressed genes between the two largest cell types in the real data. This proportion was then used as the input parameter in the simulation function to control the proportion generated in the simulation data.

Methods	Simulation Strategy for evaluating data property estimation	Simulation Strategy for evaluating biological signals
Splat	Estimated the parameters and simulated each cell type separately.	Estimated parameters from the largest cell type in a dataset, set the number of groups to 2 and the proportion of differential expressed (DE) genes to the proportion between the two largest cell types in the dataset (*). This is because the genes in the simulated data do not have a one-to-one matching relationship with the input data and hence it is not possible to combine two simulated data generated from two cell types separately.
powsimR	This method generates DE genes from a homogenous population, for example, a particular cell type from one patient to create two artificial populations. We therefore estimated the parameters and simulated each cell type separately. The proportion of DE and log fold change were set to be a null scenario to maintain the biological signals in the original cell type population.	This method generates DE genes from a homogenous population. We therefore estimated the parameters and simulated the largest cell type. The proportion of DE was set to the proportion between the two largest cell types in the dataset.
SymSim	Estimated the parameters and simulated each cell type separately.	Estimated the parameters and simulated the two largest cell types separately.
scDesign	Estimated the parameters and simulated each cell type separately.	This method generates DE genes from a homogenous population. We therefore estimated the parameters and simulated the largest cell type. The proportion of DE was set to the proportion between the two largest cell types in the dataset
SPARSim	Estimated the parameters and simulated each cell type separately.	Estimated the parameters and simulated the two largest cell types separately. This is because the method returns gene names in the simulated data and therefore we can combine the two datasets and evaluate the biological signals between the two cell types.
SPsimSeq	Estimated the parameters and simulated each cell type separately.	Estimated the parameters and simulated the two largest cell types separately.
POWSC	Estimated the parameters and simulated each cell type separately.	Estimated the parameters and simulated the two largest cell types separately.
zingeR	We estimated and simulated every two cell types at a time with the proportion of DE gene set to 10%. This is the setting used by the authors of this method when comparing their simulated dataset to the original dataset.	We estimated and simulated the two largest cell types at a time with the proportion of DE gene set to the proportion between these two cell types.
scDD	We estimated and simulated every two cell types at a time with the proportion of DE genes set to 10%. This is because the method requires two cell types to be simulated at once with a given proportion of DE genes between them.	We estimated and simulated the largest two cell types with the proportion of DE genes set to the proportion between these two cell types.
ZINB-WaVE	This method takes cell types label into consideration in the parameter estimation step, thus estimation and simulation were performed directly on the entire dataset with cell type labels provided.	Estimation and simulation were performed directly on the entire dataset with cell type labels provided. We then evaluated the biological signals between the two largest cell types.
SparseDC	This method requires two conditions such as treatment and control, with multiple cell types in each condition, as an internal clustering step is performed to differentiate the cell types. We followed the procedure in he SparseDC documentation and split half of the cell types into condition 1 and half of the cell types into condition 2, and specified the number of clusters to be the number of cell types in condition 1 and 2.	Due to the unique setting, we did not evaluate this method for biological signals.
cscGAN	This method takes cell types label into consideration in the parameter estimation step, thus estimation and simulation were performed directly on the entire dataset with cell type labels provided.	Estimation and simulation were performed directly on the entire dataset with cell type labels provided. We then evaluated the biological signals between the two largest cell types.

B

APPENDIX FOR CHAPTER 4

B.1 SUPPLEMENTARY FIGURES

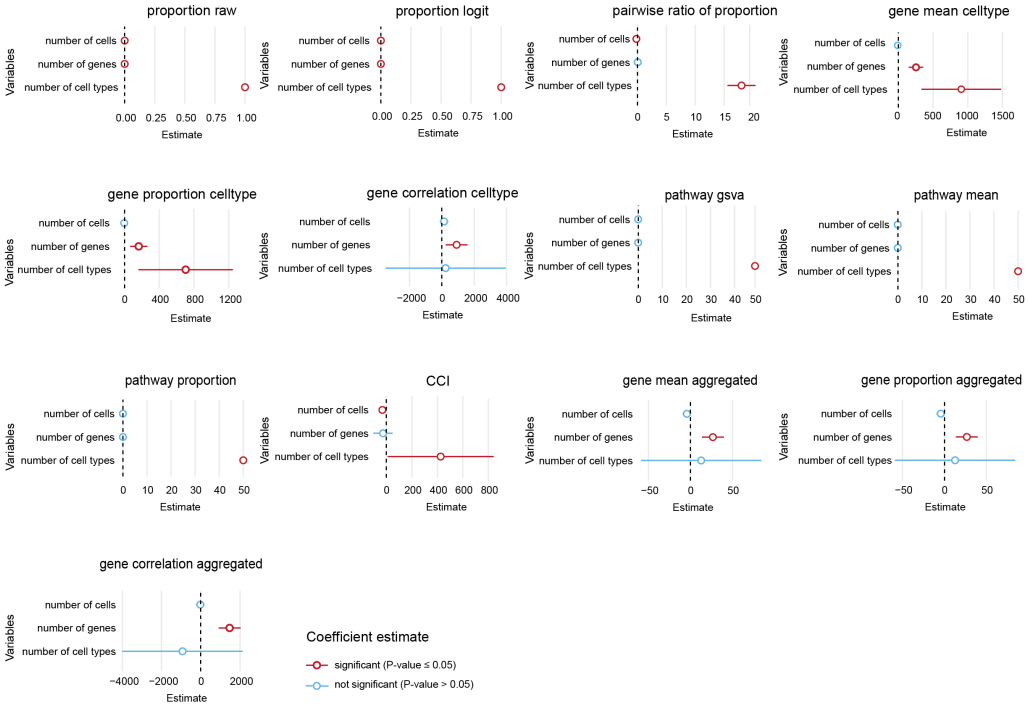


Figure B1: Impact of dataset characteristics on number of features generated by scFeatures. We generated features on 15 scRNA-seq datasets (see Methods). Linear regression model was fitted to explore the relationship between the number of features and dataset characteristics such as number of cells, genes, cell types and patients. The regression coefficient for each variable is shown in the line plots, with red denoting a significant relationship and blue denoting an insignificant relationship.

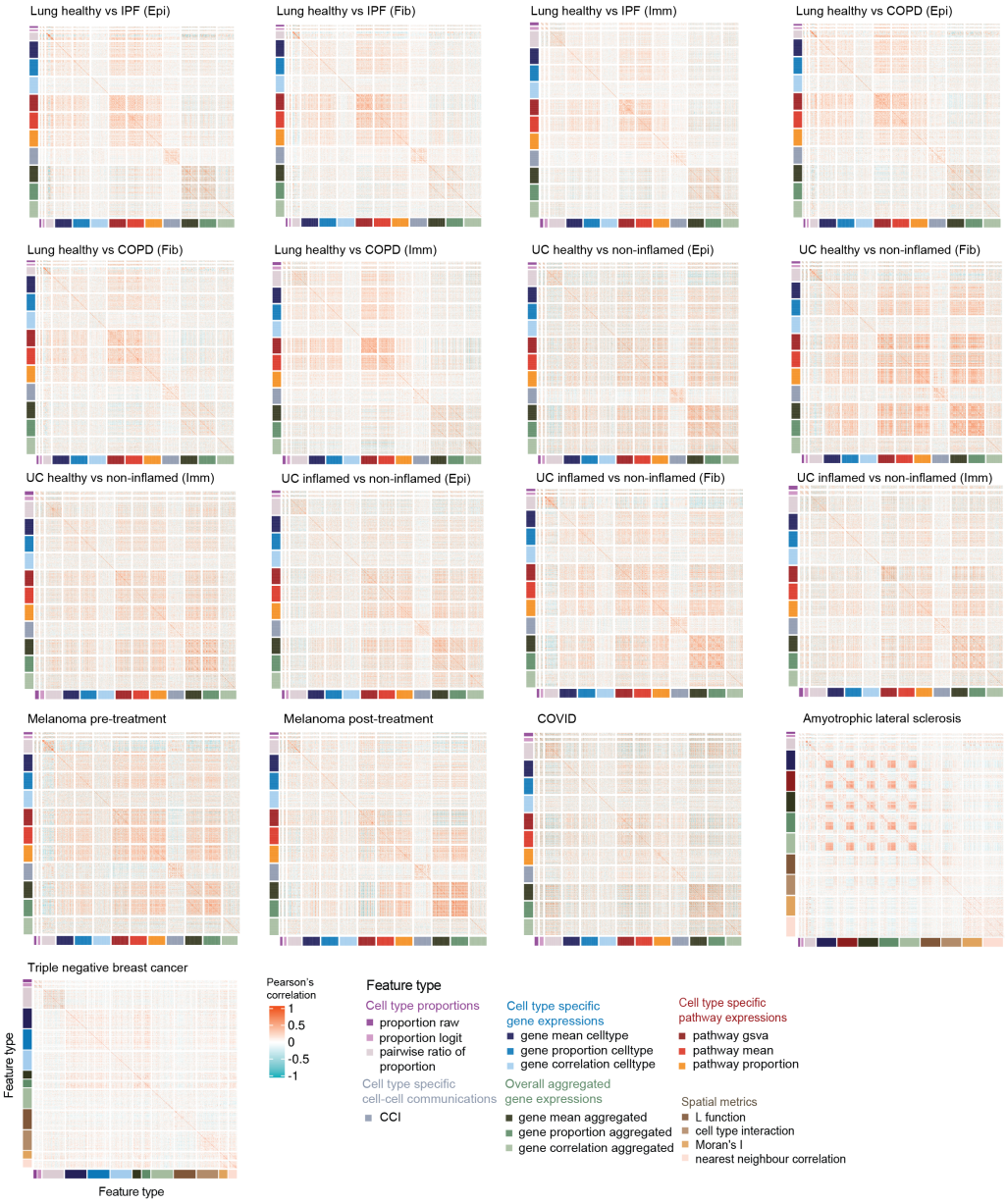


Figure B2: Correlation amongst feature pairs for each dataset. Plots show the Pearson's correlation between features on each dataset. The features are colour labelled by feature class for ease of interpretation. To avoid the correlation plot being dominated by feature classes with more features, we subsampled 100 features from feature classes with more than 100 features.

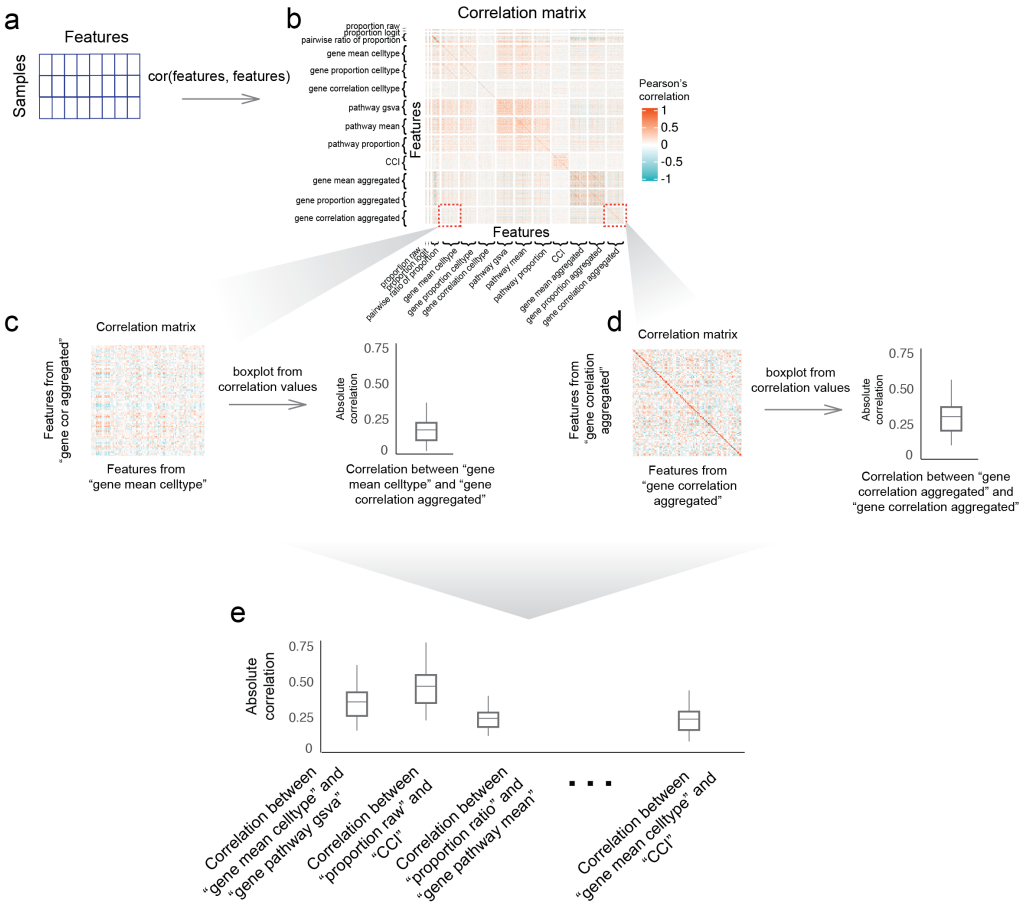


Figure B3: Schematic representation of the calculation of correlation between feature types. (a) First, for a given dataset, the features from all feature types are created, yielding a samples by features matrix. Pearson's correlation is calculated on the features matrix to result in a typical correlation matrix comprising the correlation between the individual features, as shown in (b). Since each feature is associated with a feature type, we can zoom into a section of the correlation matrix that contains the correlations of features from two feature types. For example, (c) shows the section of correlation matrix, which contains the features from the feature type "gene mean celltype" and "gene correlation aggregated". A boxplot can then be constructed to summarise the correlations between these two feature types. d shows another section of the correlation matrix, which contains the correlations between all the features from the feature type "gene correlation aggregated". (e) Repeating this across each section of the correlation matrix produces boxplots summarising the correlation between all pairwise combinations of feature types.

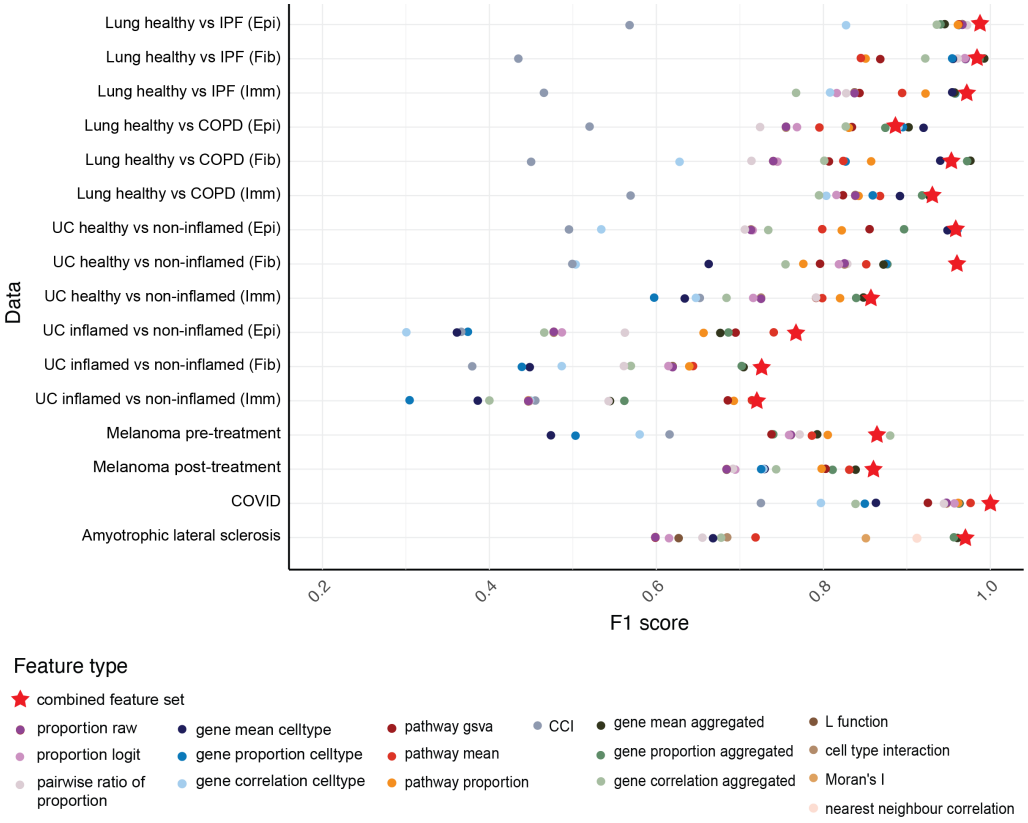


Figure B4: Classification performance of the top 100 features from each individual feature type and the combined feature set on 16 datasets. Models were first trained on the entire feature space of each individual feature type (see Methods). The top 8 features from each feature type were identified and combined into the "combined feature set" containing around 100 features, which was used for model training and testing. For comparison, the top 100 features from each feature type were identified and used for model training and testing. Random forest was used as the model choice and performance was evaluated in terms of F1 score. Each point represents the average from 50 cross-validation.

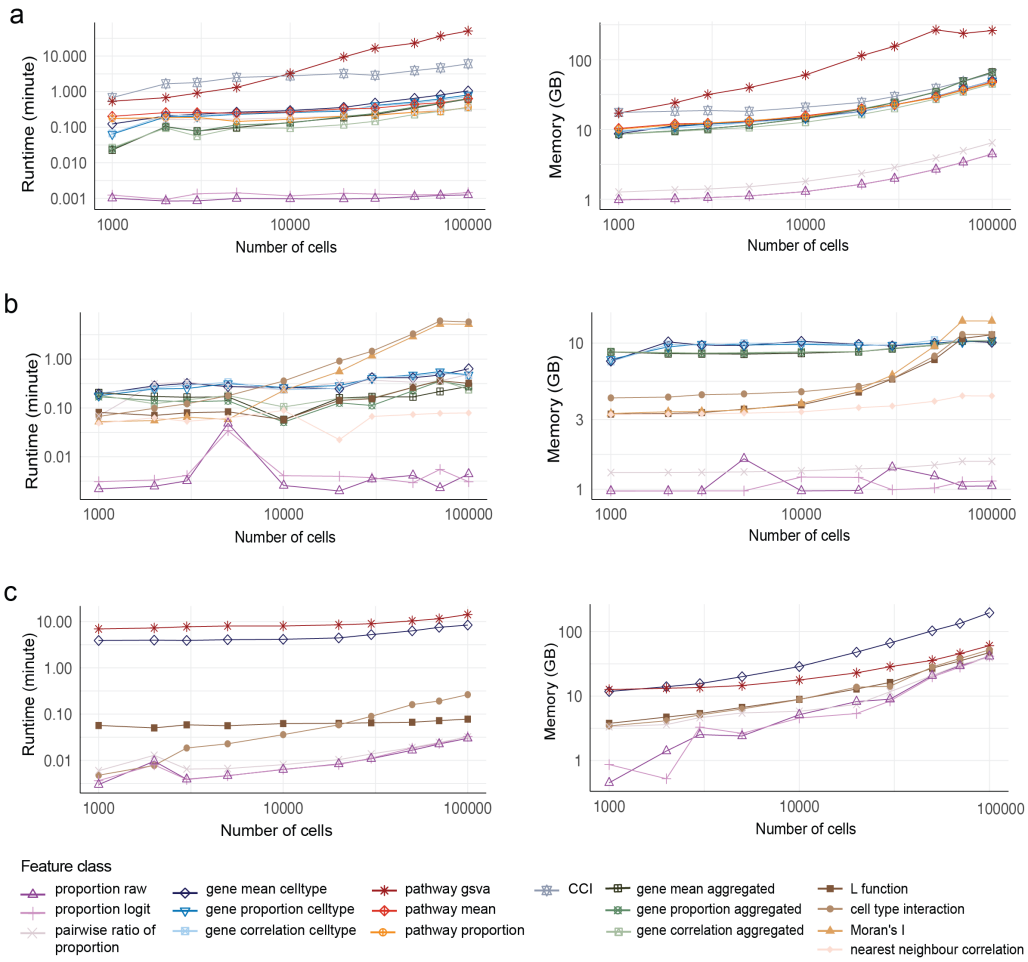


Figure B5: Scalability analysis of feature types. The x and y axes are displayed on a log10 scale in all panels. (a) The runtime and memory usage of feature types benchmarked on subsampled scRNA-seq data. (b) The runtime and memory usage of feature types benchmarked on subsampled spatial proteomics data. (c) The runtime and memory usage of the feature types adapted for the spot-based data, evaluated using spatial transcriptomics data.

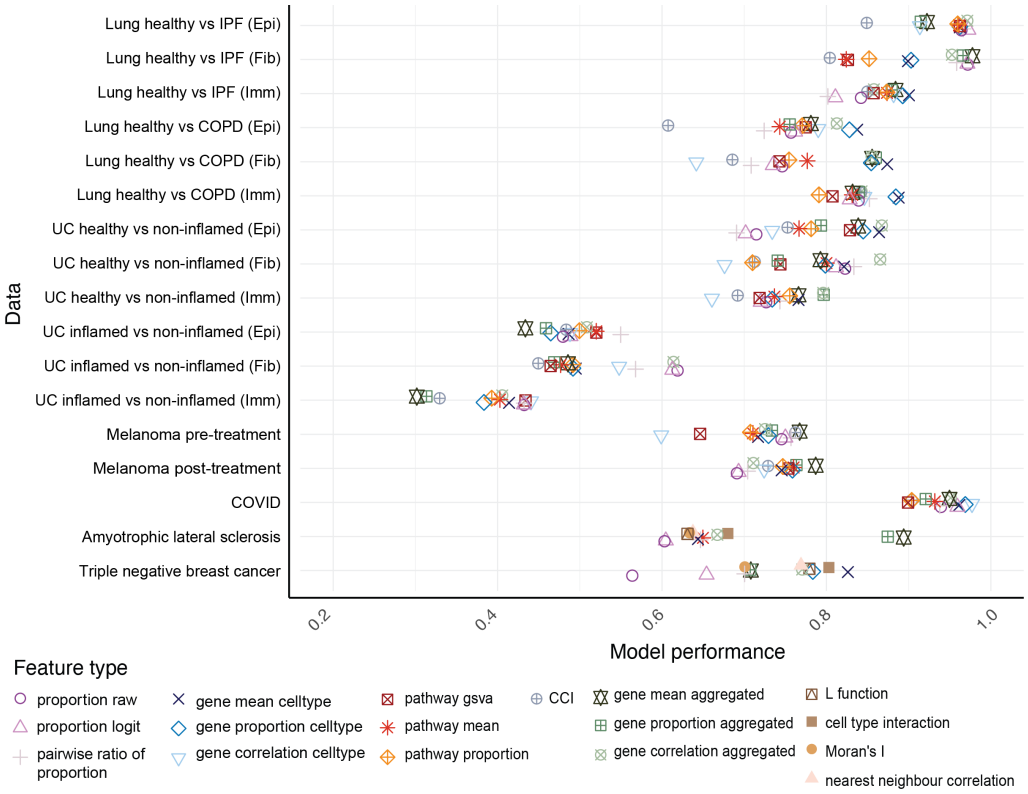


Figure B6: Model performance of each feature class on all datasets. For datasets with disease outcome, random forest was used and model performance was evaluated in terms of F1 score. For the dataset "Triple negative breast cancer" with survival outcome, cox proportional-hazard model was used and model performance was evaluated in terms of C-index. Each point represents the average from 50 cross-validation models.

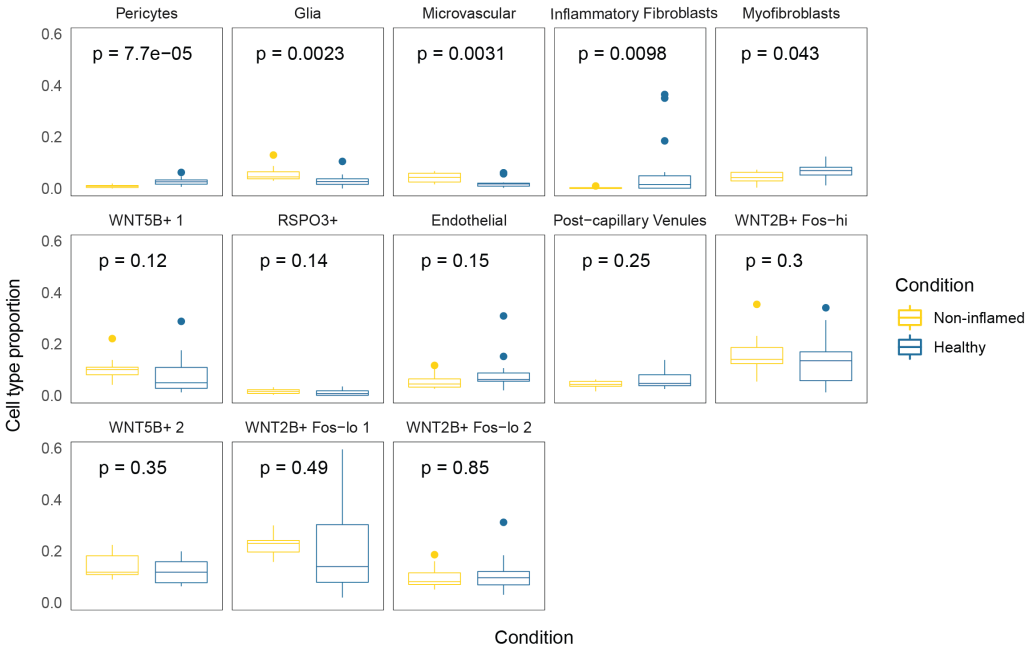


Figure B7: Cell type proportion of the patients in the "UC healthy vs non - inflamed (Fib)" dataset. Wilcoxon test was performed on each cell type to compare the cell type proportion between the non-inflamed and healthy samples.

B.2 SUPPLEMENTARY TABLES

Table B1: Implementation details of each feature type.

Feature category	Feature class	Applicable data types	Implementation details
Cell type proportions	Proportion raw	scRNA-seq and spatial proteomic	Calculates the proportion of each cell type in each sample.
		spatial transcriptomics	Each spot is represented as multiple single cells by multiplying the relative number of cells with the cell type probability of each spot. The proportion of each cell type is then calculated based on this single cell representation.
	Proportion logit	scRNA-seq and spatial proteomics	Performs logit transformation of the cell type proportion as it is one of the most common transformations for proportional data.
		spatial transcriptomics	Performs logit transformation of the cell type proportion based on the single cell representation as described in the implementation of "Proportion raw".
	Proportion ratio	scRNA-seq and spatial proteomics	Computes the pairwise ratio of two cell types' proportions, i.e. cell type 1 divided by cell type 2. This is calculated for each paired cell type combination. To avoid dividing by zero when a cell type is not present in a patient, we add 1 to both the numerator and denominator. The range of value is then scaled using log2 transformation.
		spatial transcriptomics	Computes the pairwise ratio of two cell types' proportions based on the single cell representation as described in the implementation of "Proportion raw".
Cell type specific gene expressions	Gene mean celltype	scRNA-seq and spatial proteomics	Calculates the mean expression of genes within each cell type. Users can provide their own genes of interest to the function. If not provided (the setting presented in this paper), we restrict to the top variable genes to reduce the dimensions of the feature. This is particularly important for scRNA-seq data as it generally contains more than 20,000 genes. We calculate two sets of highly variable genes, 1) across all cells within each cell type and 2) across all cells. First, for each cell type, the genes of interest are obtained by selecting the top variable genes per sample, followed by taking the union of the genes across all samples. The default number of variable genes is set to 100 per cell type and is a parameter that can be specified by the user. Since the variable genes are calculated separately for each cell type, this results in a different set of genes for each cell type. Then, the top variable genes across all cells are calculated per sample, followed by taking the union of the genes across all samples. The default number of variable genes is set to 100 and is also a parameter that can be specified by the user. The final output is a vector of mean expression for the variable genes.
		spatial transcriptomics	Given the expression values in each spot represents multiple cells of potentially different cell types, we devised the following approach. For each gene, we regress the count against the probability of each cell type across all spots in a sample to obtain the coefficient and P-value of each cell type. Then for each cell type identify the top genes that have smallest P-values on average across all samples. These would be the genes most associated with the cell type. The regression coefficients of these genes are then the features. The number of top genes is default to 50.
	Gene proportion celltype	scRNA-seq and spatial proteomics	For each gene, we calculate the proportion that this gene is expressed across all cells. This is performed separately for each cell type of each patient. Users can provide their own genes of interest to the function. If not provided (the setting presented in this paper), we restrict to the top variable using the same procedure as defined in "gene mean celltype". The final output is a vector of proportion expressed for the subset of cell type specific genes.
		spatial transcriptomics	N/A as the expression values in each spot represents multiple cells of potentially different cell types
	Gene correlation celltype	scRNA-seq and spatial proteomics	Users can provide their own genes of interest to the function. If not provided (the setting presented in this paper), we restrict to the top variable using the same procedure as defined in "gene mean celltype". Then, for the selected genes, we calculate the pairwise correlation between two genes based on their expression values. The final output is a vector of genewise correlation for the subset of cell type specific genes.
		spatial transcriptomics	N/A as the expression values in each spot represents multiple cells of potentially different cell types
Cell type specific pathway expressions	Pathway GSVA	scRNA-seq	We implemented methods from the GSVA (Hänzelmann *et al.*, 2013) package to obtain the gene set enrichment score for each single cell of a patient. Implementation of AUCell (Aibar *et al.*, 2017) is also provided as an option to the user. The enrichment score is then summarised for each cell type by averaging the scores from all the single cells within a cell type. As a result, this approach converts the matrix of gene expressions by single cells into pathways by cell types for each patient. The matrix of pathways by cell types is further converted into a single vector by concatenating the scores from each cell type. By default (the setting presented in this paper), we use the 50 hallmark pathways from MSigDB (Liberzon *et al.*, 2015). Users can also provide their own pathways of interest to the function.
		spatial proteomics	N/A as the number of proteins in spatial proteomics is generally too few to calculate pathway enrichment.
		spatial transcriptomics	We obtain the regression coefficients of each gene associated with each cell type as described in the implementation details of "Gene mean celltype". The regression coefficients for all the genes involved in a particular pathway are then summed. The summation is done separately for each cell type.

	Pathway mean	scRNA-seq	For each pathway, we average the gene expression values for all the genes in the pathway across all cells. This is done separately for each cell type of each patient. By default (the setting presented in this paper), we use the 50 hallmark pathways from MSigDB (Liberzon *et al.*, 2015). Users can also provide their own pathways of interest to the function.
		spatial proteomics	N/A as the number of proteins is too small to calculate pathway enrichment.
		spatial transcriptomics	N/A as the expression values in each spot represents multiple cells of potentially different cell types.
	Pathway proportion	scRNA-seq	For each pathway, we average the gene expression values for all the genes in the pathway for each cell and used the third quantile of this value as a threshold. We then calculate the proportion of cells in each patient that have a higher average expression greater than the threshold. This is done separately for each cell type so that the final output for a patient is a vector By default (the setting presented in this paper), we use the 50 hallmark pathways from MSigDB (Liberzon *et al.*, 2015). Users can also provide their own pathways of interest to the function.
		spatial proteomics	N/A as the number of proteins is too small to calculate pathway enrichment.
		spatial transcriptomics	N/A as the expression values in each spot represents multiple cells of potentially different cell types
Cell-cell interactions	CCI	scRNA-seq	We implemented methods from the CellChat30 package to calculate the cell - cell interaction probability between ligand and receptor pairs. This feature class is cell type specific, as the interaction between ligand and receptor is quantified separately for each cell type. The final output is a vector of interaction probabilities for each patient.
		spatial proteomics	N/A as the number of proteins is too small to query the cell cell interactions.
		spatial transcriptomics	N/A as calculation of cell-cell interaction relies on expression of individual cells.
Overall aggregated gene expressions	Gene mean aggregated	scRNA-seq, spatial proteomics and spatial transcriptomics	First the mean expression of genes across all cells is computed for each sample. We then restrict to the top variable genes using the same procedure as defined in "gene mean celltype". The number of variable genes is set to 1500 by default and can be specified by the user. Alternatively, users can provide their own genes of interest to the function. The function then uses the provided set of genes instead of the top variable genes.
	Gene proportion aggregated	scRNA-seq, spatial proteomics and spatial transcriptomics	For each gene, we calculate the proportion that this gene is expressed across all cells for each patient. We then restrict to the top variable genes using the same procedure as defined in "gene mean celltype". The number of variable genes is set to 1500 by default and can be specified by the user. Alternatively, users can provide their own genes of interest to the function. The function then uses the provided set of genes instead of the top variable genes.
	Gene correlation aggregated	scRNA-seq, spatial proteomics and spatial transcriptomics	We first obtain top variable genes using the same procedure as defined in "gene mean celltype". The number of variable genes is set to 100 by default and can be specified by the user. Then, for the selected genes, we calculate the pairwise correlation between two genes based on their expression values. The final output is a vector of genewise correlation. Alternatively, users can provide their own genes of interest to the function. The function then uses the provided set of genes instead of the top variable genes.
Spatial metrics	L function	scRNA-seq data	N/A as there are no spatial coordinates.
		spatial proteomics	The L values between the pairs of proteins are calculated using the L function defined in literature 31 and used as the features. L value greater than zero indicates spatial attraction of the pair of proteins whereas L value less than zero indicates spatial repulsion.
		spatial transcriptomics	We calculate the L function based on the single cell representation as described in the implementation of "Proportion raw".
	Cell type interaction	scRNA-seq data	N/A as there are no spatial coordinates.
		spatial proteomics	We find the nearest neighbours of each cell and the cell types of these neighbours. These are considered as spatial interaction pairs. The cell type composition of the spatial interaction pairs are used as features.
		spatial transcriptomics	We assume that the nearest neighbours should be the cells captured within each spot and consider them as the spatial interaction pairs. We use single cell representation as described in the implementation of "Proportion raw" to calculate the following procedure: for a spot containing n1 cell type x and n2 cell type y, the spatial interaction composition of of cell type x with cell type x is calculated as n1 / (n1+n2) * n1 / (n1+n2). Similarly for the spatial interaction composition of cell type x with cell type y. We then sum the spatial interaction composition across all spots and use them as the features.
	Moran's I	scRNA-seq data	N/A as there are no spatial coordinates.
		spatial proteomics and spatial transcriptomics	Moran's I are calculated using the function defined in literature 32 and used as the features. It calculates the spatial autocorrelation based on both the locations and values simultaneously. A value closer to 1 indicates clustering of similar values and a value closer to -1 indicates clustering of dissimilar values. A value of 0 indicates no particular clustering structure, ie, the values are spatially distributed randomly.
	Nearest neighbour correlation	scRNA-seq data	N/A as there are no spatial coordinates.

		spatial proteomics and spatial transcriptomics	Pearson correlation is calculated for the protein expression between a cell with its nearest neighbour cell for spatial proteomics and for gene expression between a spot with its nearest neighbour spot for spatial transcriptomics.

Table B2: Details of the datasets used in the study.

Dataset name referred in the study	Study reference	Outcome	Number of genes/proteins	Number of cells/spots	Number of samples	Species	Type of data
Lung healthy vs IPF (Epi)	Adams *et al.* (2020)	Healthy vs IPF	45947	17970	53	Human	scRNA-seq
Lung healthy vs IPF (Fib)		Healthy vs IPF	45947	12753	49	Human	scRNA-seq
Lung healthy vs IPF (Imm)		Healthy vs IPF	45947	208774	60	Human	scRNA-seq
Lung healthy vs COPD (Epi)		Healthy vs COPD	45947	7888	38	Human	scRNA-seq
Lung healthy vs COPD (Fib)		Healthy vs COPD	45947	5286	36	Human	scRNA-seq
Lung healthy vs COPD (Imm)		Healthy vs COPD	45947	149875	46	Human	scRNA-seq
UC healthy vs non-inflamed (Epi)	Smillie *et al.* (2019)	Healthy vs non-inflamed	20028	99962	30	Human	scRNA-seq
UC healthy vs non-inflamed (Fib)		Healthy vs non-inflamed	19076	21627	30	Human	scRNA-seq
UC healthy vs non-inflamed (Imm)		Healthy vs non-inflamed	19076	118784	30	Human	scRNA-seq
UC inflamed vs non-inflamed (Epi)		Inflamed vs non-inflamed	20028	72748	35	Human	scRNA-seq
UC inflamed vs non-inflamed (Fib)		Inflamed vs non-inflamed	19076	23392	36	Human	scRNA-seq
UC inflamed vs non-inflamed (Imm)		Inflamed vs non-inflamed	20529	159242	36	Human	scRNA-seq
Melanoma pre-treatment	Sade-Feldman *et al.* (2019)	Responder vs non-responder	50513	5925	19	Human	scRNA-seq
Melanoma post-treatment		Responder vs non-responder	50513	10357	29	Human	scRNA-seq
COVID	Schulte-Schrepping *et al.* (2020a)	Mild vs severe	24794	48069	27	Human	scRNA-seq
Amyotrophic lateral sclerosis	Maniatis *et al.* (2019)	ALS vs normal	9129	23373	33	Mouse	Spatial transcriptomics
Triple negative breast cancer	Keren *et al.* (2019)	Survival period	38	199817	39	Human	Spatial proteomics

Supplementary Notes: Association study of features with conditions

Contents

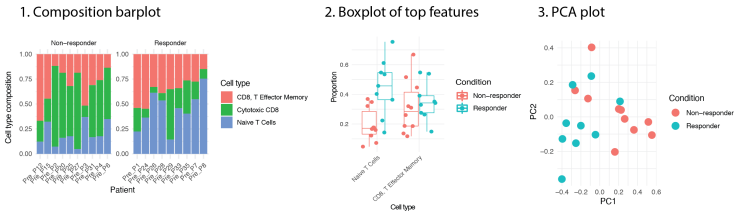
This file runs association study using the given features and sample conditions and plots the key features from each feature category using a representative figure. The purpose is not to provide a comprehensive analysis in a single HTML but to help point directions for future investigation.

Overview of the association study result

Here we provide a brief overview of the association study result, including the number of features in each feature type, and the number of features that are significantly associated (P-value < 0.1) with the conditions of the interest.

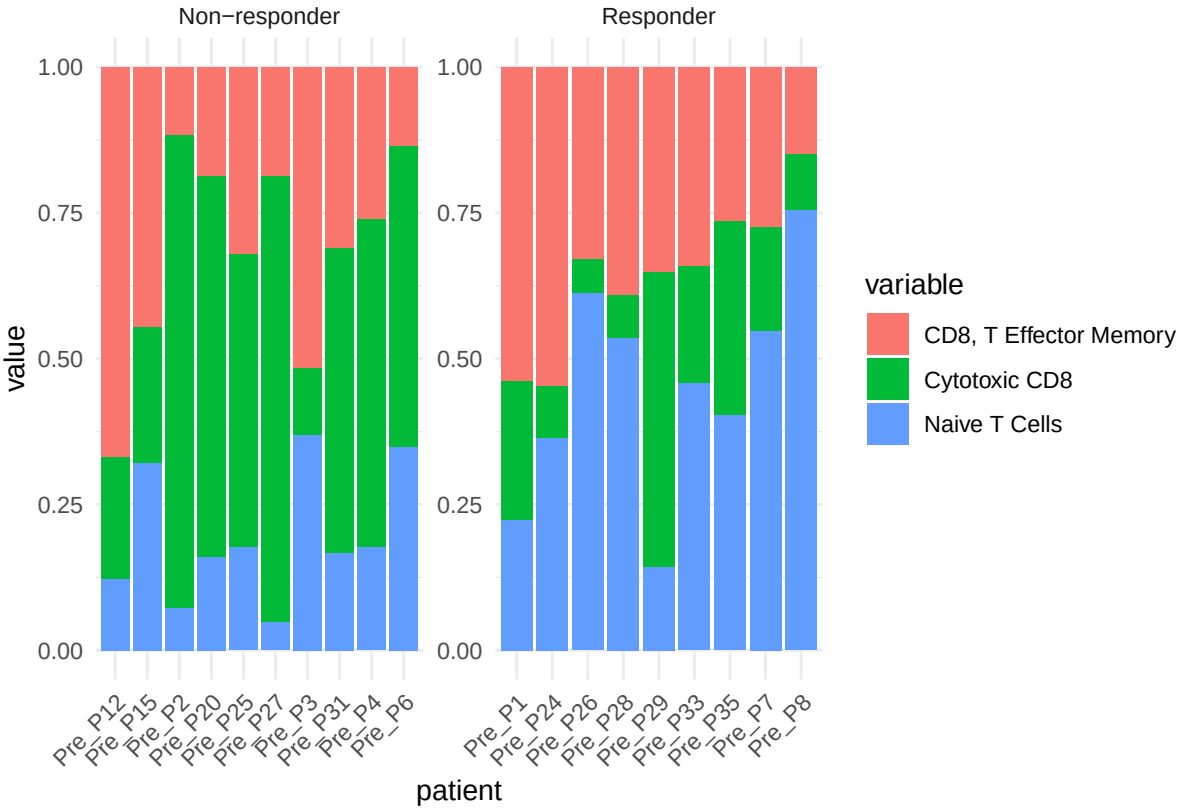
Cell type proportions

```
knitr::include_graphics( system.file("extdata/figure", "celltypeproportion_example_figures.png",    packa
```



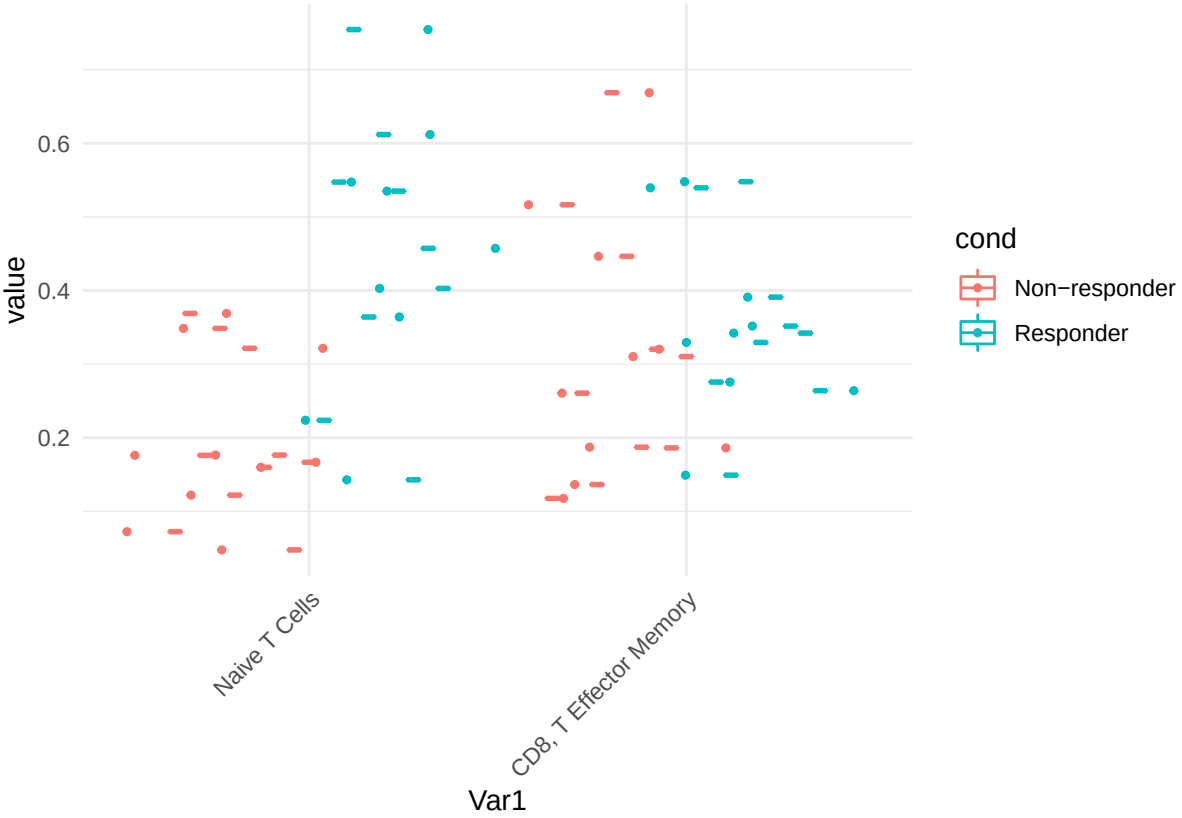
1. Barplot shows the composition of cells types
2. Boxplot shows the top cell types that differs between conditions
3. PCA plot shows the separation of conditions based on the cell type proportion features

Composition barplot

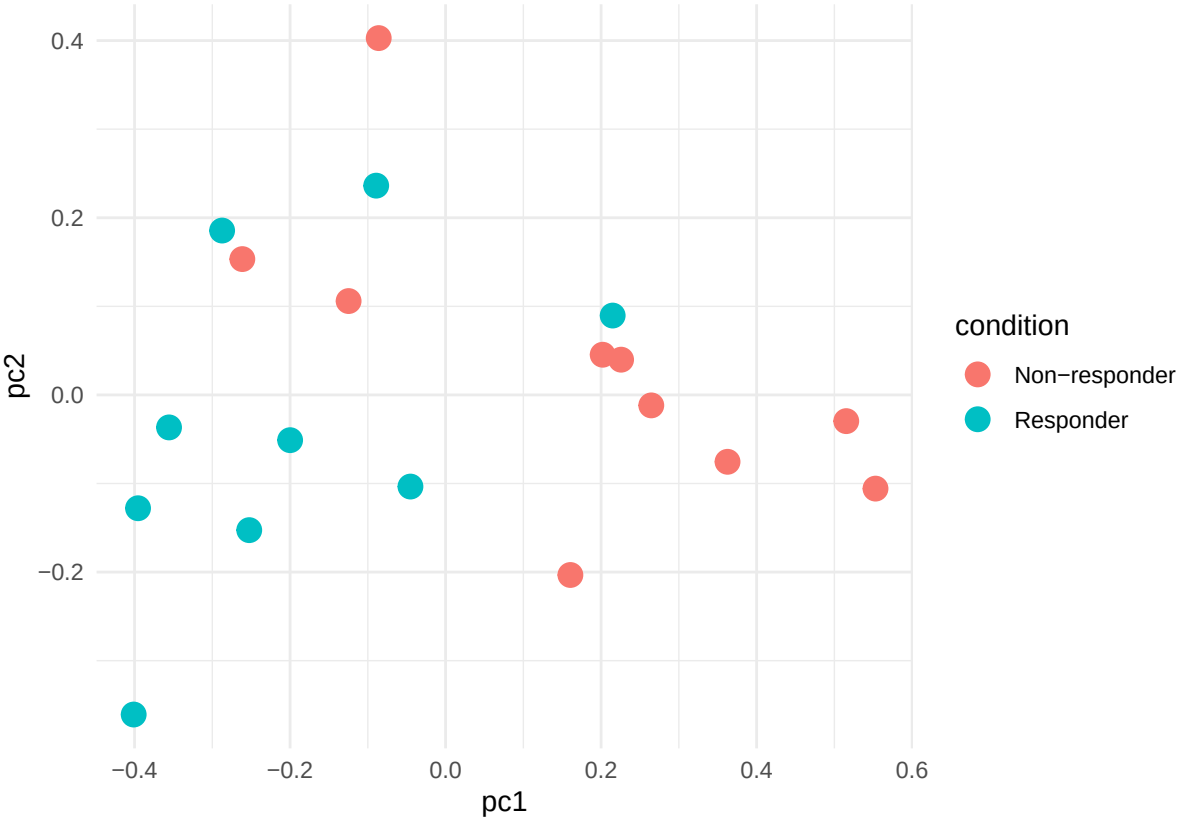


Boxplot of top features

```
## [1] "up regulated in Responder"
```

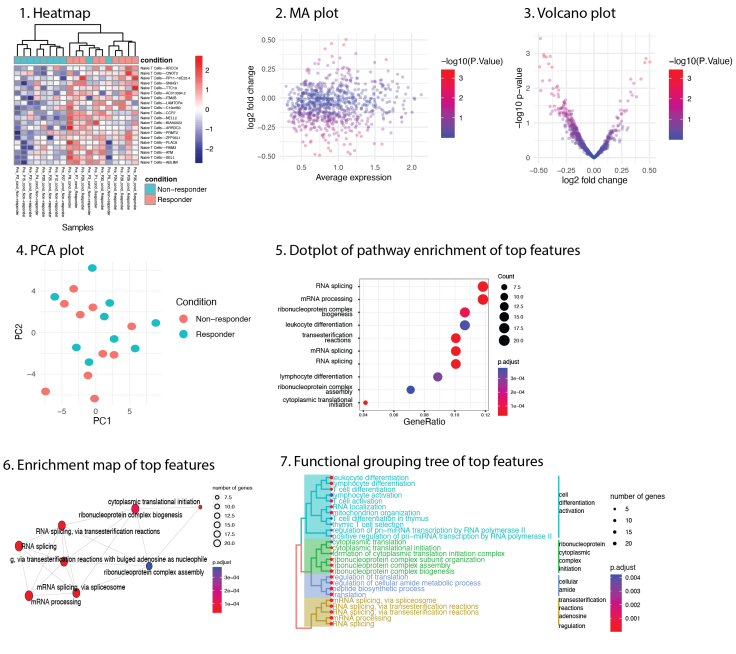


PCA plot



Cell type specific gene expressions

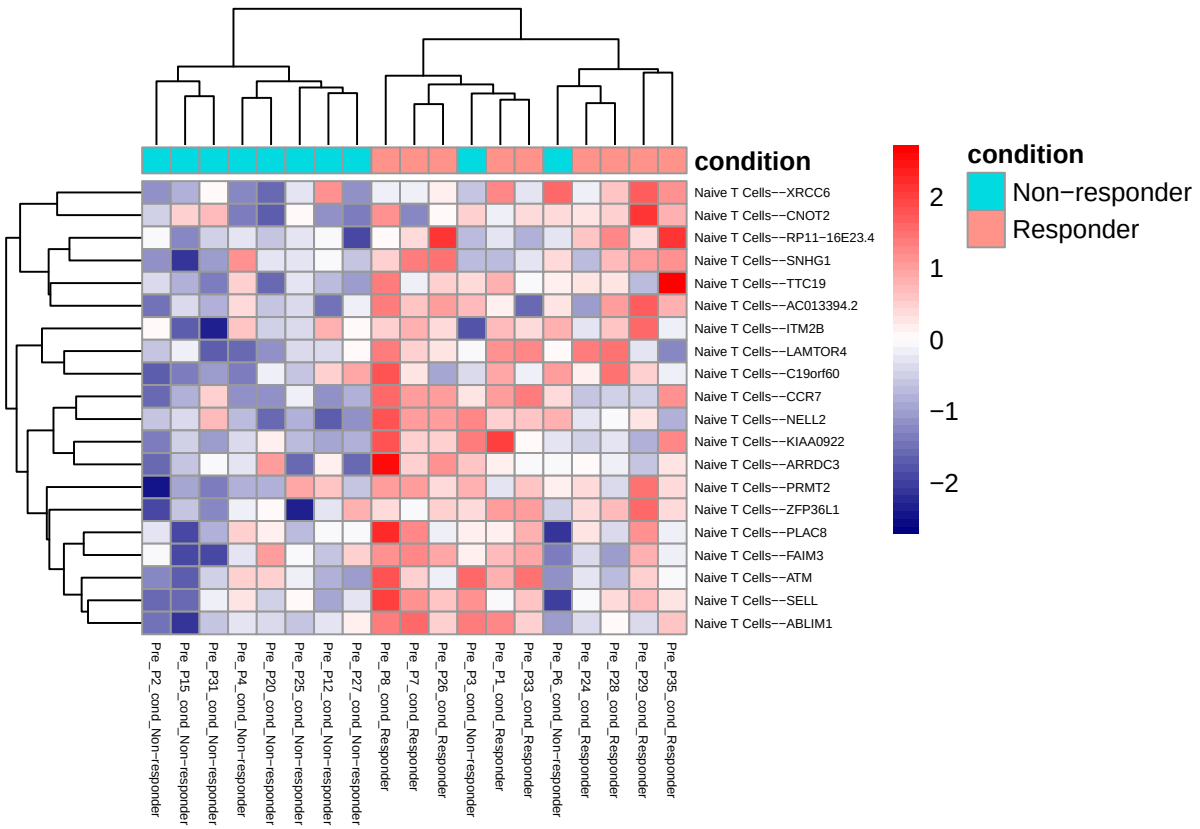
```
knitr::include_graphics( system.file("extdata/figure",  "celltypegene_example_figures.png" ,    package =
```



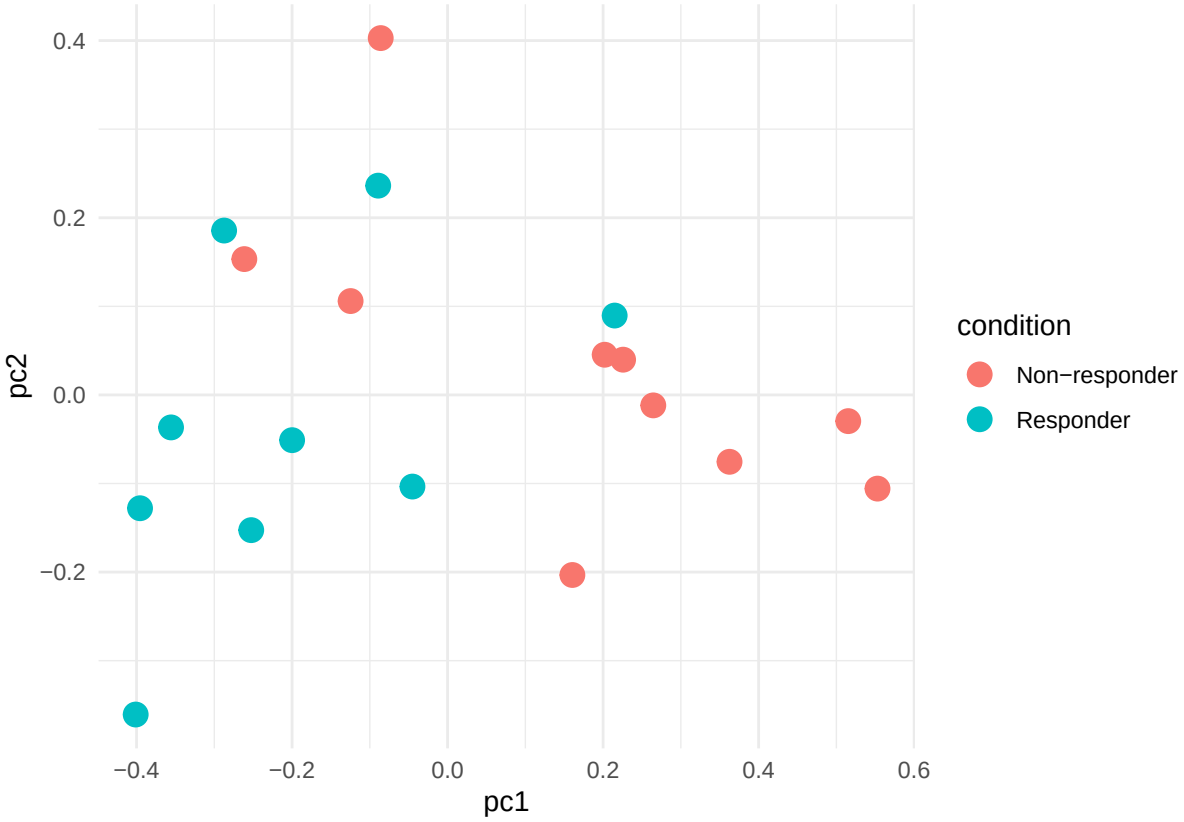
1. Heatmaps shows the top cell type specific gene expression features that differs between conditions
2. MA plot shows the expression and log2 fold change of the cell type specific gene expression features
3. Volcano plot shows the log2 fold change and P-values of the cell type specific gene expression features
4. PCA plot shows the separation of conditions based on the cell type specific gene expression features
5. Dot plot shows the pathway enrichment of the top cell type specific gene expression features that differs between conditions
6. Enrichment map of the top cell type specific gene expression features that differs between conditions
7. Functional grouping of the top cell type specific gene expression features that differs between conditions

Heatmap

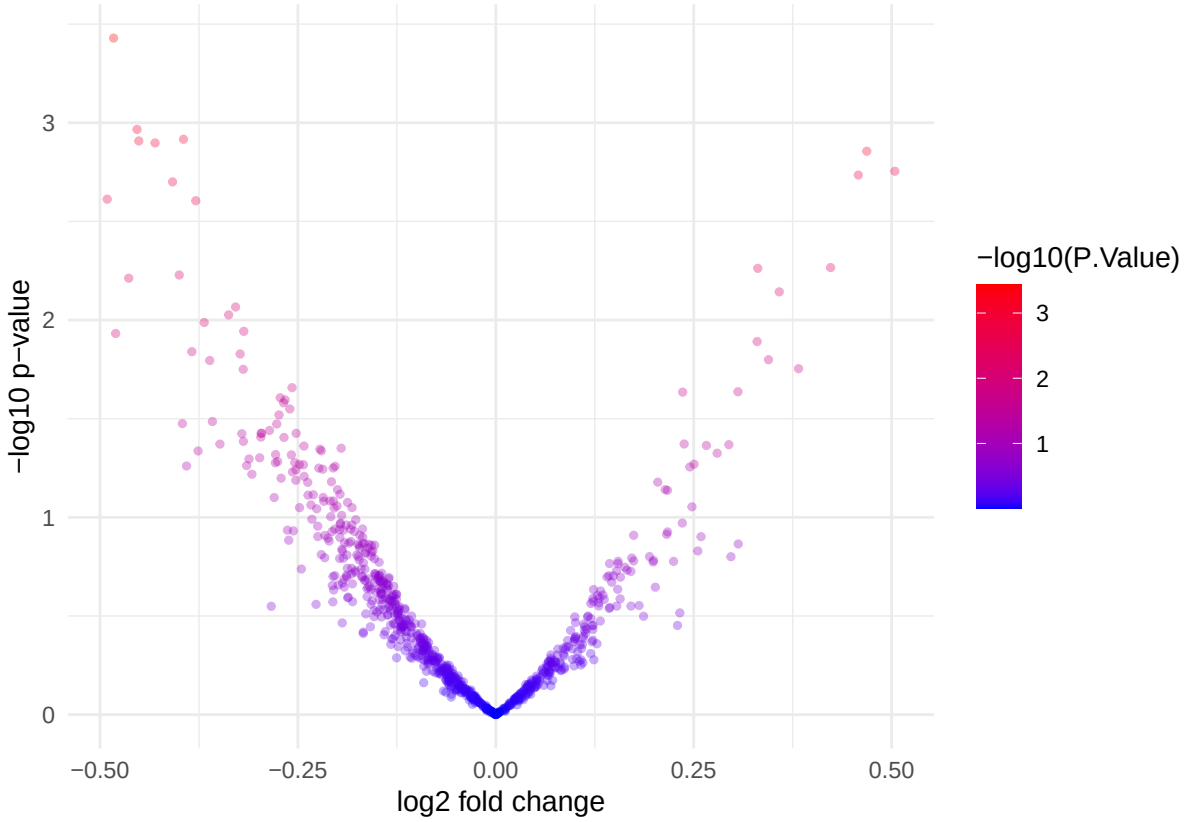
```
## [1] "up regulated in Responder"
```



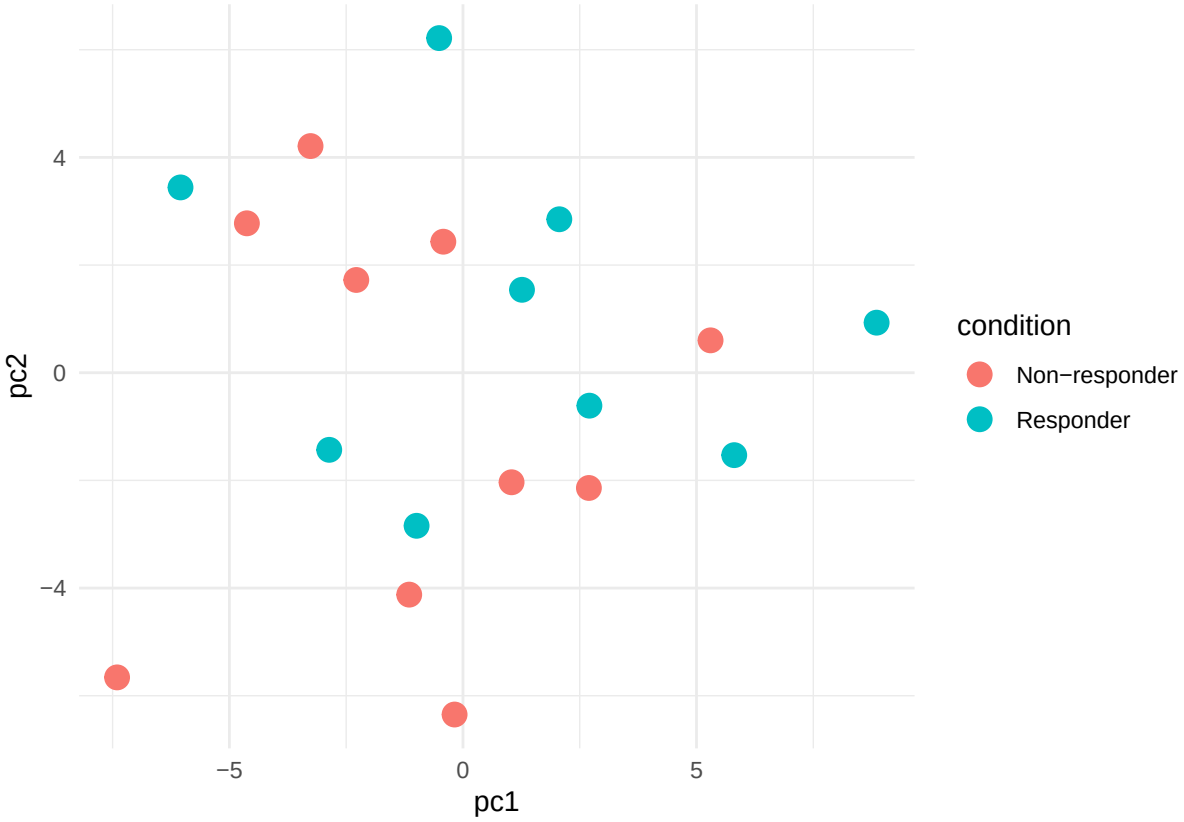
MA plot


0.4
0.2
0.0
−0.2
pc2
−0.4
−0.2
0.0
0.2
0.4
0.6
pc1
condition
Non−responder
Responder

Volcano plot

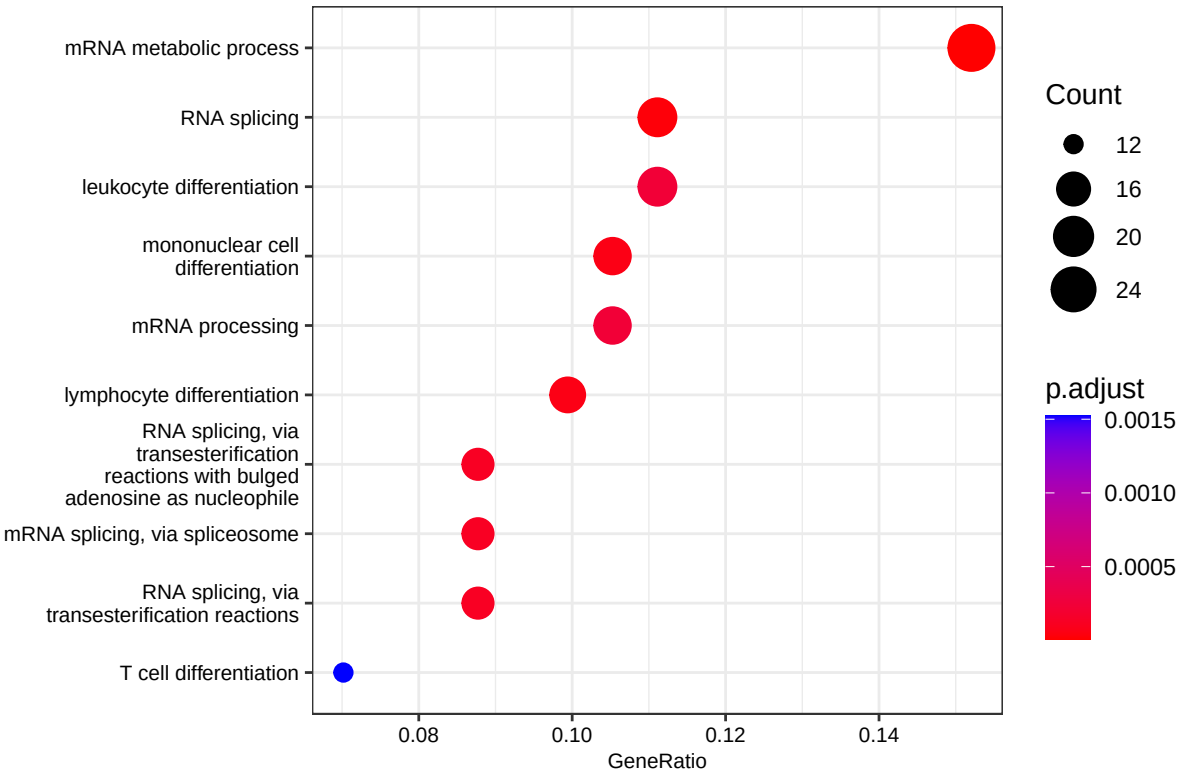


PCA plot

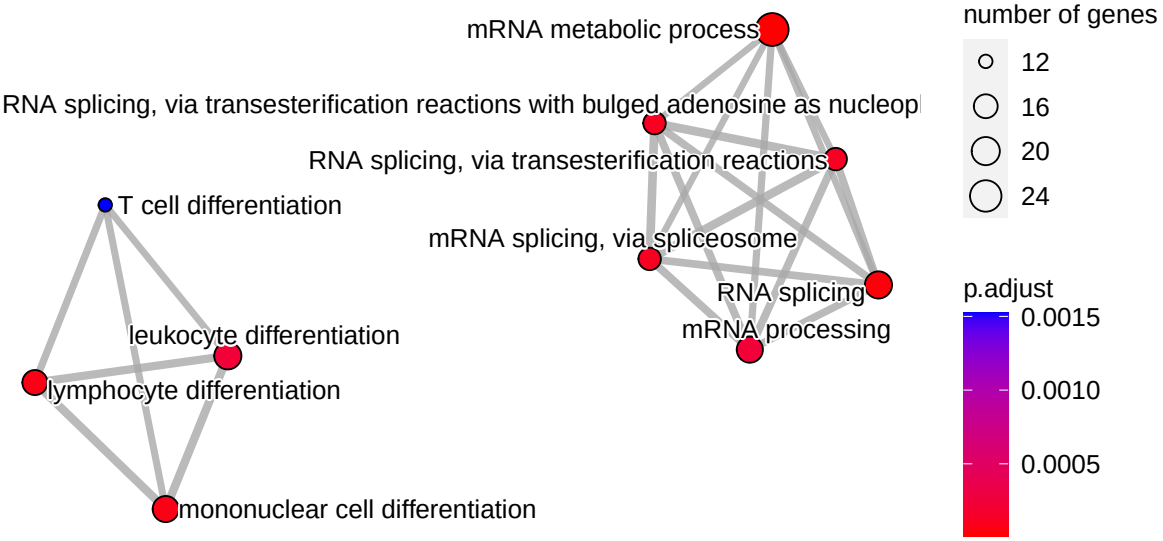


Dot plot

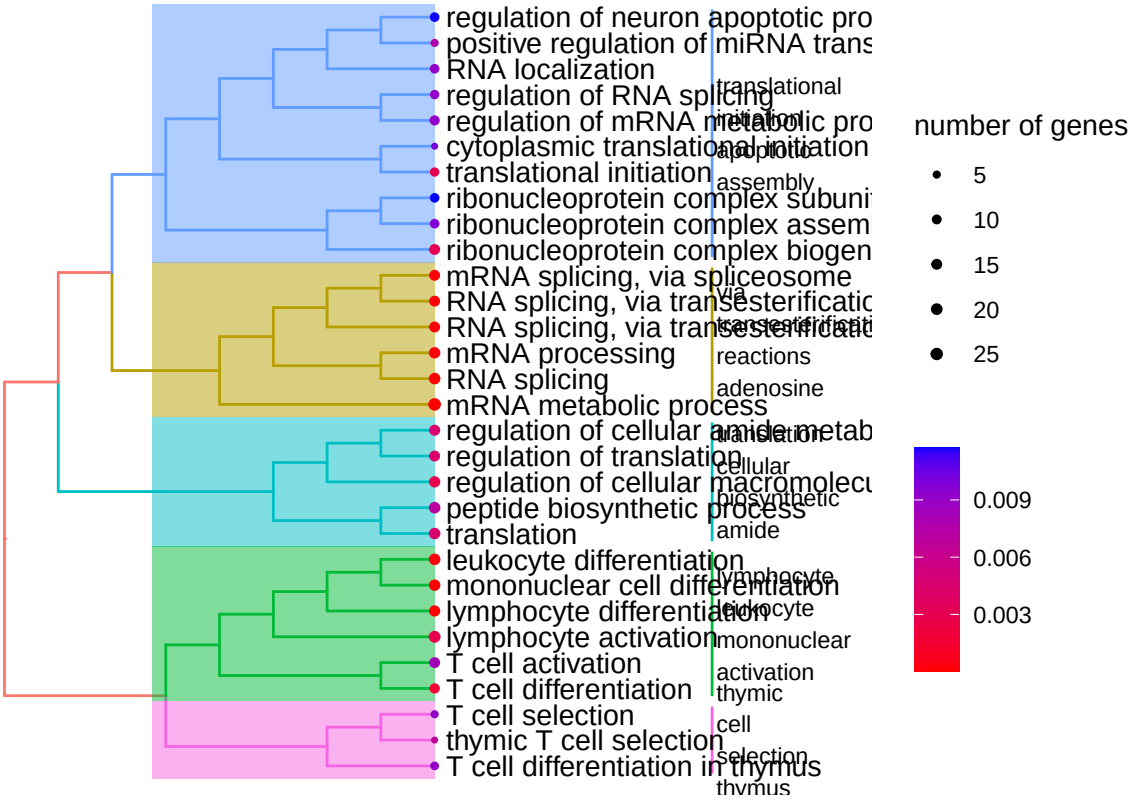
```
## [1] "up regulated"
```



Enrichment map

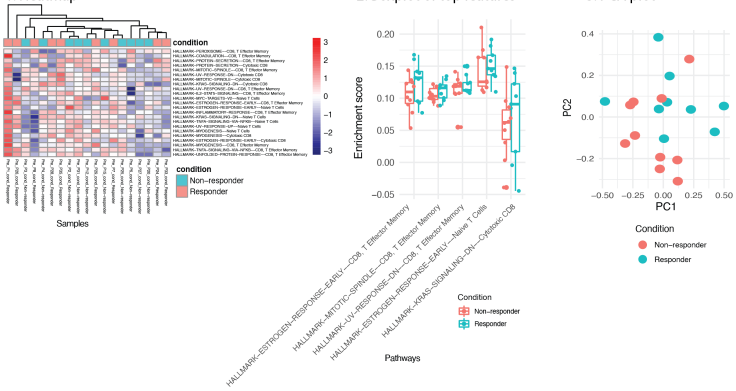


Functional grouping



Cell type specific pathway expressions

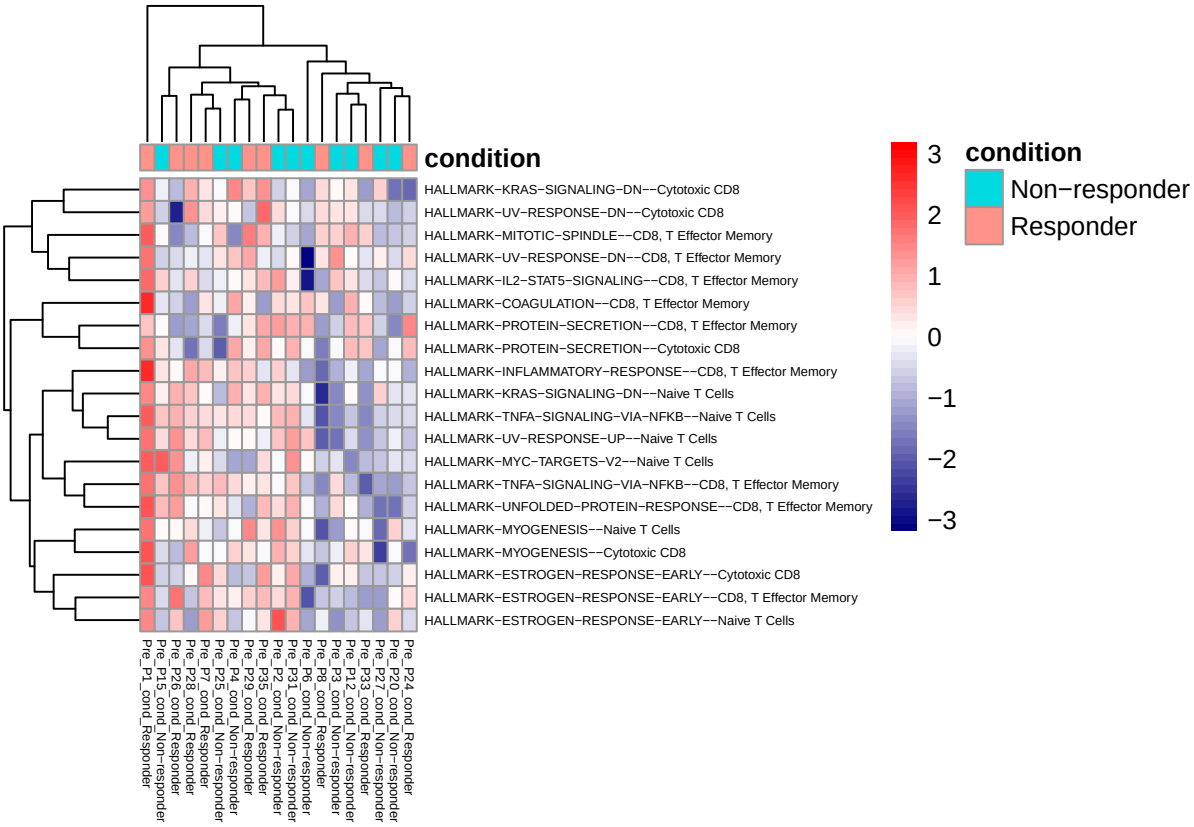
```
knitr::include_graphics(  system.file("extdata/figure",  "pathway_example_figures.png" ,   package = "s
```



1. Heatmaps shows the top cell type specific pathway expression features that differs between conditions
2. Boxplot shows the top cell type specific pathway expression features that differs between conditions
3. PCA plot shows the separation of conditions based on the cell type specific pathway expression features

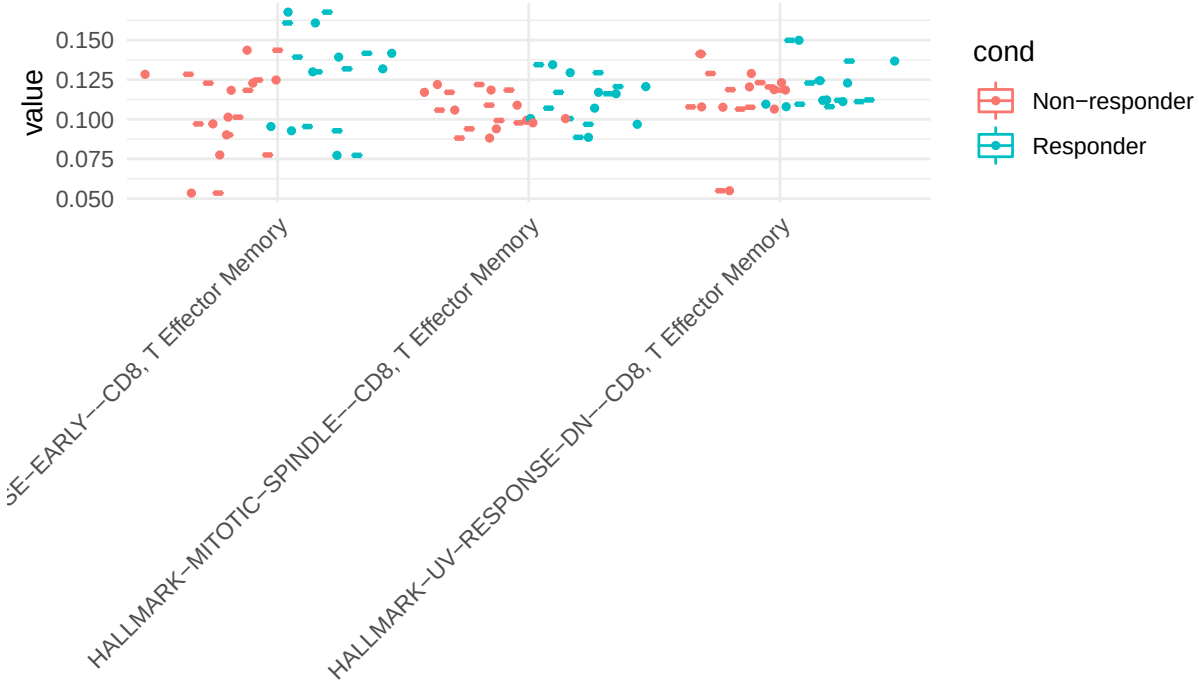
Heatmap

```
## [1] "up regulated in Responder"
```



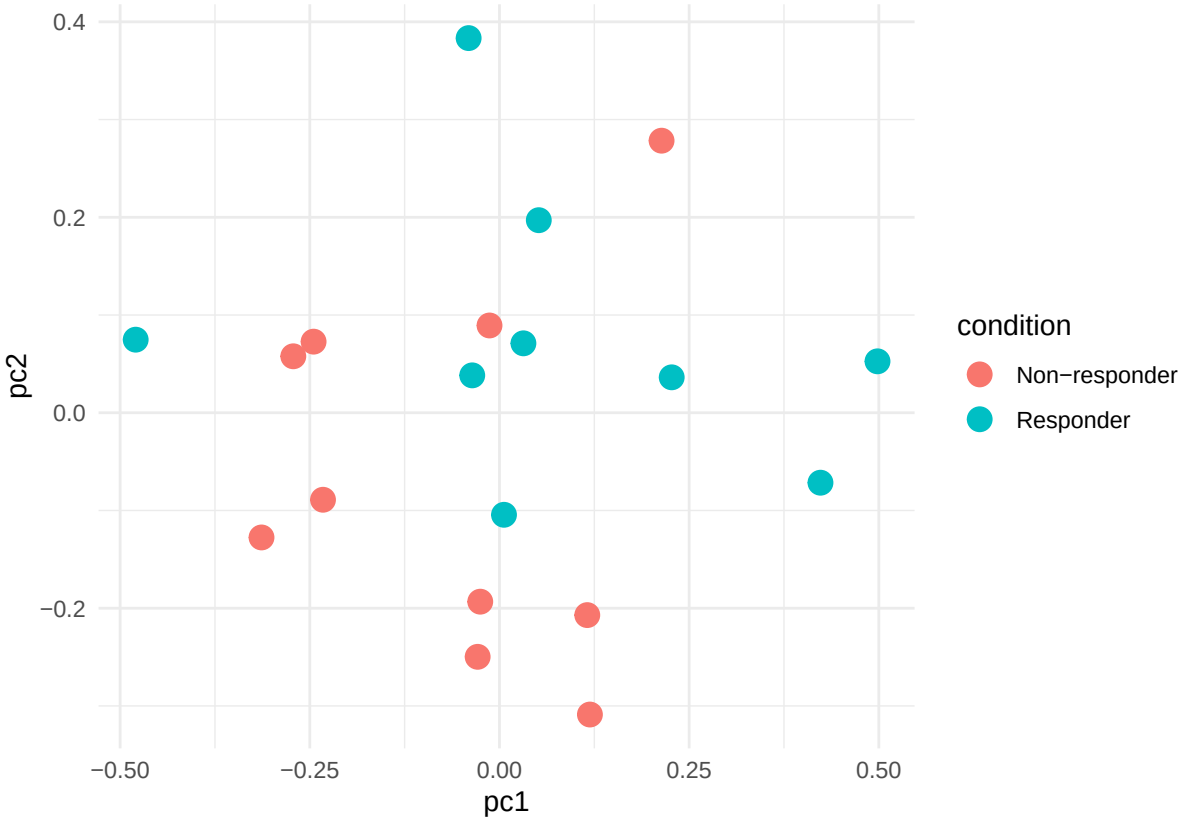
Boxplot

```
## [1] "up regulated in Responder"
```


value
0.150
0.125
0.100
0.075
0.050
cond
Non−responder
Responder
5E−EARLY−−CD8, T Effector Memory
HALLMARK−MITOTIC−SPINDLE−−CD8, T Effector Memory
HALLMARK−UV−RESPONSE−DN−−CD8, T Effector Memory

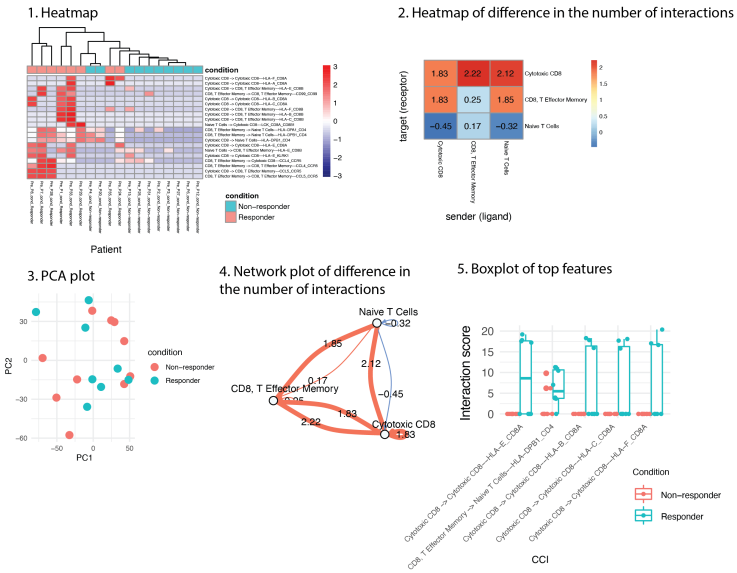
Var1

PCA plot



Cell type specific cell-cell communications

```
knitr::include_graphics(  system.file("extdata/figure",
                  "CCI_example_figures.png"   ,   package = "scFeatures")    )
```



1. Heatmap shows the top cell cell interactions features that differs between conditions
2. Heatmap shows the difference in the number of interactions between conditions

For each interacting cell type, the difference is calculated as:

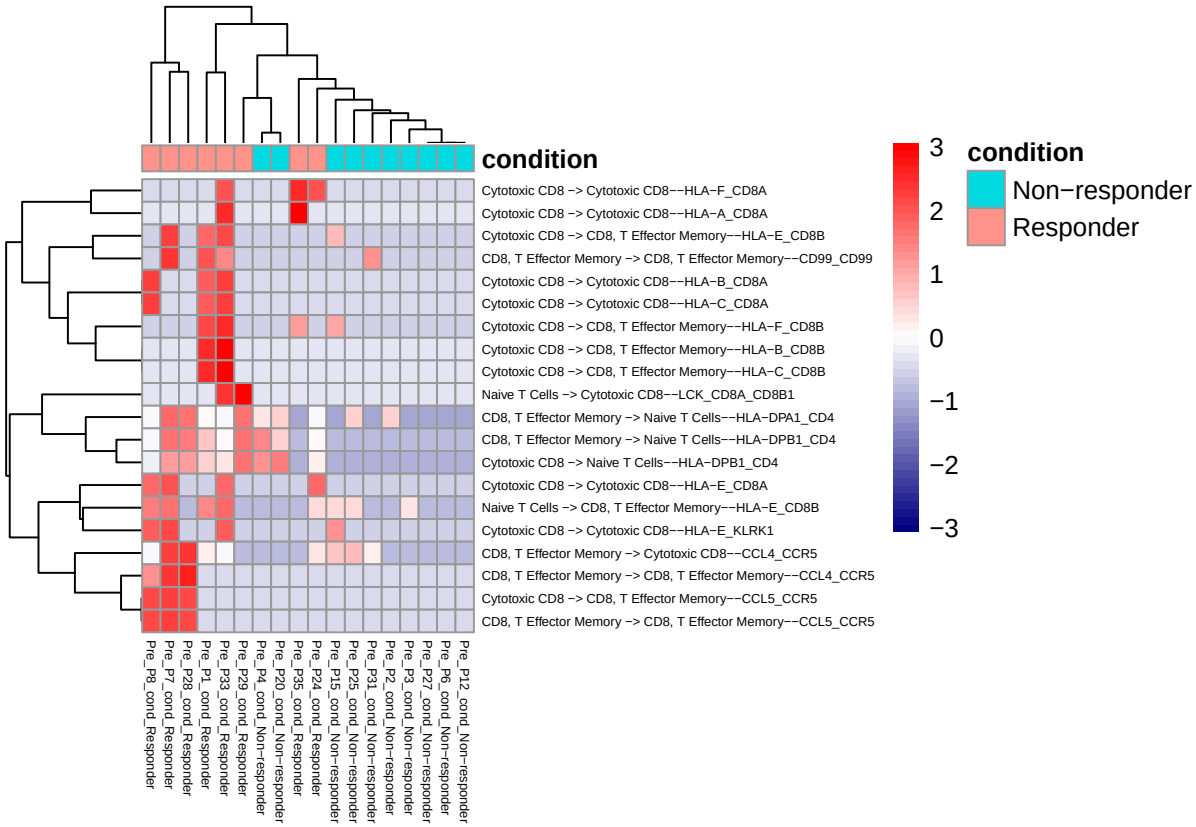
$$\frac{total\ number\ of\ non-zero\ interactions\ in\ condition1\ patients}{number\ of\ condition1\ patients} - \frac{total\ number\ of\ non-zero\ interactions\ in\ condition2\ pat}{number\ of\ condition2\ patients}$$

3. PCA plot shows the separation of conditions based on the cell type specific pathway expression features

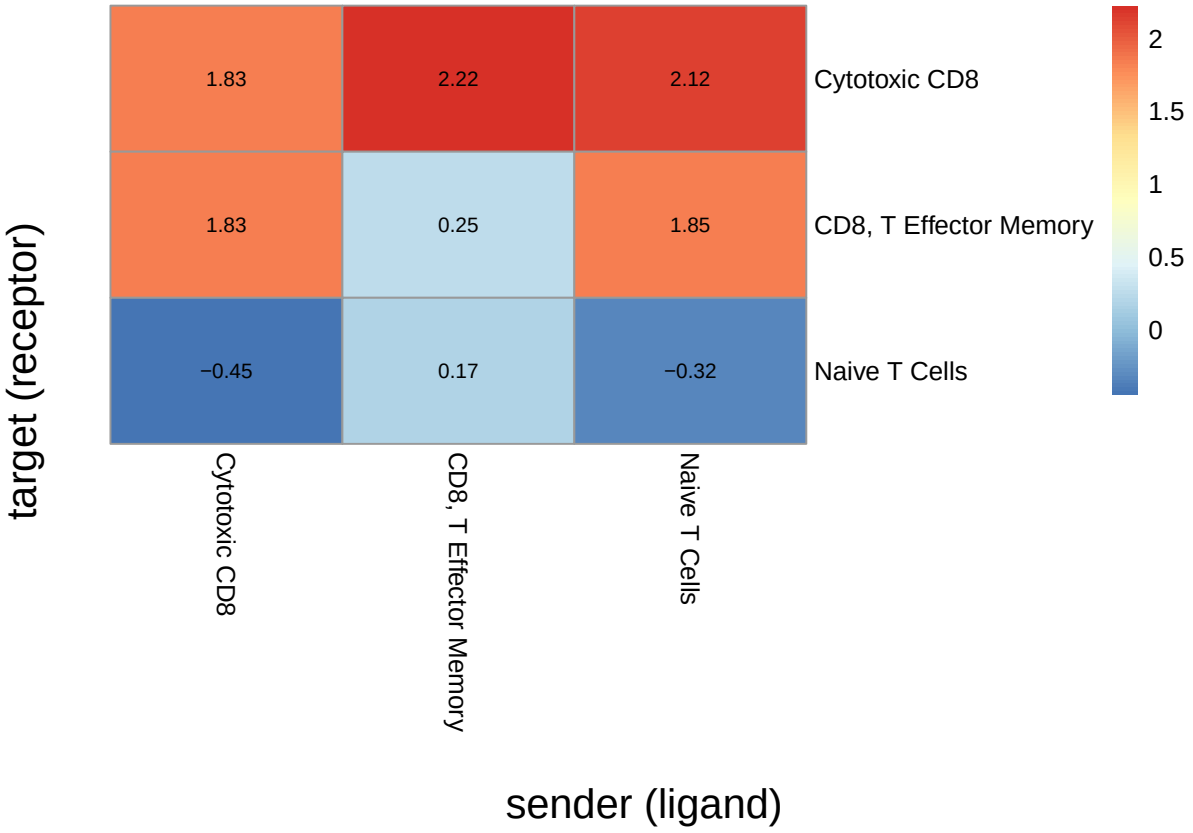
4. Network plot shows the difference in the number of interactions between conditions
5. Boxplot shows the top cell cell interaction features that differs between conditions

Heatmap of top cell cell interactions

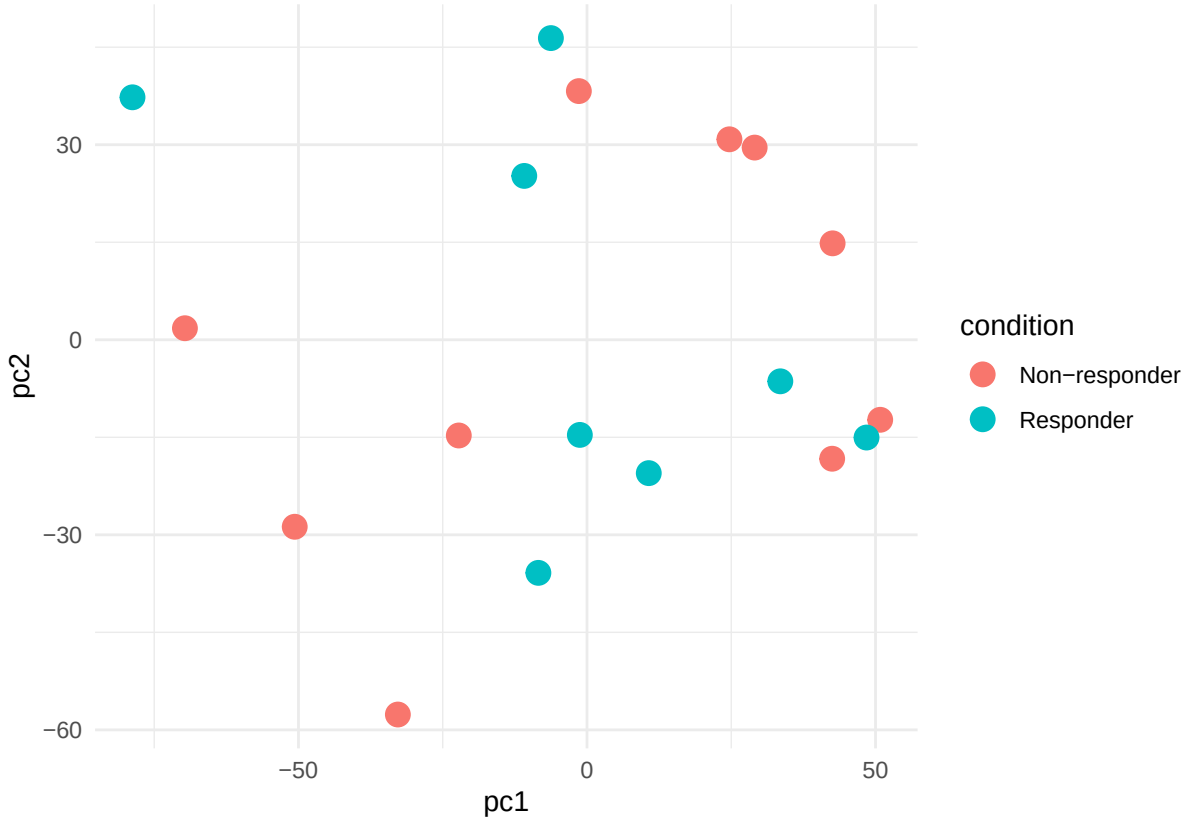
```
## [1] "up regulated in Responder"
```



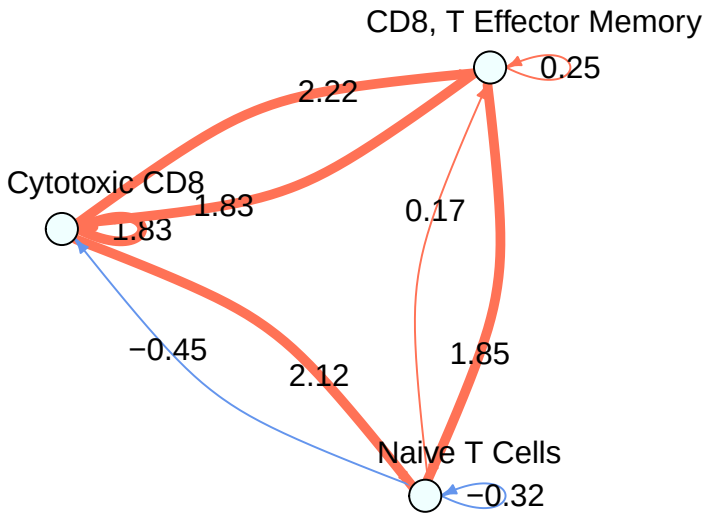
Heatmap of difference in number of interactions



PCA plot

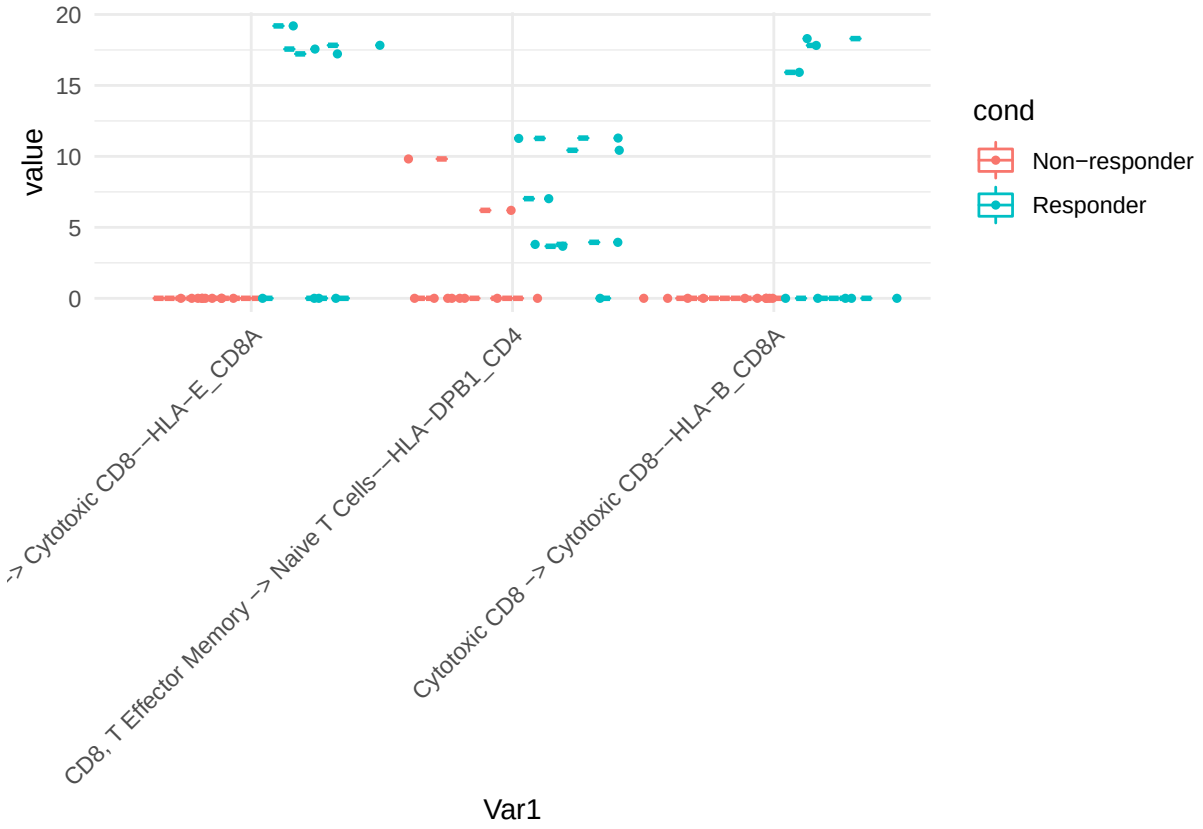


Network plot



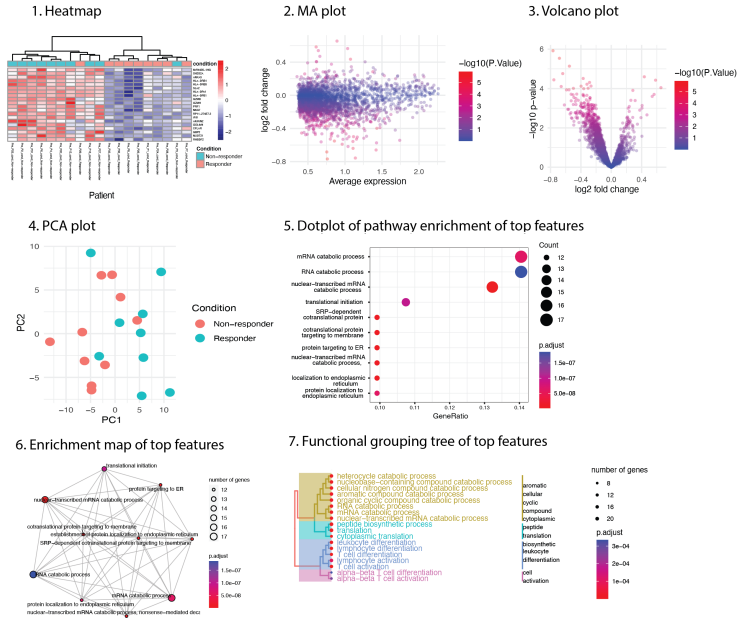
Boxplot

```
## [1] "up regulated in Responder"
```



Overall aggregated gene expressions

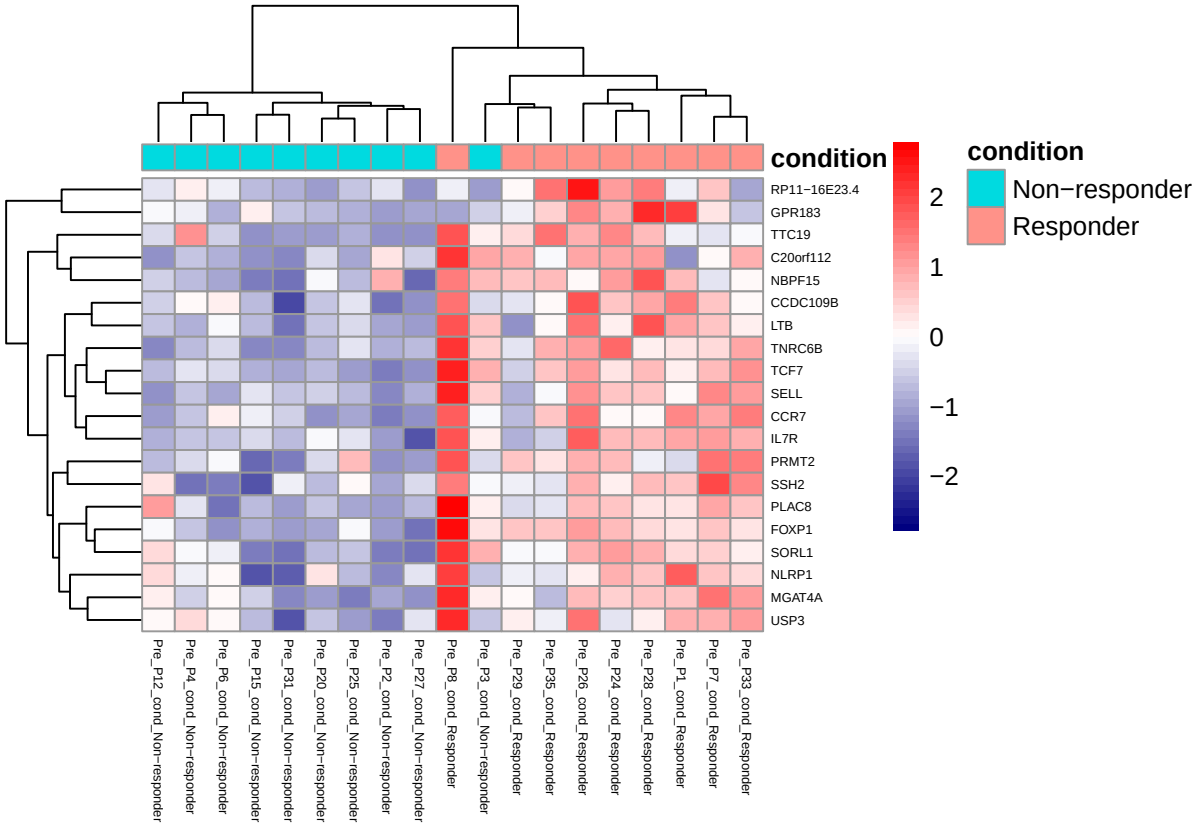
```
knitr::include_graphics(  system.file("extdata/figure",
        "aggregatedgene_example_figures.png" ,   package = "scFeatures")    )
```



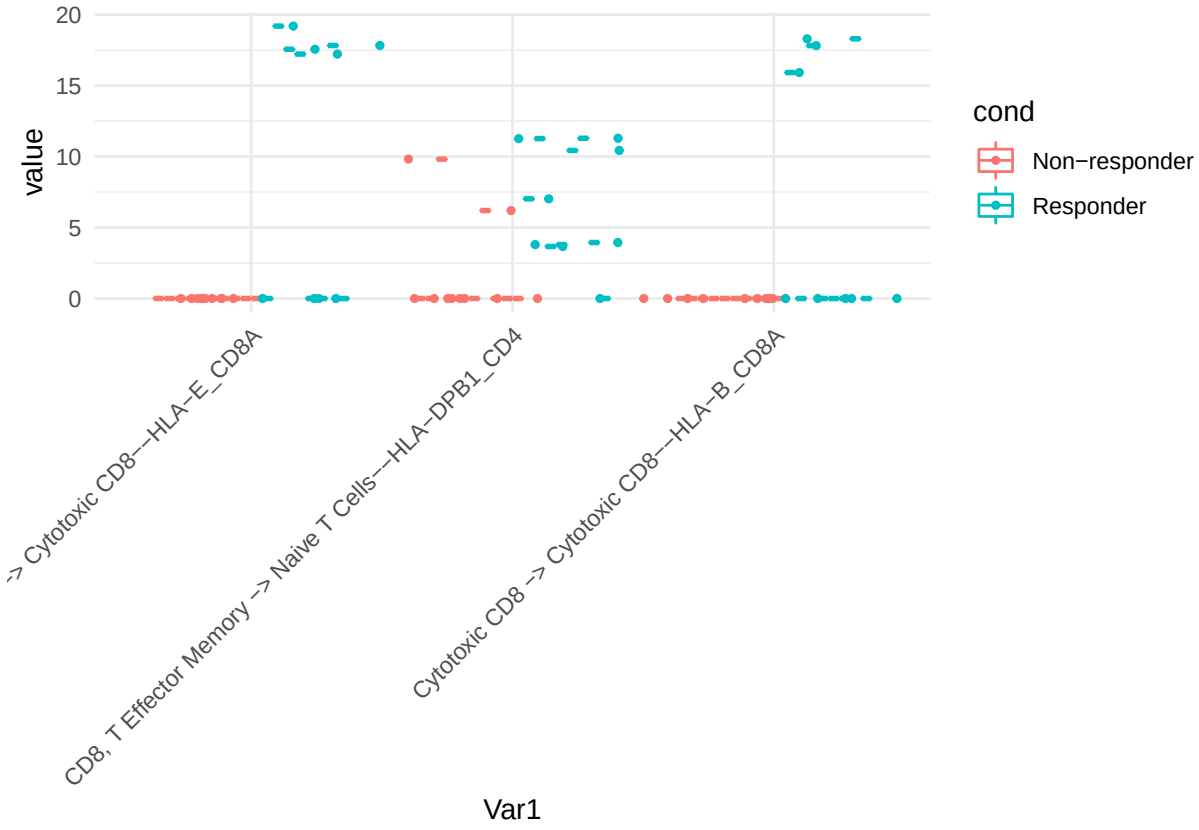
1. Heatmaps shows the top aggregated gene expression features that differs between conditions
2. MA plot shows the expression and log2 fold change of the aggregated gene expression features
3. Volcano plot shows the log2 fold change and P-values of the aggregated gene expression features
4. PCA plot shows the separation of conditions based on the aggregated gene expression features
5. Dot plot shows the pathway enrichment of the top aggregated gene expression features that differs between conditions
6. Enrichment map of the top aggregatedgene expression features that differs between conditions
7. Functional grouping of the top aggregated gene expression features that differs between conditions

Heatmap

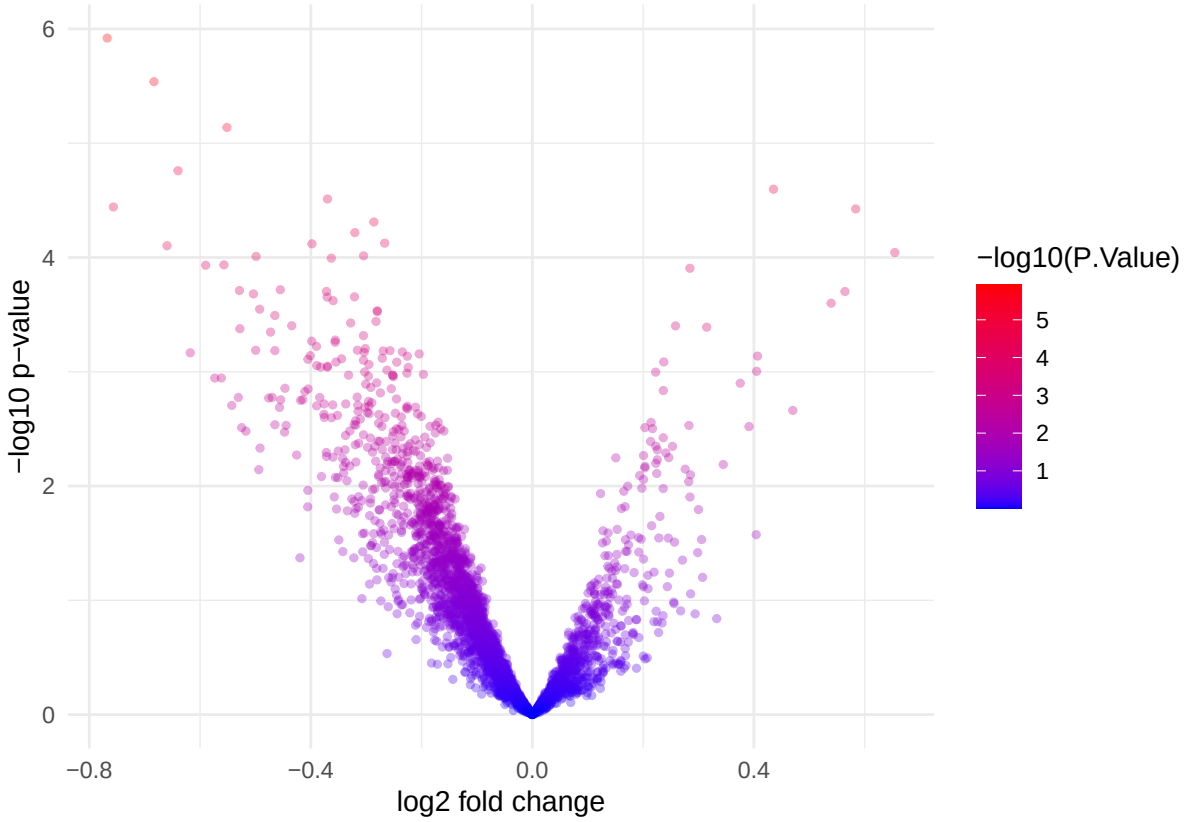
```
## [1] "up regulated in Responder"
```



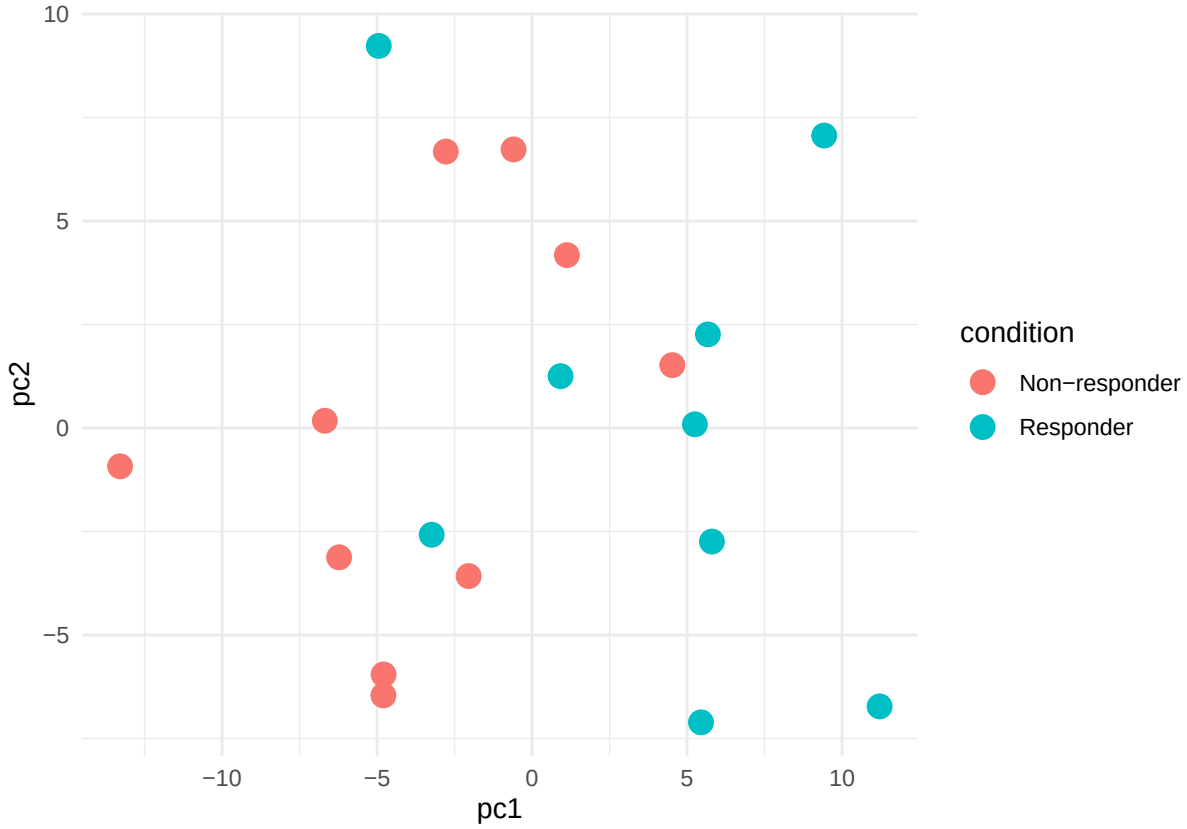
MA plot


20
15
10
5
0
value
-> Cytotoxic CD8--HLA-E_CD8A
CD8, T Effector Memory -> Naive T Cells--HLA-DPB1_CD4
Cytotoxic CD8 -> Cytotoxic CD8--HLA-B_CD8A
Var1
cond
Non−responder
Responder

Volcano plot

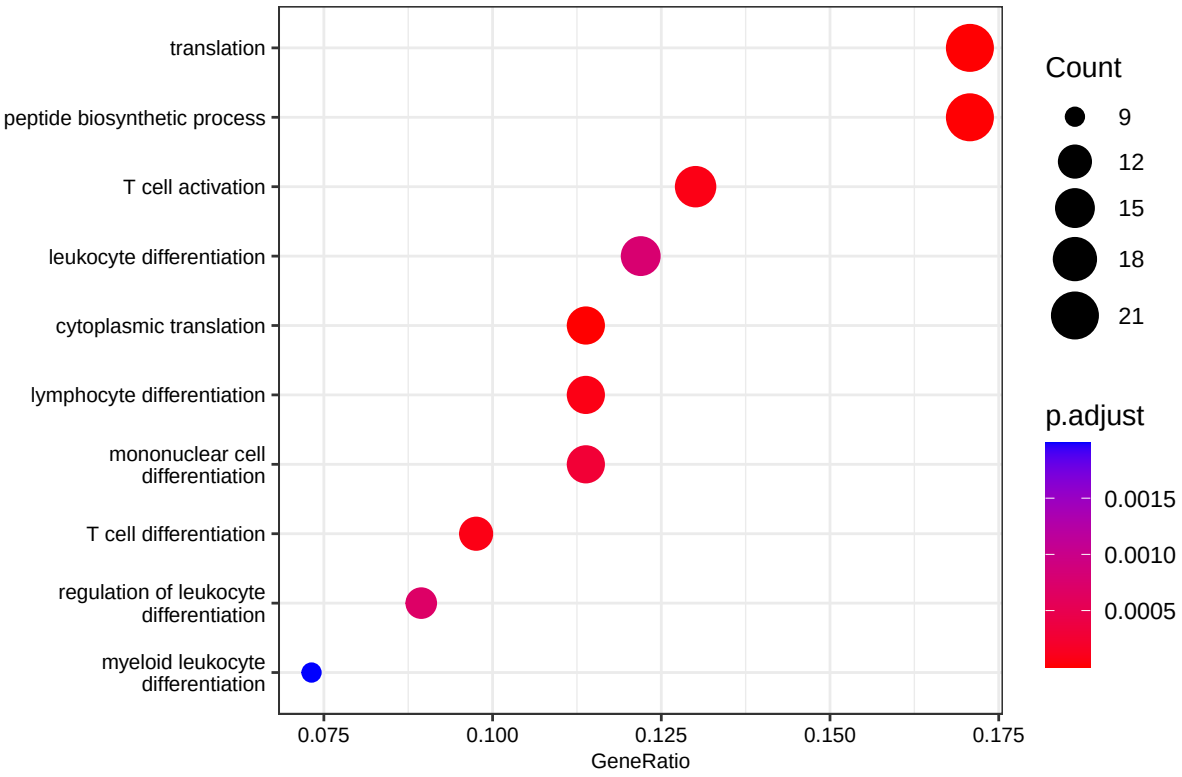


PCA plot

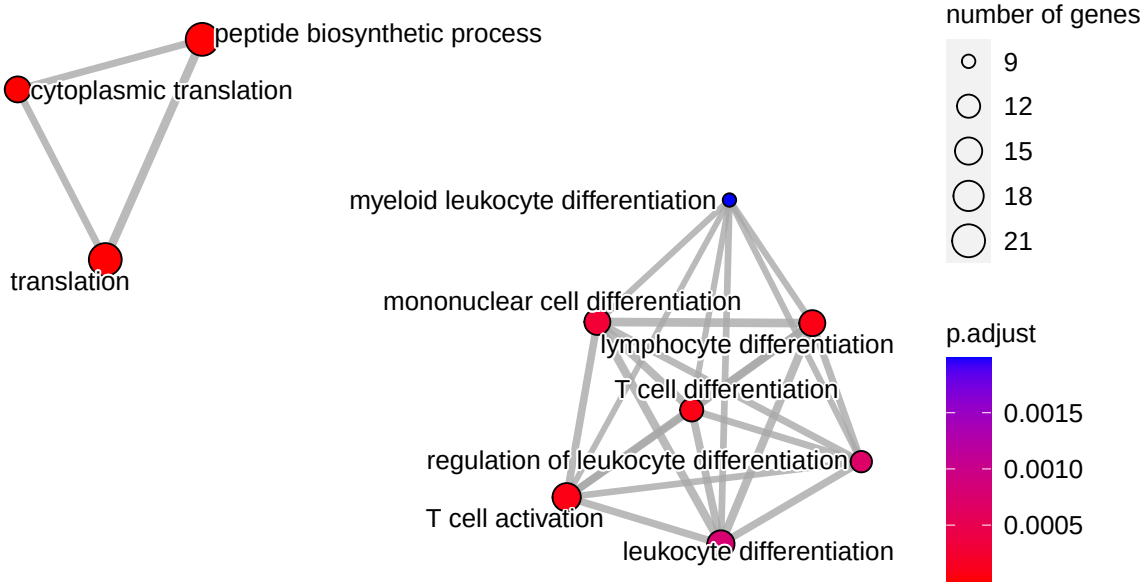


Dot plot

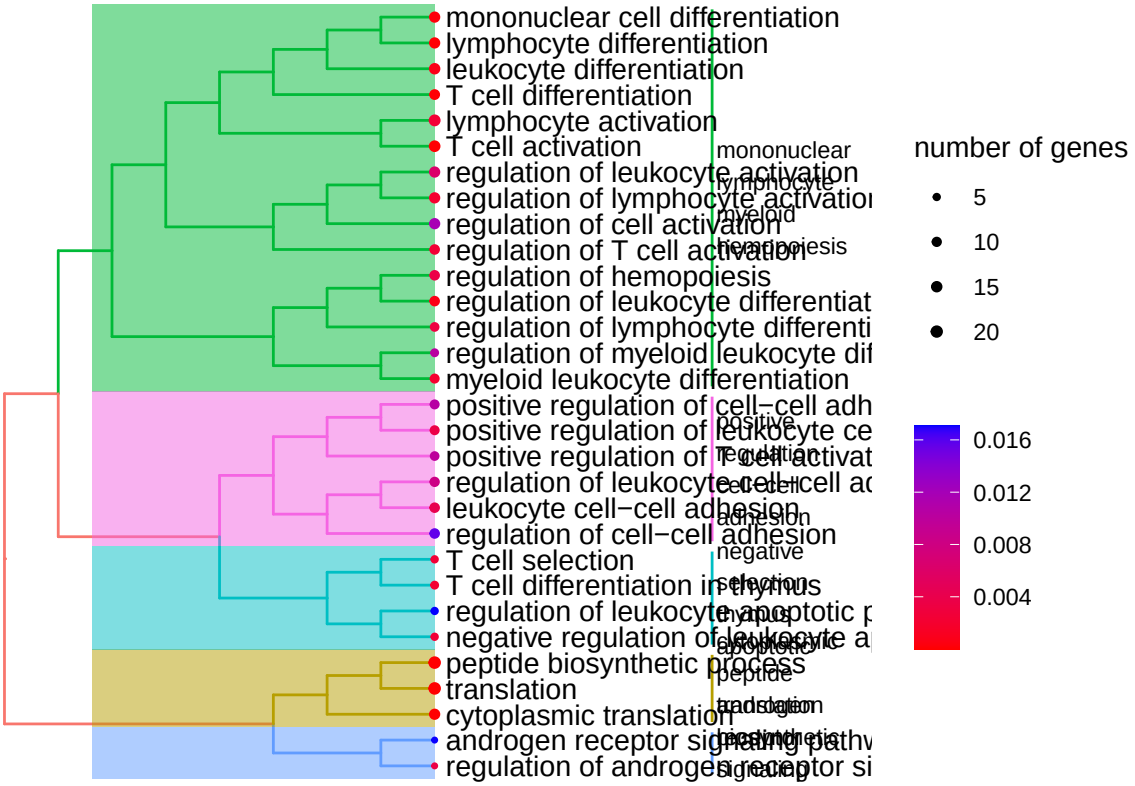
```
## [1] "up regulated"
```



Enrichment map

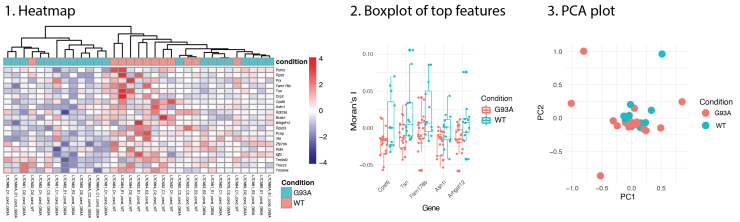


Functional grouping



Spatial metrics

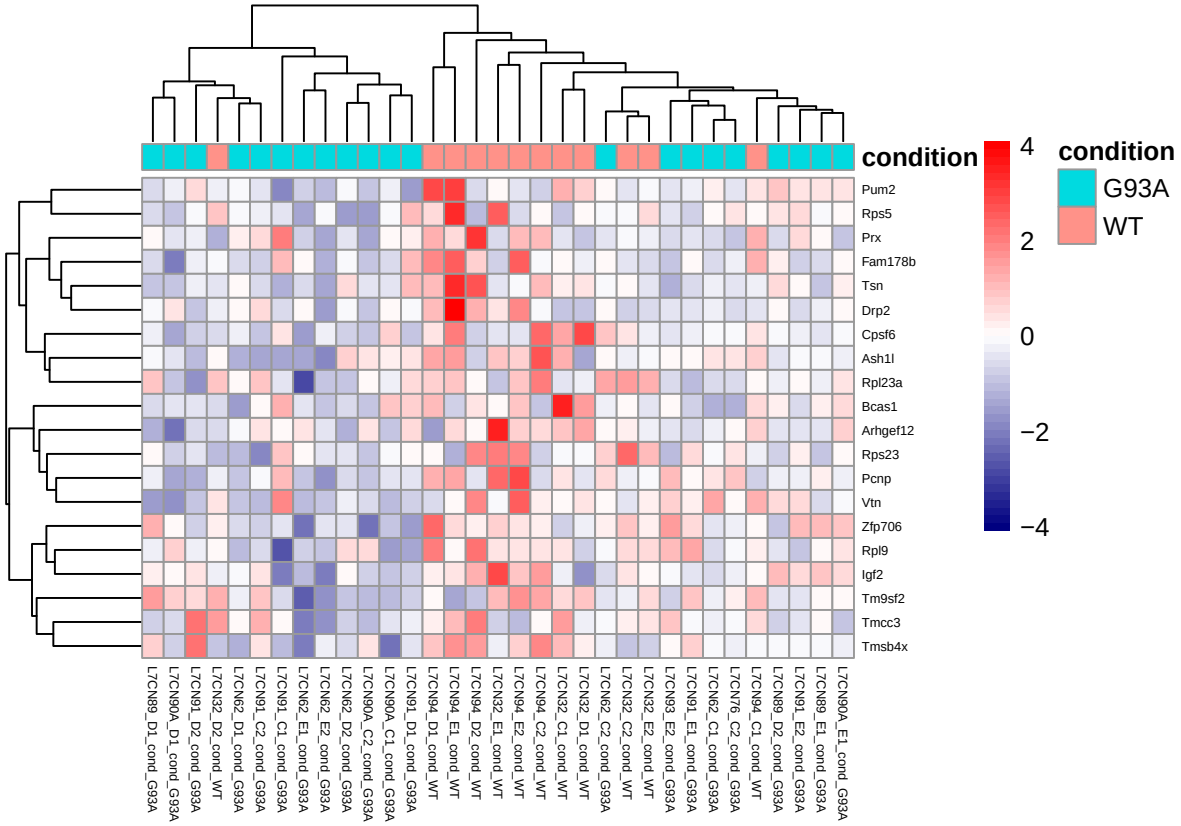
```
knitr::include_graphics(  system.file("extdata/figure",
       "spatial_example_figures.png",   package = "scFeatures")     )
```



1. Heatmaps shows the top spatial features that differs between conditions
2. Boxplot shows the top spatial features that differs between conditions
3. PCA plot shows the separation of conditions based on the spatial features

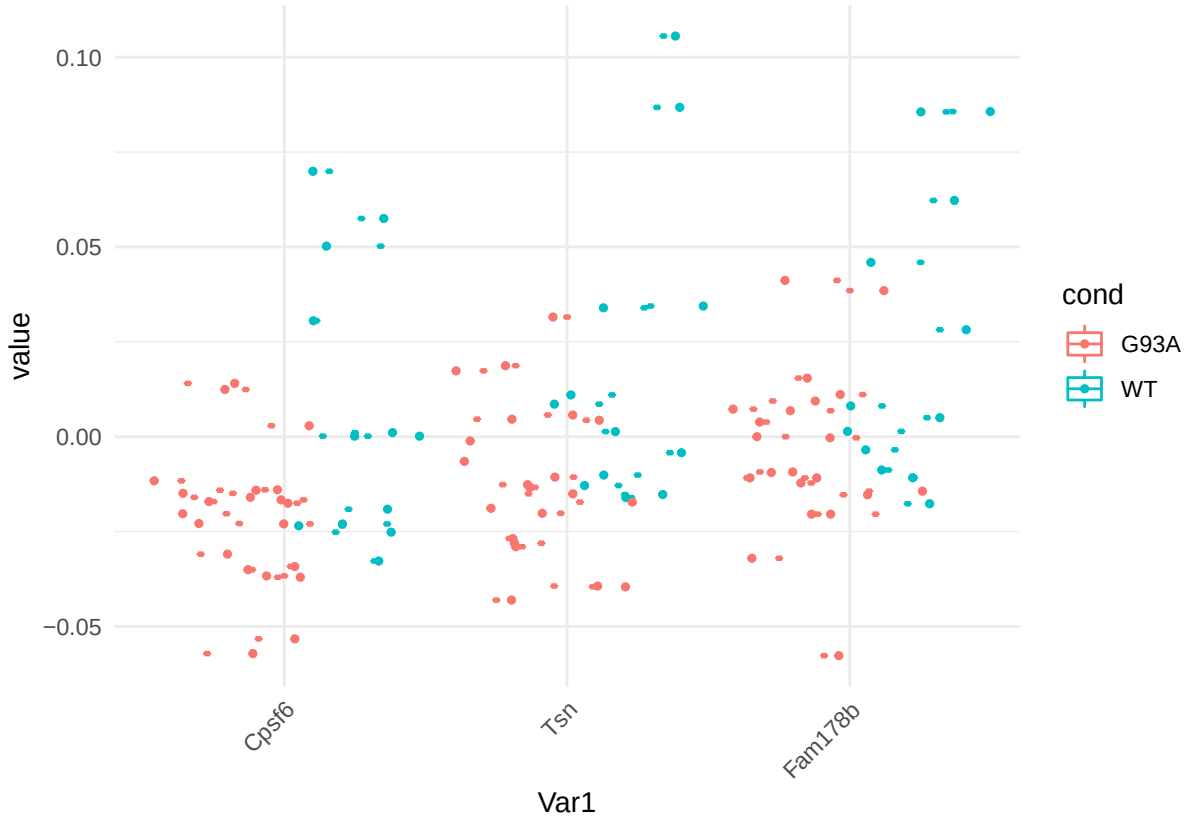
Heatmap

```
## [1] "up regulated in WT"
```

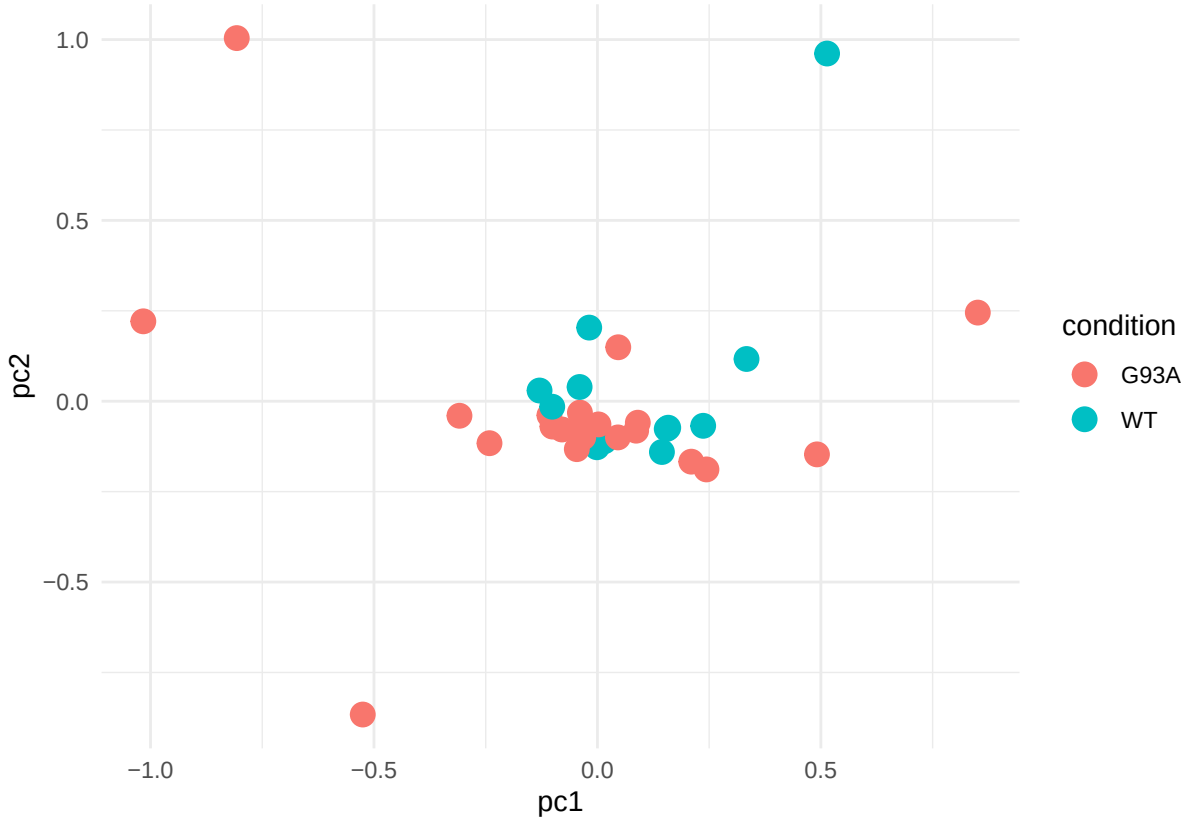


Boxplot

```
## [1] "up regulated in WT"
```


0.10
0.05
0.00
−0.05
value
Cpsf6
Tsn
Fam178b
Var1
cond
G93A
WT

PCA plot



C

APPENDIX FOR CHAPTER 5

C.1 SUPPLEMENTARY FIGURES

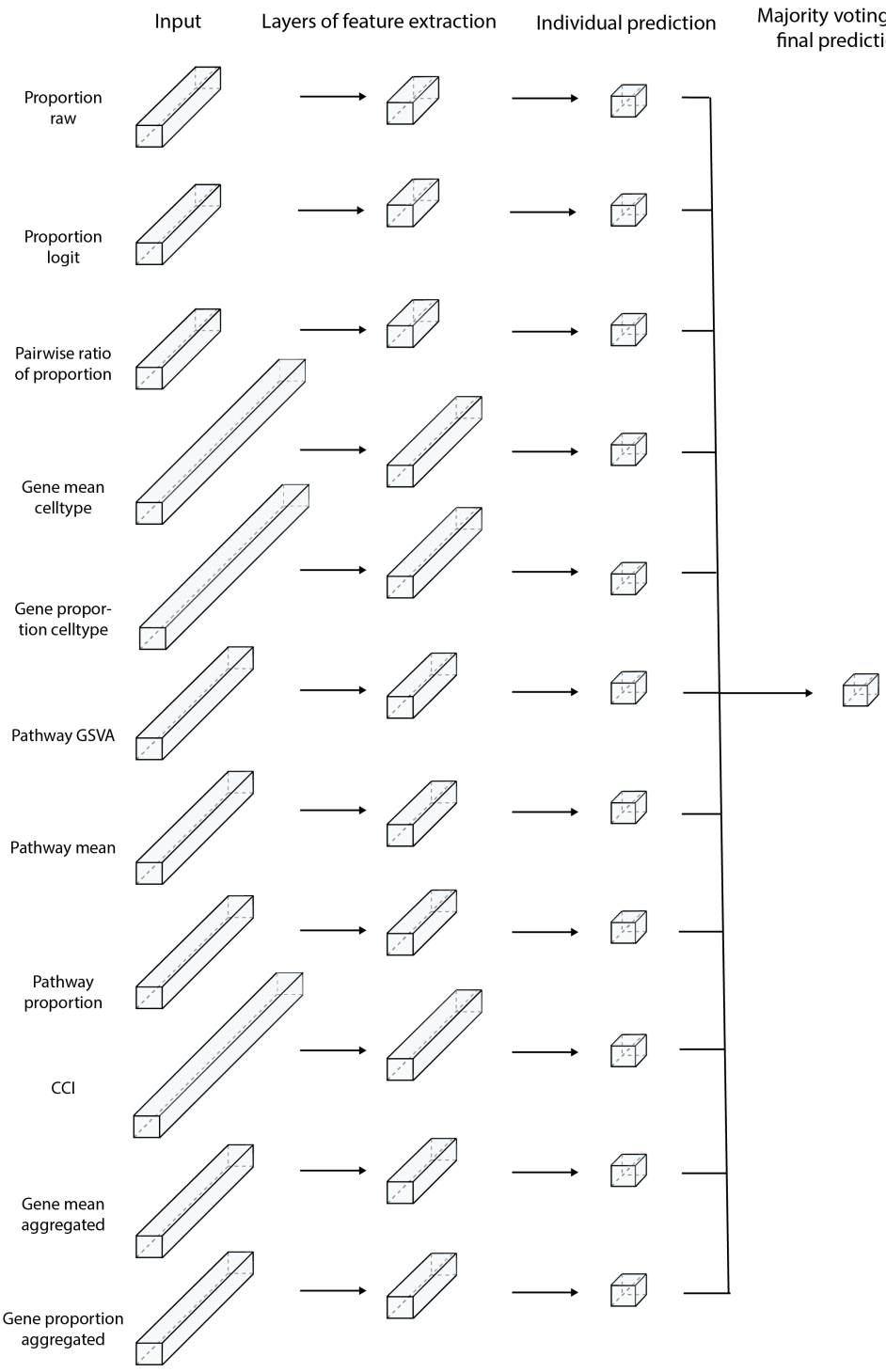


Figure C1: Schematic of the stacked ensemble approach for deep learning. The neural network is composed of a total of 11 subnetworks, each takes one feature type as the input and performs feature extraction for that feature type. The extracted features from each feature type are then concatenated and passed through another network for feature extraction and final prediction.

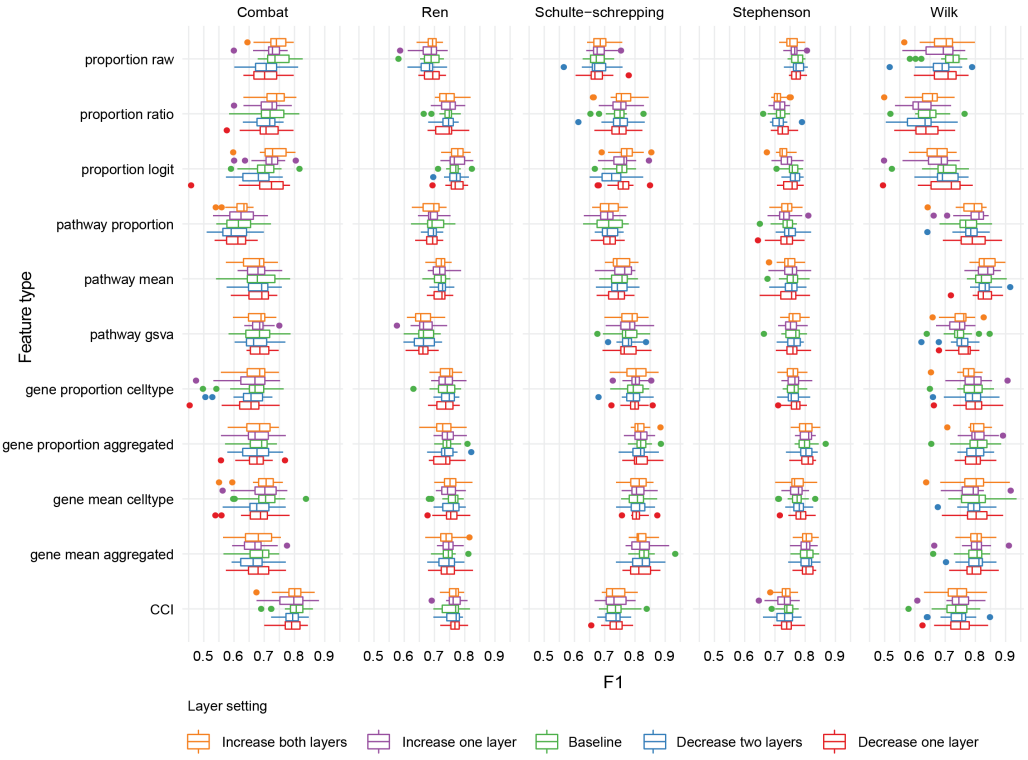


Figure C2: Performance of outcome prediction model on each dataset. Each data point in the boxplot represents one F1 score. Each box is made up of 20 F1 scores from the 20 repeated cross-validation performed on the individual dataset.

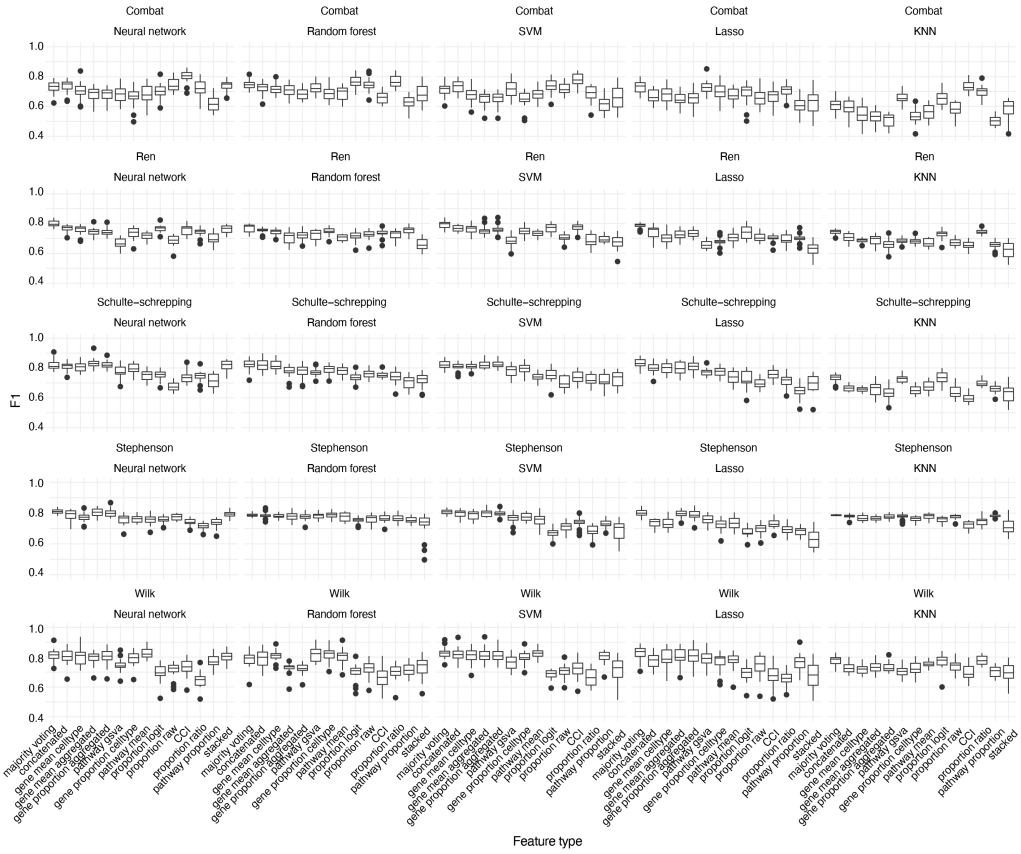


Figure C3: Performance of outcome prediction model on each dataset. Each data point in the boxplot represents one F1 score. Each box is made up of 20 F1 scores from the 20 repeated cross-validation performed on the individual dataset.

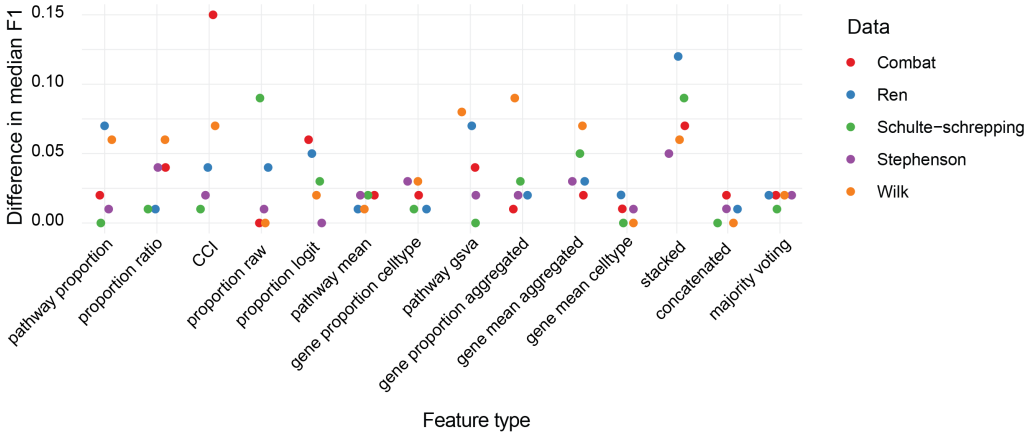


Figure C4: Difference in median F1 between neural network, random forest and SVM. For each feature type, we calculate the difference between the maximum and minimum F1 score obtained by neural network, random forest and SVM.

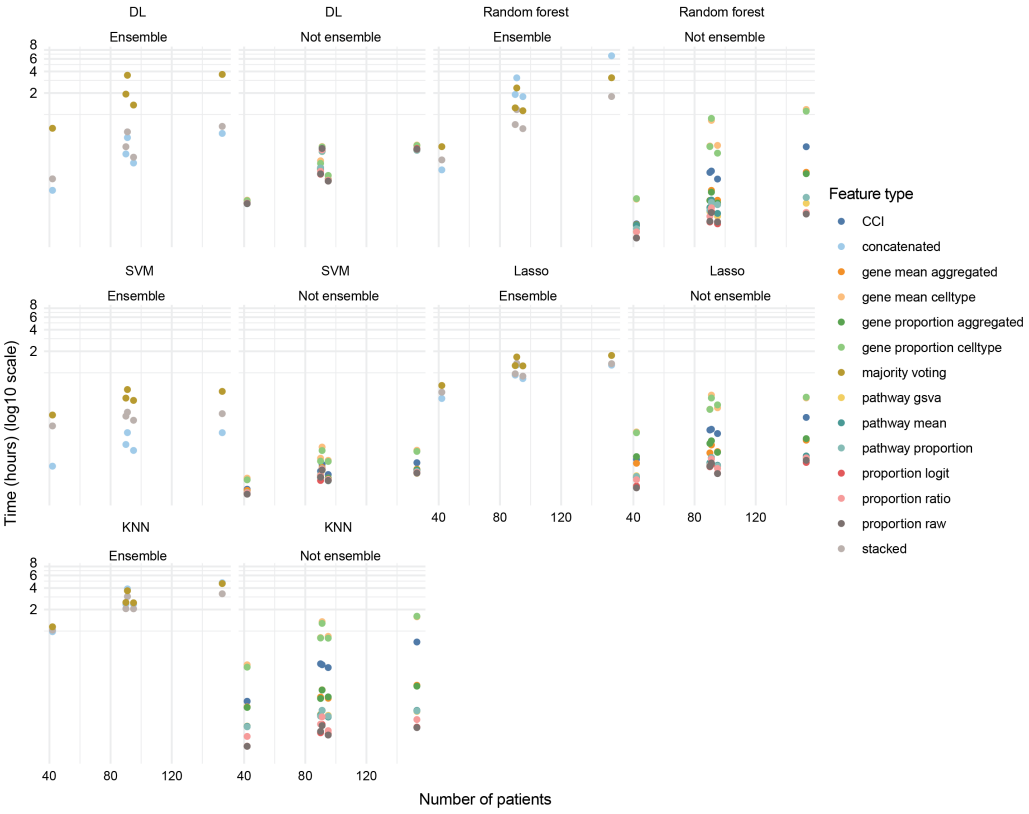


Figure C5: Run time of each feature type for the five COVID-19 datasets. Run time was recorded as the CPU time it took to train 20 repeated cross-validation models for each feature type.

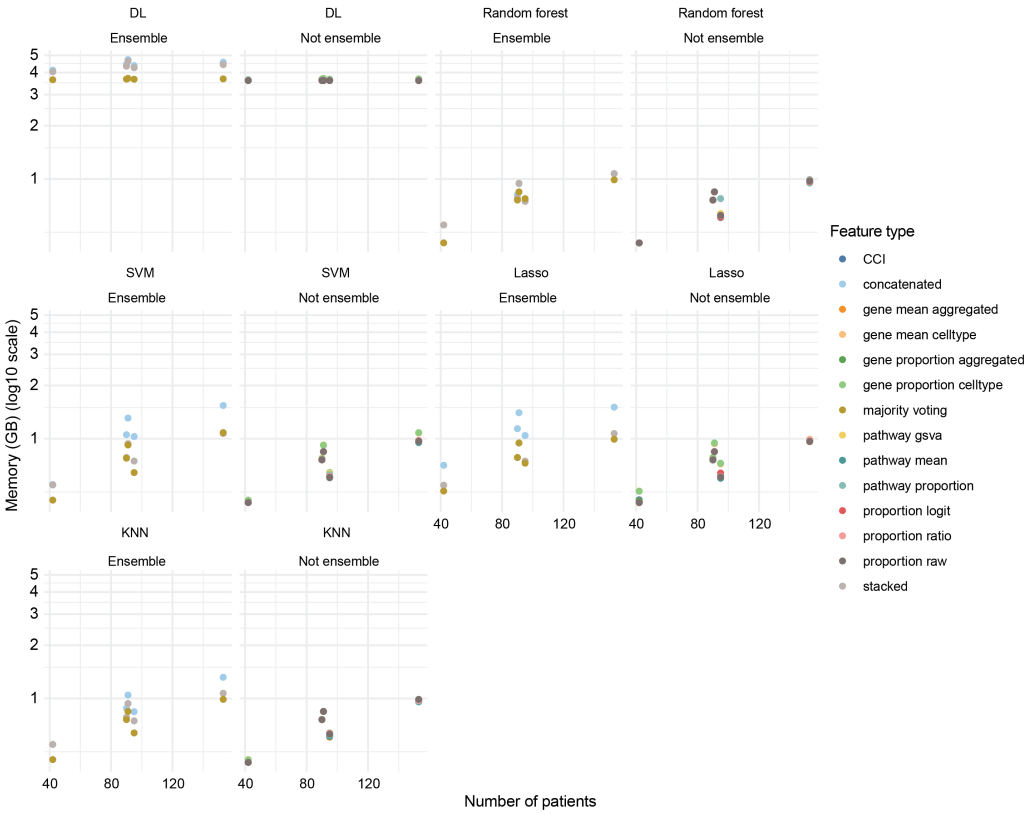


Figure C6: Peak memory usage of each feature type for the five COVID-19 datasets. For deep learning model, this was recorded as the sum of peak memory usage from both CPU and GPU. For machine learning models, this was recorded as peak CPU memory.

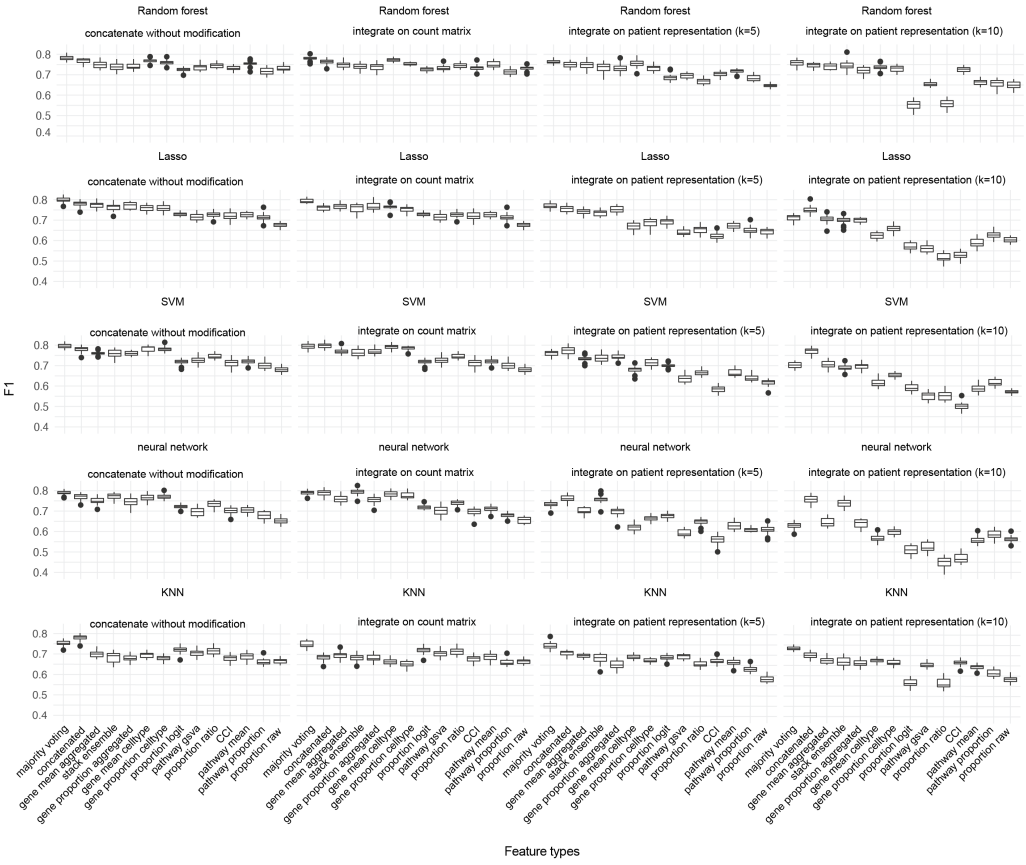


Figure C7: Performance of outcome prediction model on the combination of five datasets. Each data point in the boxplot represents one F1 score. Each box is made up of 20 F1 scores from the 20 repeated cross-validation performed on the combined dataset.

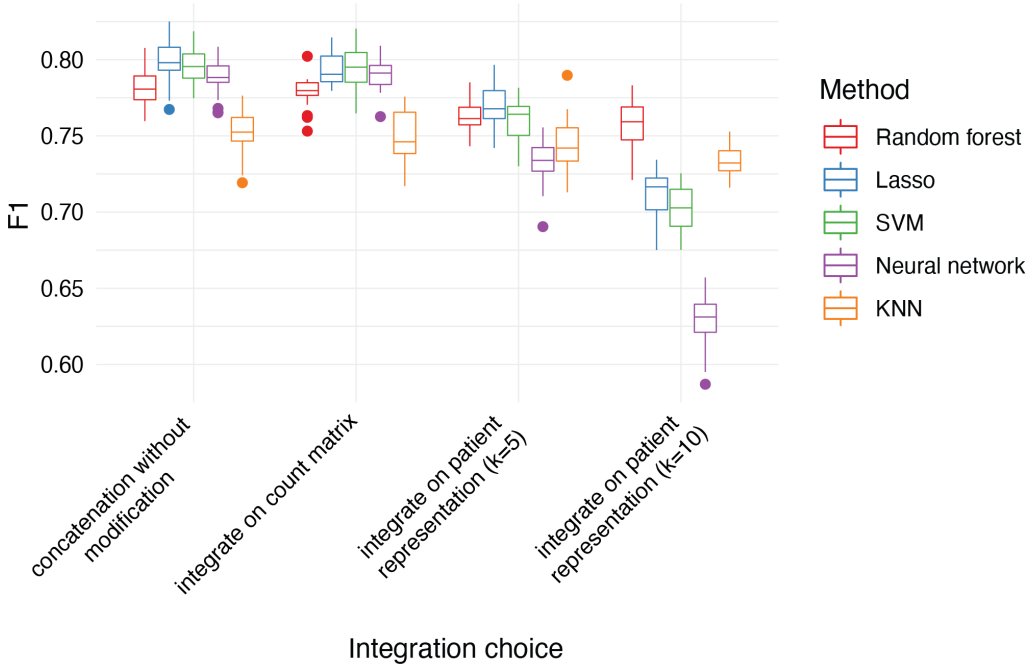


Figure C8: The F1 score for different integration choices and method choices using the ensemble feature type majority voting. Each data point in the boxplot represents one F1 score. Each box is made up of 20 F1 scores from repeated cross-validation with 20 repetitions performed on the combined dataset using the ensemble feature type majority voting.

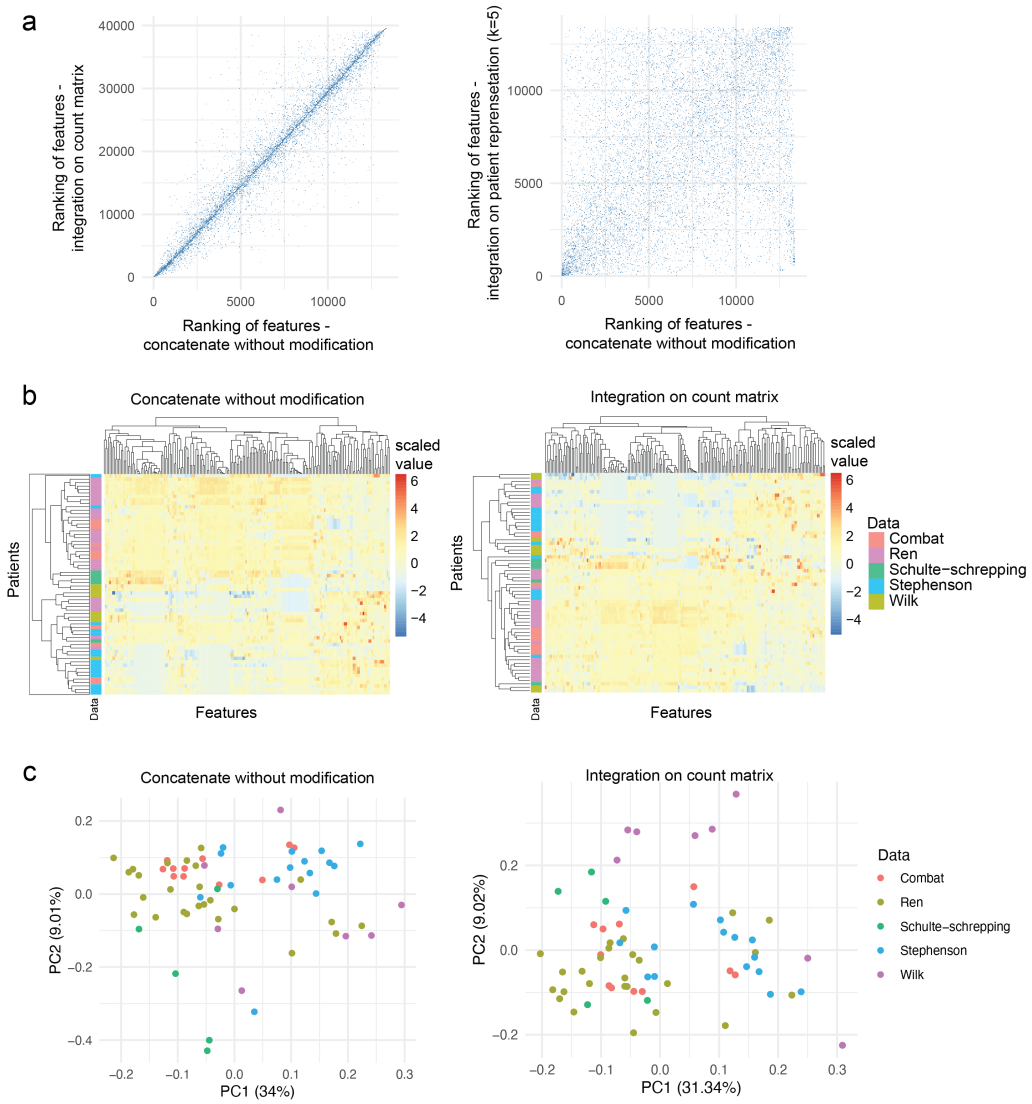


Figure C9: Examination into the features for the different integration choices. We selected patients in the 41-50 age group, and ran prediction model using the concatenated features as input and SVM as the classification method. We then obtained the rankings of the features based on feature importance score. a compares the ranks of the features from different integration choices. The heatmap in b plots the values of the top 200 features in each patient. The patient is colour coded by the dataset to reveal any potential batch effect in the features. c shows the PCA of the top 200 features, where each dot represents a patient, coloured by the dataset.

C.2 SUPPLEMENTARY TABLES

Table C1: Parameter search settings for the neural network structure. The numbers in the bracket denote the number of nodes in the first layer and second layer of the network.

Feature type	Increase both layers	Increase one layer	Baseline	Decrease both layers	Decrease one layer
proportion raw, proportion logit, proportion ratio	[50, 50]	[50, 20]	[20, 20]	[20, 10]	[10,10]
pathway mean, pathway gsva, pathway proportion, gene mean aggregated, gene proportion aggregated	[1000, 200]	[1000, 100]	[500, 100]	[500, 50]	[200, 50]
gene mean celltype, gene proportion celltype	[2000, 200]	[2000, 100]	[1000, 100]	[1000, 50]	[500, 50]

BIBLIOGRAPHY

Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M. J. T., and Mahfouz, A. (2019). A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol.*, **20**(1), 194.

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, **26**(3), 392–398.

Adams, T. S., Schupp, J. C., Poli, S., Ayaub, E. A., Neumark, N., Ahangari, F., Chu, S. G., Raby, B. A., DeIuliis, G., Januszyk, M., Duan, Q., Arnett, H. A., Siddiqui, A., Washko, G. R., Homer, R., Yan, X., Rosas, I. O., and Kaminski, N. (2020). Single-cell RNA-seq reveals ectopic and aberrant lung-resident cell populations in idiopathic pulmonary fibrosis. *Sci Adv*, **6**(28), eaba1983.

Ahern, D. J., Ai, Z., Ainsworth, M., Allan, C., Allcock, A., Angus, B., Ansari, M. A., Arancibia-Cárcamo, C. V., Aschenbrenner, D., Attar, M., Baillie, J. K., Barnes, E., Bashford-Rogers, R., Bashyal, A., Beer, S., Berridge, G., Beveridge, A., Bibi, S., Bicanic, T., Blackwell, L., Bowness, P., Brent, A., Brown, A., Broxholme, J., Buck, D., Burnham, K. L., Byrne, H., Camara, S., Candido Ferreira, I., Charles, P., Chen, W., Chen, Y.-L., Chong, A., Clutterbuck, E. A., Coles, M., Conlon, C. P., Cornall, R., Cribbs, A. P., Curion, F., Davenport, E. E., Davidson, N., Davis, S., Dendrou, C. A., Dequaire, J., Dib, L., Docker, J., Dold, C., Dong, T., Downes, D., Drakesmith, H., Dunachie, S. J., Duncan, D. A., Eijsbouts, C., Esnouf, R., Espinosa, A., Etherington, R., Fairfax, B., Fairhead, R., Fang, H., Fassih, S., Felle, S., Fernandez Mendoza, M., Ferreira, R., Fischer, R., Foord, T., Forrow, A., Frater, J., Fries, A., Gallardo Sanchez, V., Garner, L. C., Geeves, C., Georgiou, D., Godfrey, L., Golubchik, T., Gomez Vazquez, M., Green, A., Harper, H., Harrington, H. A., Heilig, R., Hester, S., Hill, J., Hinds, C., Hird, C., Ho, L.-P., Hoekzema, R., Hollis, B., Hughes, J., Hutton, P., Jackson-Wood, M. A., Jainarayanan, A., James-Bott, A., Jansen, K., Jeffery, K., Jones, E., Jostins, L., Kerr, G., Kim, D., Klenerman, P., Knight, J. C., Kumar, V., Kumar Sharma, P., Kurupati, P., Kwok, A., Lee, A., Linder, A., Lockett, T., Lonie, L., Lopopolo, M., Lukoseviciute, M., Luo, J., Marinou, S., Marsden, B., Martinez, J., Matthews, P. C., Mazurczyk, M., McGowan, S., McKechnie, S., Mead, A., Mentzer, A. J., Mi, Y., Monaco, C., Montadon, R., Napolitani, G., Nassiri, I., Novak, A., O'Brien, D. P., O'Connor, D., O'Donnell, D., Ogg, G., Overend, L., Park, I., Pavord, I., Peng, Y., Penkava, F., Pereira Pinho, M., Perez, E., Pollard, A. J., Powrie, F., Psaila, B., Quan, T. P., Repapi, E., Revale, S., Silva-Reyes, L., Richard, J.-B., Rich-Griffin, C., Ritter, T., Rollier, C. S., Rowland, M., Ruehle, F., Salio, M., Sansom, S. N., Sanches Peres, R., Santos Delgado, A., Sauka-Spengler, T., Schwessinger, R., Scozzafava, G., Screaton, G., Seigal, A., Semple, M. G., Sergeant, M., Simoglou Karali, C., Sims, D., Skelly, D., Slawinski, H., Sobrinodiaz, A., Sousos, N., Stafford, L., Stockdale, L., Strickland, M., Sumray, O., Sun, B., Taylor, C., Taylor, S., Taylor, A., Thongjuea, S., Thraves, H., Todd, J. A., Tomic, A., Tong, O., Trebes, A., Trzupek, D., Tucci, F. A., Turtle, L., Udalova, I., Uhlig, H., van Grinsven, E., Vendrell, I., Verheul, M., Voda, A., Wang, G., Wang, L., Wang, D., Watkinson, P., Watson, R., Weinberger, M., Whalley, J., Witty, L., Wray, K., Xue, L., Yeung, H. Y., Yin, Z., Young, R. K., Youngs, J., Zhang, P., and Zurke, Y.-X. (2022). A blood atlas of covid-19 defines hallmarks of disease severity and specificity. *Cell*, **185**(5), 916–938.e58.

Aibar, S., González-Blas, C. B., Moerman, T., Huynh-Thu, V. A., Imrichova, H., Hulselmans, G., Rambow, F., Marine, J.-C., Geurts, P., Aerts, J., van den Oord, J., Atak, Z. K., Wouters, J., and Aerts, S. (2017). SCENIC: single-cell regulatory network inference and clustering. *Nat. Methods*, **14**(11), 1083–1086.

Altman, N. and Krzywinski, M. (2017). Points of significance: ensemble methods: bagging and random forests. *Nat. Methods*, **14**(10), 933–935.

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol.*, **11**(10), R106.

Angermueller, C., Lee, H. J., Reik, W., and Stegle, O. (2017). Deepcpg: accurate prediction of single-cell dna methylation states using deep learning. *Genome Biol.*, **18**(1), 67.

Antelmi, L., Ayache, N., Robert, P., and Lorenzi, M. (ICML, 2019). Sparse multi-channel variational autoencoder for the joint analysis of heterogeneous data. In *Proc. 36th International Conference on Machine Learning*, pages 302–311.

Arefeen, A., Xiao, X., and Jiang, T. (2019). Deeppasta: deep neural network based polyadenylation site analysis. *Bioinformatics*, **35**(22), 4577–4585.

Armingol, E., Officer, A., Harismendy, O., and Lewis, N. E. (2020). Deciphering cell–cell interactions and communication from gene expression. *Nat. Rev. Genet.*, **22**(2), 71–88.

Armingol, E., Officer, A., Harismendy, O., and Lewis, N. E. (2021). Deciphering cell–cell interactions and communication from gene expression. *Nature Reviews Genetics*, **22**(2), 71–88.

Armstrong, J. S. (1978). *Long-range forecasting*. Wiley.

Ashley, E. A. (2016). Towards precision medicine. *Nat. Rev. Genet.*, **17**(9), 507–522.

Assefa, A. T., Vandesompele, J., and Thas, O. (2020). SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. *Bioinformatics*, **36**(10), 3276–3278.

Bachman, P., Alsharif, O., and Precup, D. (NIPS, 2014). Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems 27*, pages 3365–3373.

Baek, S. and Lee, I. (2020). Single-cell ATAC sequencing analysis: From data preprocessing to hypothesis generation. *Comput. Struct. Biotechnol. J.*, **18**, 1429–1439.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Baldi, P. (ICML, 2012). Autoencoders, unsupervised learning, and deep architectures. In *Proc. of ICML Workshop on Unsupervised and Transfer learning*, pages 37–49.

Baldi, P. and Sadowski, P. J. (NIPS, 2013). Understanding dropout. In *Advances in Neural Information Processing Systems 26*, pages 2814–2822.

Bao, S., Li, K., Yan, C., Zhang, Z., Qu, J., and Zhou, M. (2022). Deep learning-based advances and applications for single-cell RNA-sequencing data analysis. *Brief. Bioinform.*, **23**(1).

Bartoszewicz, J. M., Seidel, A., Rentzsch, R., and Renard, B. Y. (2020). Deepac: predicting pathogenic potential of novel dna with reverse-complement neural networks. *Bioinformatics*, **36**(1), 81–89.

Baruzzo, G., Patuzzi, I., and Di Camillo, B. (2020). SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics*, **36**(5), 1468–1475.

Bengio, Y. (2009). Learning deep architectures for ai. *Found. Trends Mach. Learn.*, **2**(1), 1–127.

Bergen, V., Soldatov, R. A., Kharchenko, P. V., and Theis, F. J. (2021). RNA velocity—current challenges and future perspectives. *Mol. Syst. Biol.*, **17**(8), e10282.

Bost, P., Giladi, A., Liu, Y., Bendjelal, Y., Xu, G., David, E., Blecher-Gonen, R., Cohen, M., Medaglia, C., Li, H., *et al.* (2020). Host-viral infection maps reveal signatures of severe covid-19 patients. *Cell*, **181**(7), 1475–1488.

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, **24**(2), 123–140.

Buettner, F., Pratanwanich, N., McCarthy, D. J., Marioni, J. C., and Stegle, O. (2017). f-scLVM: scalable and versatile factor analysis for single-cell RNA-seq. *Genome Biol.*, **18**(1), 212.

Bzdok, D., Nichols, T. E., and Smith, S. M. (2019). Towards algorithmic analytics for large-scale datasets. *Nat. Mach. Intell.*, **1**(7), 296–306.

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., and Collins, J. J. (2018). Next-generation machine learning for biological networks. *Cell*, **173**(7), 1581–1592.

Cannoodt, R., Saelens, W., Deconinck, L., and Saeys, Y. (2021). Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat. Commun.*, **12**(1), 3942.

Cao, Y., Lin, Y., Ormerod, J. T., Yang, P., Yang, J. Y., and Lo, K. K. (2019). scdc: single cell differential composition analysis. *BMC Bioinfo.*, **20**(19), 1–12.

Cao, Y., Geddes, T. A., Yang, J. Y. H., and Yang, P. (2020). Ensemble deep learning in bioinformatics. *Nat. Mach. Intell.*, **2**(9), 500–508.

Cao, Y., Yang, P., and Yang, J. Y. H. (2021). A benchmark study of simulation methods for single-cell rna sequencing data. *Nat. Commun.*, **12**(1), 1–12.

Cao, Y., Yisimayi, A., Jian, F., Song, W., Xiao, T., Wang, L., Du, S., Wang, J., Li, Q., Chen, X., *et al.* (2022a). Ba. 2.12. 1, ba. 4 and ba. 5 escape antibodies elicited by omicron infection. *Nature*, pages 1–10.

Cao, Y., Lin, Y., Patrick, E., Yang, P., and Yang, J. Y. H. (2022b). scFeatures: multi-view representations of single-cell and spatial data for disease outcome prediction. *Bioinformatics*. btac590.

Cao, Z., Pan, X., Yang, Y., Huang, Y., and Shen, H.-B. (2018). The lnclocator: a subcellular localization predictor for long non-coding rnas based on a stacked ensemble classifier. *Bioinformatics*, **34**(13), 2185–2194.

Chen, W., Li, Y., Easton, J., Finkelstein, D., Wu, G., and Chen, X. (2018). Umi-count modeling and differential expression analysis for single-cell rna sequencing. *Genome Biol.*, **19**(1), 1–17.

Cheng, C., Easton, J., Rosencrance, C., Li, Y., Ju, B., Williams, J., Mulder, H. L., Pang, Y., Chen, W., and Chen, X. (2019). Latent cellular analysis robustly

reveals subtle diversity in large-scale single-cell RNA-seq data. *Nucleic Acids Res.*, **47**(22), e143.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (EMNLP, 2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proc. 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Codella, N. C. F., Nguyen, Q.-B., Pankanti, S., Gutman, D. A., Helba, B., Halpern, A. C., and Smith, J. R. (2017). Deep learning ensembles for melanoma recognition in dermoscopy images. *IBM J. Res. Dev.*, **61**(4–5), 5:1–5:15.

Cox, J. and Mann, M. (2011). Quantitative, high-resolution proteomics for data-driven systems biology. *Annu. Rev. Biochem.*, **80**, 273–299.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., *et al.* (NIPS, 2012). Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25*, pages 1223–1231.

Deaton, A. M., Webb, S., Kerr, A. R., Illingworth, R. S., Guy, J., Andrews, R., and Bird, A. (2011). Cell type–specific dna methylation at intragenic cpg islands in the immune system. *Genome research*, **21**(7), 1074–1086.

Demichev, V., Messner, C. B., Vernardis, S. I., Lilley, K. S., and Ralser, M. (2020). Dia-nn: neural networks and interference correction enable deep proteome coverage in high throughput. *Nat. Methods*, **17**(1), 41–44.

Dibaeinia, P. and Sinha, S. (2020). SERGIO: A Single-Cell expression simulator guided by gene regulatory networks. *Cell Syst.*, **11**(3), 252–271.e11.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer.

Ding, J., Adiconis, X., Simmons, S. K., Kowalczyk, M. S., Hession, C. C., Marjanovic, N. D., Hughes, T. K., Wadsworth, M. H., Burks, T., Nguyen, L. T., *et al.* (2020). Systematic comparison of single-cell and single-nucleus rna-sequencing methods. *Nature biotechnology*, **38**(6), 737–746.

Dong, X., Yu, Z., Cao, W., Shi, Y., and Ma, Q. (2020). A survey on ensemble learning. *Front. Comput. Sci.*, **14**(2), 241–258.

Duong, T., Goud, B., and Schauer, K. (2012). Closed-form density-based framework for automatic detection of cellular morphology changes. *Proc. Natl. Acad. Sci. U. S. A.*, **109**(22), 8382–8387.

Dvornik, N., Schmid, C., and Mairal, J. (ICCV, 2019). Diversity with cooperation: ensemble methods for few-shot classification. In *Proc. IEEE International Conference on Computer Vision*, pages 3723–3731.

Eraslan, G., Avsec, Ž., Gagneur, J., and Theis, F. J. (2019). Deep learning: new computational modelling techniques for genomics. *Nat. Rev. Genet.*, **20**(7), 389–403.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res*, **11**(Feb), 625–660.

Feuk, L., Carson, A. R., and Scherer, S. W. (2006). Structural variation in the human genome. *Nat. Rev. Genet.*, **7**(2), 85–97.

Gagnon-Bartsch, J. A. and Speed, T. P. (2012). Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, **13**(3), 539–552.

Gala, R., Gouwens, N., Yao, Z., Budzillo, A., Penn, O., Tasic, B., Murphy, G., Zeng, H., and Sümbül, U. (NIPS, 2019). A coupled autoencoder approach for multi-modal analysis of cell types. In *Advances in Neural Information Processing Systems 32*, pages 9263–9272.

Garg, M., Li, X., Moreno, P., Papatheodorou, I., Shu, Y., Brazma, A., and Miao, Z. (2021). Meta-analysis of covid-19 single-cell studies confirms eight key immune responses. *Sci. Rep.*, **11**(1), 1–10.

Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A., and Yosef, N. (2021). Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nat. Methods*, **18**(3), 272–282.

Geddes, T. A., Kim, T., Nan, L., Burchfield, J. G., Yang, J. Y. H., Tao, D., and Yang, P. (2019). Autoencoder-based cluster ensembles for single-cell rna-seq data analysis. *BMC Bioinf.*, **20**(19), 660.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Comput.*, **4**(1), 1–58.

Goldman, S. L., MacKay, M., Afshinnekoo, E., Melnick, A. M., Wu, S., and Mason, C. E. (2019). The impact of heterogeneity on Single-Cell sequencing. *Front. Genet.*, **10**, 8.

González-Silva, L., Quevedo, L., and Varela, I. (2020). Tumor functional heterogeneity unraveled by scrna-seq technologies. *Trends in cancer*, **6**(1), 13–19.

Goodwin, S., McPherson, J. D., and McCombie, W. R. (2016). Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**(6), 333–351.

Granitto, P. M., Verdes, P. F., and Ceccatto, H. A. (2005). Neural network ensembles: evaluation of aggregation algorithms. *Artif. Intell.*, **163**(2), 139–162.

Grewal, J. K., Tessier-Cloutier, B., Jones, M., Gakkhar, S., Ma, Y., Moore, R., Mungall, A. J., Zhao, Y., Taylor, M. D., Gelmon, K., Lim, H., Renouf, D., Laskin, J., Marra, M., Yip, S., and Jones, S. J. M. (2019). Application of a neural network whole transcriptome–based pan-cancer method for diagnosis of primary and metastatic cancers. *JAMA Netw. Open*, **2**(4), e192597–e192597.

Hafemeister, C. and Satija, R. (2019). Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. *Genome Biol.*, **20**(1), 296.

Han, B., Sim, J., and Adam, H. (CVPR, 2017). Branchout: regularization for online ensemble tracking with convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3356–3365.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE Trans. Pattern Anal. Mach.*, **12**(10), 993–1001.

Hänzelmann, S., Castelo, R., and Guinney, J. (2013). GSVA: gene set variation analysis for microarray and RNA-seq data. *BMC Bioinf.*, **14**, 7.

Hara, K., Saitoh, D., and Shouno, H. (ICANN, 2016). Analysis of dropout learning regarded as ensemble learning. In *Proc. 25th International Conference on Artificial Neural Networks*, pages 72–79.

He, F., Wang, R., Gao, Y., Wang, D., Yu, Y., Xu, D., and Zhao, X. (IEEE BIBM, 2019). Protein ubiquitylation and sumoylation site prediction based on ensemble and transfer learning. In *Proc. 2019 IEEE International Conference on Bioinformatics and Biomedicine*, pages 117–123.

He, K., Zhang, X., Ren, S., and Sun, J. (CVPR, 2016). Deep residual learning for image recognition. In *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

He, Y., Yuan, H., Wu, C., and Xie, Z. (2020). DISC: a highly scalable and accurate inference of gene expression and structure for single-cell transcriptomes using semi-supervised deep learning. *Genome Biol.*, **21**(1), 170.

Hinton, G., Vinyals, O., Dean, J., *et al.* (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, **2**(7).

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, **9**(8), 1735–1780.

Hu, H., Xiao, A., Zhang, S., Li, Y., Shi, X., Jiang, T., Zhang, L., Zhang, L., and Zeng, J. (2018). Deephint: understanding HIV-1 integration via deep learning with attention. *Bioinformatics*, **35**(10), 1660–1667.

Hu, S., Zhang, C., Chen, P., Gu, P., Zhang, J., and Wang, B. (2019a). Predicting drug-target interactions from drug structure and protein sequence using novel convolutional neural networks. *BMC Bioinf.*, **20**(25), 1–12.

Hu, Y., Wang, Z., Hu, H., Wan, F., Chen, L., Xiong, Y., Wang, X., Zhao, D., Huang, W., and Zeng, J. (2019b). Acme: pan-specific peptide–mhc class i binding prediction through attention-based deep neural networks. *Bioinformatics*, **35**(23), 4946–4954.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *14th European Conference on Computer Vision*, pages 646–661. Springer.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. (2017). Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*.

Huang, Z., Zhou, J. T., Peng, X., Zhang, C., Zhu, H., and Lv, J. (IJCAI, 2019). Multi-view spectral clustering network. In *Proc. 28th International Joint Conference on Artificial Intelligence*, pages 2563–2569.

Jin, K., Bardes, E. E., Mitelpunkt, A., Wang, J. Y., Bhatnagar, S., Sengupta, S., Krummel, D. P., Rothenberg, M. E., and Aronow, B. J. (2021). An interactive single cell web portal identifies gene and cell networks in COVID-19 host responses. *iScience*, **24**(10), 103115.

Ju, C., Bibaut, A., and van der Laan, M. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. *J. Appl. Stat.*, **45**(15), 2800–2818.

Junttila, S., Smolander, J., and Elo, L. L. (2022). Benchmarking methods for detecting differential states between conditions from multi-subject single-cell RNA-seq data. *Brief. Bioinform.*

Karim, M. R., Rahman, A., Jares, J. B., Decker, S., and Beyan, O. (2019). A snapshot neural ensemble method for cancer-type prediction based on copy number variations. *Neural Comput. Appl.*

Karimi, M., Wu, D., Wang, Z., and Shen, Y. (2019). Deepaffinity: interpretable deep learning of compound–protein affinity through unified recurrent and convolutional neural networks. *Bioinformatics*, **35**(18), 3329–3338.

Keren, L., Bosse, M., Thompson, S., Risom, T., Vijayaragavan, K., McCaffrey, E., Marquez, D., Angoshtari, R., Greenwald, N. F., Fienberg, H., Wang, J., Kambham, N., Kirkwood, D., Nolan, G., Montine, T. J., Galli, S. J., West, R., Bendall, S. C., and Angelo, M. (2019). MIBI-TOF: A multiplexed imaging platform relates cellular phenotypes and tissue structure. *Sci Adv*, **5**(10), eaax5851.

Keskar, N. S., Nocedal, J., Tang, P. T. P., Mudigere, D., and Smelyanskiy, M. (ICLR, 2017). On large-batch training for deep learning: generalization gap and sharp minima. In *Proc. 5th International Conference on Learning Representations*.

Kim, H. J., Osteil, P., Humphrey, S. J., Cinghu, S., Oldfield, A. J., Patrick, E., Wilkie, E. E., Peng, G., Suo, S., Jothi, R., *et al.* (2020). Transcriptional network dynamics during the progression of pluripotency revealed by integrative statistical learning. *Nucleic Acids Res.*, **48**(4), 1828–1842.

Kim, H. J., Wang, K., Chen, C., Lin, Y., Tam, P. P., Lin, D. M., Yang, J. Y., and Yang, P. (2021). Uncovering cell identity through differential stability with cepo. *Nature Computational Science*, **1**(12), 784–790.

Kinker, G. S., Greenwald, A. C., Tal, R., Orlova, Z., Cuoco, M. S., McFarland, J. M., Warren, A., Rodman, C., Roth, J. A., Bender, S. A., Kumar, B., Rocco, J. W., Fernandes, P. A., Mader, C. C., Keren-Shaul, H., Plotnikov, A., Barr, H., Tsherniak, A., Rozenblatt-Rosen, O., Krizhanovsky, V., Puram, S. V., Regev, A., and Tirosh, I. (2019). Pan-cancer single cell RNA-seq uncovers recurring programs of cellular heterogeneity.

Kitano, H. (2002). Computational systems biology. *Nature*, **420**(6912), 206–210.

Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., and Teichmann, S. A. (2015a). The technology and biology of single-cell rna sequencing. *Mol. Cell*, **58**(4), 610–620.

Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., and Teichmann, S. A. (2015b). The technology and biology of single-cell RNA sequencing. *Mol. Cell*, **58**(4), 610–620.

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

Korthauer, K. D., Chu, L.-F., Newton, M. A., Li, Y., Thomson, J., Stewart, R., and Kendziorski, C. (2016). A statistical approach for identifying differential distributions in single-cell RNA-seq experiments. *Genome Biol.*, **17**(1), 222.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (NIPS, 2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software, Articles*, **28**(5), 1–26.

Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D. J., Hicks, S. C., Robinson, M. D., Vallejos, C. A., Campbell, K. R., Beerenwinkel, N., Mahfouz, A., Pinello, L., Skums, P., Stamatakis, A., Attolini, C. S.-O., Aparicio, S., Baaijens, J., Balvert, M., Barbanson, B. d., Cappuccio, A., Corleone, G., Dutilh, B. E.,

Florescu, M., Guryev, V., Holmer, R., Jahn, K., Lobo, T. J., Keizer, E. M., Khatri, I., Kielbasa, S. M., Korbel, J. O., Kozlov, A. M., Kuo, T.-H., Lelieveldt, B. P. F., Mandoiu, I. I., Marioni, J. C., Marschall, T., Mölder, F., Niknejad, A., Raczkowski, L., Reinders, M., Ridder, J. d., Saliba, A.-E., Somarakis, A., Stegle, O., Theis, F. J., Yang, H., Zelikovsky, A., McHardy, A. C., Raphael, B. J., Shah, S. P., and Schönhuth, A. (2020). Eleven grand challenges in single-cell data science. *Genome Biol.*, **21**(1), 31.

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armananzas, R., Santafé, G., Pérez, A., *et al.* (2006). Machine learning in bioinformatics. *Briefings Bioinf.*, **7**(1), 86–112.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436–444.

Lee, D., Redfern, O., and Orengo, C. (2007). Predicting protein function from sequence and structure. *Nat. Rev. Mol. Cell Biol.*, **8**(12), 995–1005.

Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). Why m heads are better than one: Training a diverse ensemble of deep networks. *arXiv preprint arXiv:1511.06314*.

Lee, S., Prakash, S. P. S., Cogswell, M., Ranjan, V., Crandall, D., and Batra, D. (NIPS, 2016). Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems 29*, pages 2119–2127.

Li, W. V. and Li, J. J. (2019). A statistical simulator scdesign for rational scRNA-seq experimental design. *Bioinformatics*, **35**(14), i41–i50.

Li, Y., Wu, F.-X., and Ngom, A. (2018). A review on machine learning principles for multi-view biological data integration. *Brief. Bioinform.*, **19**(2), 325–340.

Li, Z. and Yu, Y. (AAAI, 2016). Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. In *Proc. 25th International Joint Conference on Artificial Intelligence*, pages 2560–2567.

Liang, M., Li, Z., Chen, T., and Zeng, J. (2014). Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **12**(4), 928–937.

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. (2015). The molecular signatures database (MSigDB) hallmark gene set collection. *Cell Syst*, **1**(6), 417–425.

Lim, B., Lin, Y., and Navin, N. (2020). Advancing cancer research and medicine with single-cell genomics. *Cancer cell*, **37**(4), 456–470.

Lin, W. N., Tay, M. Z., Lu, R., Liu, Y., Chen, C.-H., and Cheow, L. F. (2020a). The role of Single-Cell technology in the study and control of infectious diseases. *Cells*, **9**(6).

Lin, Y., Ghazanfar, S., Wang, K. Y., Gagnon-Bartsch, J. A., Lo, K. K., Su, X., Han, Z.-G., Ormerod, J. T., Speed, T. P., Yang, P., *et al.* (2019). scmerge leverages factor analysis, stable expression, and pseudoreplication to merge multiple single-cell rna-seq datasets. *Proceedings of the National Academy of Sciences*, **116**(20), 9775–9784.

Lin, Y., Cao, Y., Kim, H. J., Salim, A., Speed, T. P., Lin, D. M., Yang, P., and Yang, J. Y. H. (2020b). scclassify: sample size estimation and multiscale classification of cells using single and multiple reference. *Mol. Syst. Biol.*, **16**(6), e9389.

Liu, Y. and Yao, X. (1999). Ensemble learning via negative correlation. *Neural Networks*, **12**(10), 1399–1404.

Longo, S. K., Guo, M. G., Ji, A. L., and Khavari, P. A. (2021). Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nat. Rev. Genet.*, **22**(10), 627–644.

Lu, Z., Bai, Y., Chen, Y., Su, C., Lu, S., Zhan, T., Hong, X., and Wang, S. (2020). The classification of gliomas based on a pyramid dilated convolution resnet model. *Pattern Recognit. Lett.*

Luecken, M. D. and Theis, F. J. (2019). Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.*, **15**(6).

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

Lun, A. T. and Marioni, J. C. (2017). Overcoming confounding plate effects in differential expression analyses of single-cell rna-seq data. *Biostatistics*, **18**(3), 451–464.

Maleki, F., Ovens, K., Hogan, D. J., and Kusalik, A. J. (2020). Gene set analysis: Challenges, opportunities, and future research. *Front. Genet.*, **11**, 654.

Maniatis, S., Äijö, T., Vickovic, S., Braine, C., Kang, K., Mollbrink, A., Fagegaltier, D., Andrusivová, Ž., Saarenpää, S., Saiz-Castro, G., Cuevas, M., Watters, A., Lundeberg, J., Bonneau, R., and Phatnani, H. (2019). Spatiotemporal dynamics of molecular pathology in amyotrophic lateral sclerosis. *Science*, **364**(6435), 89–93.

Marouf, M., Machart, P., Bansal, V., Kilian, C., Magruder, D. S., Krebs, C. F., and Bonn, S. (2020). Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nat. Commun.*, **11**(1), 166.

Marx, V. (2021). Method of the year: spatially resolved transcriptomics. *Nat. Methods*, **18**(1), 9–14.

Miao, Z., Deng, K., Wang, X., and Zhang, X. (2018). Desingle for detecting three types of differential expression in single-cell rna-seq data. *Bioinformatics*, **34**(18), 3223–3224.

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings Bioinf.*, **18**(5), 851–869.

Mou, T., Deng, W., Gu, F., Pawitan, Y., and Vu, T. N. (2020). Reproducibility of methods to detect differentially expressed genes from single-cell rna sequencing. *Frontiers in genetics*, **10**, 1331.

Nature Methods (2014). Method of the year 2013. *Nat. Methods*, **11**(1), 1–1.

Nguyen, N. D. and Wang, D. (2020). Multiview learning for understanding functional multiomics. *PLoS Comput. Biol.*, **16**(4), e1007677.

Ozsolak, F. and Milos, P. M. (2011). Rna sequencing: advances, challenges and opportunities. *Nat. Rev. Genet.*, **12**(2), 87–98.

Papadopoulos, N., Gonzalo, P. R., and Söding, J. (2019). PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes. *Bioinformatics*, **35**(18), 3517–3519.

Parisotto, E., Ba, J., and Salakhutdinov, R. (ICLR, 2016). Actor-mimic: deep multitask and transfer reinforcement learning. In *Proc. International Conference on Learning Representations*.

Penaranda, C. and Hung, D. T. (2019). Single-cell rna sequencing to understand host–pathogen interactions. *ACS infectious diseases*, **5**(3), 336–344.

Portela, A. and Esteller, M. (2010). Epigenetic modifications and human disease. *Nat. Biotechnol.*, **28**(10), 1057.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., and Murali, T. M. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods*, **17**(2), 147–154.

Pusztai, L., Hatzis, C., and Andre, F. (2013). Reproducibility of research and preclinical validation: problems and solutions. *Nature Reviews Clin. Oncol.*, **10**(12), 720.

Qi, C., Wang, C., Zhao, L., Zhu, Z., Wang, P., Zhang, S., Cheng, L., and Zhang, X. (2022). SCovid: single-cell atlases for exposing molecular characteristics of COVID-19 across 10 human tissues. *Nucleic Acids Res.*, **50**(D1), D867–D874.

Ramazzotti, D., Lal, A., Wang, B., Batzoglou, S., and Sidow, A. (2018). Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival. *Nat. Commun.*, **9**(1), 1–14.

Rasti, R., Teshnehlab, M., and Phung, S. L. (2017). Breast cancer diagnosis in dce-mri using mixture ensemble of convolutional neural networks. *Pattern Recognit.*, **72**, 381–390.

Ren, X., Wen, W., Fan, X., Hou, W., Su, B., Cai, P., Li, J., Liu, Y., Tang, F., Zhang, F., Yang, Y., He, J., Ma, W., He, J., Wang, P., Cao, Q., Chen, F., Chen, Y., Cheng, X., Deng, G., Deng, X., Ding, W., Feng, Y., Gan, R., Guo, C., Guo, W., He, S., Jiang, C., Liang, J., Li, Y.-M., Lin, J., Ling, Y., Liu, H., Liu, J., Liu, N., Liu, S.-Q., Luo, M., Ma, Q., Song, Q., Sun, W., Wang, G., Wang, F., Wang, Y., Wen, X., Wu, Q., Xu, G., Xie, X., Xiong, X., Xing, X., Xu, H., Yin, C., Yu, D., Yu, K., Yuan, J., Zhang, B., Zhang, P., Zhang, T., Zhao, J., Zhao, P., Zhou, J., Zhou, W., Zhong, S., Zhong, X., Zhang, S., Zhu, L., Zhu, P., Zou, B., Zou, J., Zuo, Z., Bai, F., Huang, X., Zhou, P., Jiang, Q., Huang, Z., Bei, J.-X., Wei, L., Bian, X.-W., Liu, X., Cheng, T., Li, X., Zhao, P., Wang, F.-S., Wang, H., Su, B., Zhang, Z., Qu, K., Wang, X., Chen, J., Jin, R., and Zhang, Z. (2021). COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas. *Cell*, **184**(23), 5838.

Risso, D., Ngai, J., Speed, T. P., and Dudoit, S. (2014). Normalization of rna-seq data using factor analysis of control genes or samples. *Nature biotechnology*, **32**(9), 896–902.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.*, **9**(1), 284.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015a). limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, **43**(7), e47–e47.

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., and Smyth, G. K. (2015b). limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.*, **43**(7), e47.

Robinson, M. D. and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, **11**(3), R25.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323**(6088), 533–536.

Sade-Feldman, M., Yizhak, K., Bjorgaard, S. L., Ray, J. P., de Boer, C. G., Jenkins, R. W., Lieb, D. J., Chen, J. H., Frederick, D. T., Barzily-Rokni, M., Freeman, S. S., Reuben, A., Hoover, P. J., Villani, A.-C., Ivanova, E., Portell, A., Lizotte, P. H., Aref, A. R., Eliane, J.-P., Hammond, M. R., Vitzthum, H., Blackmon, S. M., Li, B., Gopalakrishnan, V., Reddy, S. M., Cooper, Z. A., Paweletz, C. P., Barbie, D. A., Stemmer-Rachamimov, A., Flaherty, K. T., Wargo, J. A., Boland, G. M., Sullivan, R. J., Getz, G., and Hacohen, N. (2019). Defining T cell states associated with response to checkpoint immunotherapy in melanoma. *Cell*, **176**(1-2), 404.

Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2019). A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.*, **37**(5), 547–554.

Saiselet, M., Rodrigues-Vitória, J., Tourneur, A., Craciun, L., Spinette, A., Larsimont, D., Andry, G., Lundeberg, J., Maenhaut, C., and Detours, V. (2020). Transcriptional output, cell-type densities, and normalization in spatial transcriptomics. *Journal of molecular cell biology*, **12**(11), 906–908.

Sathyamurthy, A., Johnson, K. R., Matson, K. J. E., Dobrott, C. I., Li, L., Ryba, A. R., Bergman, T. B., Kelly, M. C., Kelley, M. W., and Levine, A. J. (2018). Massively parallel single nucleus transcriptional profiling defines spinal cord neurons and their activity during behavior. *Cell Rep.*, **22**(8), 2216–2225.

Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., *et al.* (1998). Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann. Stat.*, **26**(5), 1651–1686.

Schmidhuber, J. (2015). Deep learning in neural networks: an overview. *Neural Networks*, **61**, 85–117.

Schulte-Schrepping, J., Reusch, N., Paclik, D., Baßler, K., Schlickeiser, S., Zhang, B., Krämer, B., Krammer, T., Brumhard, S., Bonaguro, L., De Domenico, E., Wendisch, D., Grasshoff, M., Kapellos, T. S., Beckstette, M., Pecht, T., Saglam, A., Dietrich, O., Mei, H. E., Schulz, A. R., Conrad, C., Kunkel, D., Vafadarnejad, E., Xu, C.-J., Horne, A., Herbert, M., Drews, A., Thibeault, C., Pfeiffer, M., Hippenstiel, S., Hocke, A., Müller-Redetzky, H., Heim, K.-M., Machleidt, F., Uhrig, A., Bosquillon de Jarcy, L., Jürgens, L., Stegemann, M., Glösenkamp, C. R., Volk, H.-D., Goffinet, C., Landthaler, M., Wyler, E., Georg, P., Schneider, M., Dang-Heine, C., Neuwinger, N., Kappert, K., Tauber, R., Corman, V., Raabe, J., Kaiser, K. M., Vinh, M. T., Rieke, G., Meisel, C., Ulas, T., Becker, M., Geffers, R., Witzenrath, M., Drosten, C., Suttorp, N., von Kalle, C., Kurth, F., Händler, K., Schultze, J. L., Aschenbrenner, A. C., Li, Y., Nattermann, J., Sawitzki, B., Saliba, A.-E., Sander, L. E., and Deutsche COVID-19 OMICS Initiative (DeCOI) (2020a). Severe COVID-19 is marked by a dysregulated myeloid cell compartment. *Cell*, **182**(6), 1419–1440.e23.

Schulte-Schrepping, J., Reusch, N., Paclik, D., Baßler, K., Schlickeiser, S., Zhang, B., Krämer, B., Krammer, T., Brumhard, S., Bonaguro, L., *et al.* (2020b). Severe

covid-19 is marked by a dysregulated myeloid cell compartment. *Cell*, **182**(6), 1419–1440.

Shao, H., Jiang, H., Lin, Y., and Li, X. (2018). A novel method for intelligent fault diagnosis of rolling bearings using ensemble deep auto-encoders. *Mech. Syst. Signal Pr.*, **102**, 278–297.

Sharifi-Noghabi, H., Zolotareva, O., Collins, C. C., and Ester, M. (2019). Moli: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics*, **35**(14), i501–i509.

Shen, Z., He, Z., and Xue, X. (AAAI, 2019). Meal: multi-model ensemble via adversarial learning. In *Proc. AAAI Conference on Artificial Intelligence*, volume 33, pages 4886–4893.

Singh, J., Hanson, J., Heffernan, R., Paliwal, K., Yang, Y., and Zhou, Y. (2018). Detecting proline and non-proline cis isomers in protein structures from sequences using deep residual ensemble learning. *J. Chem. Inf. Model.*, **58**(9), 2033–2042.

Singh, J., Hanson, J., Paliwal, K., and Zhou, Y. (2019). Rna secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat. Commun.*, **10**(1), 1–13.

Singh, S., Hoiem, D., and Forsyth, D. (NIPS, 2016). Swapout: learning an ensemble of deep architectures. In *Advances in Neural Information Processing Systems*, pages 28–36.

Smillie, C. S., Biton, M., Ordovas-Montanes, J., Sullivan, K. M., Burgin, G., Graham, D. B., Herbst, R. H., Rogel, N., Slyper, M., Waldman, J., Sud, M., Andrews, E., Velonias, G., Haber, A. L., Jagadeesh, K., Vickovic, S., Yao, J., Stevens, C., Dionne, D., Nguyen, L. T., Villani, A.-C., Hofree, M., Creasey, E. A., Huang, H., Rozenblatt-Rosen, O., Garber, J. J., Khalili, H., Desch, A. N., Daly, M. J., Ananthakrishnan, A. N., Shalek, A. K., Xavier, R. J., and Regev, A. (2019). Intra- and inter-cellular rewiring of the human colon during ulcerative colitis. *Cell*, **178**(3), 714–730.e22.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. (NIPS, 2017). Federated multi-task learning. In *Advances in Neural Information Processing Systems 30*, pages 4424–4434.

Soneson, C. and Robinson, M. D. (2018a). Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods*, **15**(4), 255–261.

Soneson, C. and Robinson, M. D. (2018b). Towards unified quality verification of synthetic count data with countsimQC. *Bioinformatics*, **34**(4), 691–692.

Soneson, C. and Robinson, M. D. (2018c). Towards unified quality verification of synthetic count data with countsimqc. *Bioinformatics*, **34**(4), 691–692.

Song, Y., Zhang, L., Chen, S., Ni, D., Lei, B., and Wang, T. (2015). Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning. *IEEE Transactions on Biomed. Eng.*, **62**(10), 2421–2433.

Squair, J. W., Gautier, M., Kathe, C., Anderson, M. A., James, N. D., Hutson, T. H., Hudelle, R., Qaiser, T., Matson, K. J. E., Barraud, Q., Levine, A. J., La Manno, G., Skinnider, M. A., and Courtine, G. (2021). Confronting false discoveries in single-cell differential expression. *Nat. Commun.*, **12**(1), 5692.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**(1), 1929–1958.

Stegle, O., Teichmann, S. A., and Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nat. Rev. Genet.*, **16**(3), 133–145.

Stephenson, E., Reynolds, G., Botting, R. A., Calero-Nieto, F. J., Morgan, M. D., Tuong, Z. K., Bach, K., Sungnak, W., Worlock, K. B., Yoshida, M., Kumasaka, N., Kania, K., Engelbert, J., Olabi, B., Spegarova, J. S., Wilson, N. K., Mende, N., Jardine, L., Gardner, L. C. S., Goh, I., Horsfall, D., McGrath, J., Webb, S., Mather, M. W., Lindeboom, R. G. H., Dann, E., Huang, N., Polanski, K., Prigmore, E., Gothe, F., Scott, J., Payne, R. P., Baker, K. F., Hanrath, A. T., Schim van der Loeff, I. C. D., Barr, A. S., Sanchez-Gonzalez, A., Bergamaschi, L., Mescia, F., Barnes, J. L., Kilich, E., de Wilton, A., Saigal, A., Saleh, A., Janes, S. M., Smith, C. M., Gopee, N., Wilson, C., Coupland, P., Coxhead, J. M., Kiselev, V. Y., van Dongen, S., Bacardit, J., King, H. W., Cambridge Institute of Therapeutic Immunology and Infectious Disease-National Institute of Health Research (CITIID-NIHR) COVID-19 BioResource Collaboration, Rostron, A. J., Simpson, A. J., Hambleton, S., Laurenti, E., Lyons, P. A., Meyer, K. B., Nikolić, M. Z., Duncan, C. J. A., Smith, K. G. C., Teichmann, S. A., Clatworthy, M. R., Marioni, J. C., Göttgens, B., and Haniffa, M. (2021). Single-cell multi-omics analysis of the immune response in COVID-19. *Nat. Med.*, **27**(5), 904–916.

Strbenac, D., Mann, G. J., Ormerod, J. T., and Yang, J. Y. H. (2015). ClassifyR: an R package for performance assessment of classification with applications to transcriptomics. *Bioinformatics*, **31**(11), 1851–1853.

Su, K., Wu, Z., and Wu, H. (2020). Simulation, power evaluation and sample size recommendation for single-cell RNA-seq. *Bioinformatics*, **36**(19), 4860–4868.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.*, **102**(43), 15545–15550.

Sun, D., Guan, X., Moran, A. E., Qian, D. Z., Schedin, P., Adey, A., and others (2020). Phenotype-guided subpopulation identification from single-cell sequencing data. *bioRxiv*.

Tan, J., Doing, G., Lewis, K. A., Price, C. E., Chen, K. M., Cady, K. C., Perchuk, B., Laub, M. T., Hogan, D. A., and Greene, C. S. (2017). Unsupervised extraction of stable expression signatures from public compendia with an ensemble of neural networks. *Cell Syst.*, **5**(1), 63–71.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., and Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods*, **6**(5), 377–382.

Teichmann, S. and Efremova, M. (2020). Method of the year 2019: Single-cell multimodal omics. *Nat. Methods*, **17**(1), 1.

Tian, Y., Carpp, L. N., Miller, H. E. R., Zager, M., Newell, E. W., and Gottardo, R. (2022). Single-cell immunology of SARS-CoV-2 infection. *Nat. Biotechnol.*, **40**(1), 30–41.

Torrisi, M., Kaleel, M., and Pollastri, G. (2019). Deeper profiles and cascaded recurrent and convolutional neural networks for state-of-the-art protein secondary structure prediction. *Sci. Rep.*, **9**(1), 12374.

Van den Berge, K., Perraudeau, F., Soneson, C., Love, M. I., Risso, D., Vert, J.-P., Robinson, M. D., Dudoit, S., and Clement, L. (2018). Observation weights unlock bulk RNA-seq tools for zero inflation and single-cell applications. *Genome Biol.*, **19**(1), 24.

Vega-Pons, S. and Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *Int. J. Pattern Recogn.*, **25**(03), 337–372.

Vieth, B., Ziegenhain, C., Parekh, S., Enard, W., and Hellmann, I. (2017). powsimr: power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics*, **33**(21), 3486–3488.

Vieth, B., Parekh, S., Ziegenhain, C., Enard, W., and Hellmann, I. (2019). A systematic evaluation of single cell RNA-seq analysis pipelines. *Nat. Commun.*, **10**(1), 4667.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res*, **11**(Dec), 3371–3408.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (ICML, 2008). Extracting and composing robust features with denoising autoencoders. In *Proc. 25th International Conference on Machine Learning*, pages 1096–1103.

Walther, T. C. and Mann, M. (2010). Mass spectrometry–based proteomics in cell biology. *J. Cell Biol.*, **190**(4), 491–500.

Wang, J., Wen, S., Symmans, W. F., Pusztai, L., and Coombes, K. R. (2009). The bimodality index: a criterion for discovering and ranking bimodal signatures from cancer gene expression profiling data. *Cancer informatics*, **7**, CIN–S2846.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. (2015). On deep multi-view representation learning. In *Proc. 32nd International Conference on International Conference on Machine Learning*, pages 1083–1092.

Wang, X., Bao, A., Cheng, Y., and Yu, Q. (2018). Multipath ensemble convolutional neural network. *IEEE Trans. Emerg. Topics Comput.*

West, M. D., Labat, I., Sternberg, H., Larocca, D., Nasonkin, I., Chapman, K. B., Singh, R., Makarev, E., Aliper, A., Kazennov, A., *et al.* (2018). Use of deep neural network ensembles to identify embryonic-fetal transition markers: repression of cox7a1 in embryonic and cancer cells. *Oncotarget*, **9**(8), 7796.

Wilk, A. J., Lee, M. J., Wei, B., Parks, B., Pi, R., Martínez-Colón, G. J., Ranganath, T., Zhao, N. Q., Taylor, S., Becker, W., Stanford COVID-19 Biobank, Jimenez-Morales, D., Blomkalns, A. L., O'Hara, R., Ashley, E. A., Nadeau, K. C., Yang, S., Holmes, S., Rabinovitch, M., Rogers, A. J., Greenleaf, W. J., and Blish, C. A. (2021). Multi-omic profiling reveals widespread dysregulation of innate immunity and hematopoiesis in COVID-19. *J. Exp. Med.*, **218**(8).

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Comput.*, **1**(2), 270–280.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, **5**(2), 241–259.

Wu, Y. and Zhang, K. (2020). Tools for the analysis of high-dimensional single-cell RNA sequencing data. *Nat. Rev. Nephrol.*, **16**(7), 408–421.

Xiao, Y., Wu, J., Lin, Z., and Zhao, X. (2018). A deep learning-based multi-model ensemble method for cancer prediction. *Comput. Methods Programs Biomed.*, **153**, 1–9.

Xie, J., Xu, B., and Chuang, Z. (2013). Horizontal and vertical ensemble with deep representation for classification. *arXiv preprint arXiv:1306.2759*.

Xie, K., Huang, Y., Zeng, F., Liu, Z., and Chen, T. (2020). scAIDE: clustering of large-scale single-cell RNA-seq data reveals putative and rare cell types. *NAR Genom Bioinform*, **2**(4), lqaa082.

Yang, P., Hwa Yang, Y., B. Zhou, B., and Y. Zomaya, A. (2010a). A review of ensemble methods in bioinformatics. *Curr. Bioinform.*, **5**(4), 296–308.

Yang, P., Hwa Yang, Y., B Zhou, B., and Y Zomaya, A. (2010b). A review of ensemble methods in bioinformatics. *Curr. Bioinform.*, **5**(4), 296–308.

Yang, P., Yoo, P. D., Fernando, J. I. E., Zhou, B. B., Zhang, Z., and Zomaya, A. Y. (2014). Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications. *IEEE Trans. Cybern.*, **44**(3), 445–455.

Yang, P., Ormerod, J., Liu, W., Ma, C., Zomaya, A., and Yang, J. (2019a). Adasampling for positive-unlabeled and label noise learning with bioinformatics applications. *IEEE Trans. Cybern.*, **49**(5), 1932–1943.

Yang, P., Humphrey, S. J., Cinghu, S., Pathania, R., Oldfield, A. J., Kumar, D., Perera, D., Yang, J. Y., James, D. E., Mann, M., *et al.* (2019b). Multi-omic profiling reveals dynamics of the phased progression of pluripotency. *Cell Syst.*, **8**(5), 427–445.

Yang, P., Huang, H., and Liu, C. (2021). Feature selection revisited in the single-cell era. *Genome Biol.*, **22**(1), 321.

Yang, Y. H. and Speed, T. (2002). Design issues for cdna microarray experiments. *Nat. Rev. Genet.*, **3**(8), 579–588.

Yuan, X., Xie, L., and Abouelenien, M. (2018). A regularized ensemble framework of deep learning for cancer detection from multi-class, imbalanced training data. *Pattern Recognit.*, **77**, 160–172.

Zacharaki, E. I. (2017). Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ Comput. Sci.*, **3**, e124.

Zappia, L. and Theis, F. J. (2021). Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol.*, **22**(1), 301.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.*, **18**(1), 174.

Zhang, B., Li, J., and Lü, Q. (2018a). Prediction of 8-state protein secondary structures by a novel deep learning architecture. *BMC Bioinf.*, **19**(1), 293.

Zhang, L., Yu, G., Xia, D., and Wang, J. (2019a). Protein–protein interactions prediction based on ensemble deep neural networks. *Neurocomputing*, **324**, 10–19.

Zhang, S., Hu, H., Jiang, T., Zhang, L., and Zeng, J. (2017). Titer: predicting translation initiation sites by deep learning. *Bioinformatics*, **33**(14), i234–i242.

Zhang, X., Xu, C., and Yosef, N. (2019c). Simulating multiple faceted variability in single cell RNA sequencing. *Nat. Commun.*, **10**(1), 2611.

Zhang, X., Zhang, J., Sun, K., Yang, X., Dai, C., and Guo, Y. (IEEE BIBM, 2019b). Integrated multi-omics analysis using variational autoencoders: application to pan-cancer classification. In *Proc. 2019 IEEE International Conference on Bioinformatics and Biomedicine*, pages 765–769.

Zhang, Y., Qiao, S., Ji, S., and Zhou, J. (ICIC, 2018b). Ensemble-cnn: predicting dna binding sites in protein sequences by an ensemble deep learning method. In *Proc. 14th International Conference on Intelligent Computing*, pages 301–306.

Zhao, D., Yu, G., Xu, P., and Luo, M. (2019). Equivalence between dropout and data augmentation: a mathematical check. *Neural Networks*, **115**, 82–89.

Zhu, X., Gong, S., *et al.* (NIPS, 2018). Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems 31*, pages 7517–7527.

Zohora, F. T., Rahman, M. Z., Tran, N. H., Xin, L., Shan, B., and Li, M. (2019). Deepiso: A deep learning model for peptide feature detection from lc-ms map. *Sci. Rep.*, **9**(1), 17168.

Zou, B., Zhang, T., Zhou, R., Jiang, X., Yang, H., Jin, X., and Bai, Y. (2021). deepMNN: Deep Learning-Based Single-Cell RNA sequencing data batch correction using mutual nearest neighbors. *Front. Genet.*, **12**, 708981.