Benchmarking of single cell ATAC sequencing (scATAC-seq) clustering tools

by

Haoyu Yang

A thesis submitted in partial fulfillment for the
degree of Master of Science (Bioinformatics)

in the
Faculty of Science
THE UNIVERSITY OF MELBOURNE

November 2020

THE UNIVERSITY OF MELBOURNE

Abstract

Faculty of Science

Master of Science (Bioinformatics)

by Haoyu Yang

In the past five years, the field of single-cell Assay for Transposase Accessible Chromatin using sequencing (scATAC-seq) has developed quickly with many new sequencing protocols, computational tools and data sets emerging. ScATAC-seq allows profiling of chromatin accessibility of thousands of individual cells. With the large number of computational tools becoming available, it can be hard to choose the most suitable tools for a specific analysis task. This study summarised the currently available computational pipelines, tools or packages specifically designed for scATAC-seq data analysis. We also benchmarked five selected scATAC-seq clustering tools and three clustering tools designed for single-cell RNA sequencing (scRNA-seq) data using a control experiment made up of an equal mixture of cells from five distinct cell lines. We found that all of the tools compared performed well on carefully filtered data, however scATAC-seq clustering tools could not handle noisy data sets, suggesting that careful filtering and parameter tuning during clustering were essential to achieve reasonable results.

Declaration of Authorship

I, Haoyu Yang, declare that this thesis titled, 'Benchmarking of single cell ATAC sequencing (scATAC-seq) clustering tools' and the work presented in it are my own. I confirm that:

- The thesis comprises only my original work towards the masters except where indicated in the preface;
- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the maximum word limit in length, exclusive of tables, maps, bibliographies and appendices as approved by the Research Higher Degrees Committee.

Preface

The cell culture, data generation and pre-processing using Cell Ranger ATAC 1.0.1 were performed by Luyi Tian.

The pre-processing using Cell Ranger ATAC 1.2.0 was performed by Yue You.

Acknowledgements

I would like to express my sincere thanks to my supervisors Dr. Shani Amarasinghe and A/Prof. Matthew Ritchie for their support of my Masters study. The guidance, teaching and encouragement they provided has helped me complete this project and thesis and helped me develop my interest and passion for Bioinformatics.

I would also like to thank my parents who supported me during my study for this degree and have always encouraged me to keep learning and focus on my interests.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Eukaryotic DNA together with histones is efficiently packed into the chromatin in the nucleus, and plays important roles in keeping DNA intact during cell division, preventing DNA damage, and regulating gene expression and DNA replication. Chromatin exists in two states, the euchromatin, which is less condensed, accessible, and allows active transcription and the heterochromatin, which is highly condensed, inaccessible and transcriptionally inactive. To change gene expression, chromatin is actively and dynamically remodelled within the cell in response to particular stimuli or during different developmental stages. Therefore, detecting and understanding the chromatin structure and nucleosome positioning reveals the transcriptional regulation at the chromatin level involved in specific cellular processes and disease states.

In particular, several methods have been developed to examine chromatin accessibility and histone positioning to understand the relationship between chromatin structure and the function in regulating gene transcription. Three major methods include DNase I hypersensitive site sequencing (DNase-seq), Micrococcal Nuclease sequencing (MNase-seq), Nucleosome Occupancy and Methylome sequencing (NOMe-seq) and Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq). The ATAC-seq method in particular, developed in 2013, has become increasingly popular compared to other methods for examining chromatin accessibility. It requires a small number of cells (500 to 50,000) to achieve comparable sensitivity and specificity to other methods. In addition, the method is highly reproducible, simple and fast due to the use of hyperactive Tn5 transposase [5].

Single-cell sequencing technology has been widely used for understanding the heterogeneity of complex tissue and for identifying novel cell types or cell states. Previous efforts of single-cell profiling are mostly performed by measuring the transcriptome using single-cell RNA sequencing (scRNA-seq). scRNA-seq is relatively well developed and around 770 analysis tools are currently available for performing different tasks [6]. In the past five years, assays for profiling chromatin accessibility in single cells have emerged to provide extra information about gene regulation at the epigenetic level. Due to its simplicity and sensitivity, single-cell ATAC-seq (scATAC-seq), developed in 2015, is widely used to profile the chromatin accessibility in 100s to 1000s of cells in a sample to explore heterogeneous cell populations in detail. In general, several scATAC-seq protocols have been developed over the past five years and multiple data sets have been generated. The timeline for scATAC-seq protocol developments and data production is summarised in Figure 1.1.

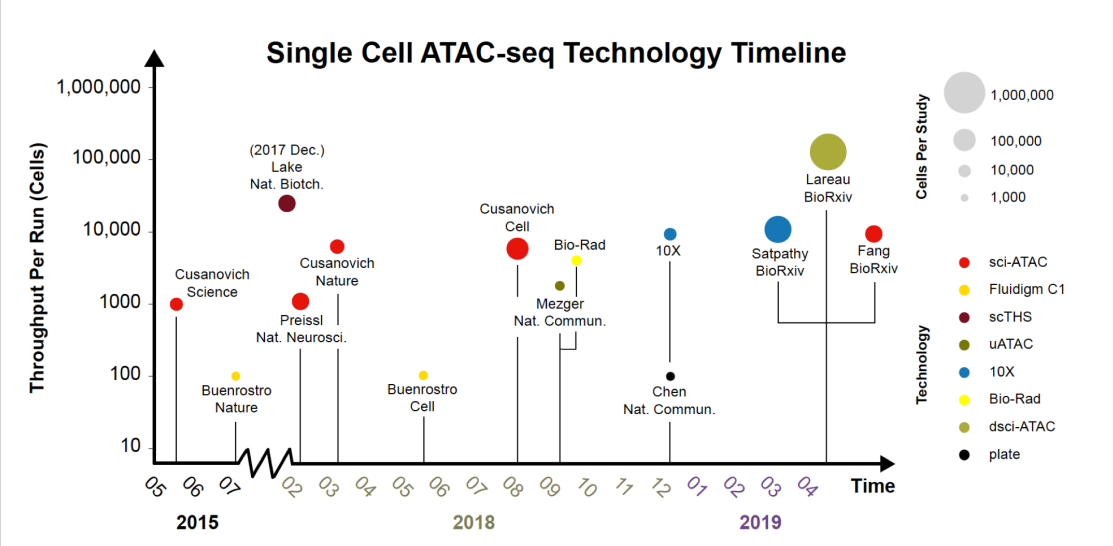


FIGURE 1.1: **Timeline of single-cell ATAC-seq technology development and data generation.** Source: https://github.com/r3fang/SnapATAC/blob/master/notebooks/experiemnt_timeline.md.

The four main protocols for scATAC-seq include the combinatorial indexing approach (sci-ATAC-seq) [7], microfluidics-based methods (scATAC-seq) [8], nano-well based protocols (μscATAC-seq) [9] and droplet-based (10X scATAC-seq, dscATAC-seq and dsciATAC-seq) approaches [1, 2]. The analysis of scATAC-seq data is different from that of scRNA-seq data and not many analysis tools are currently available for scATAC-seq data analysis. Analysing scATAC-seq data is considered challenging because of its sparsity and binary nature. The data is sparse because more features (peaks or bins) throughout

the genome are generated during analysis, while in scRNA-seq the features are usually restricted to genes which are fewer in number. The binary nature of the data means that there are only two copies of DNA that can be sequenced in diploid cells, leading to higher dropout rate and low coverage [10].

The general workflow for scATAC-seq data analysis comprises (1) pre-processing: demultiplexing, adaptor trimming, read mapping, quality control, cell calling and multiplet removal (optional); (2) feature matrix construction: defining regions via peak calling or genome binning, counting defined features, transformation and dimensionality reduction; and (3) downstream analysis: cell clustering, peak calling (optional), visualisation, differential accessibility analysis and cis-regulatory network analysis [10, 11]. Several analysis methods have been generated for scATAC-seq analysis, including APEC[12], ArchR [13], BROCKMAN [14], CellRanger ATAC by 10x Genomics (https://www.10xgenomics.com), ChromSCape [15], chromVAR [4], cicero [16], cisTopic [17], workflow developed by Cusanovich *et al.* [7], Destin [18], epiConv [19], Garnett [20], Gene scoring [2], scABC [21], SCALE [22], ScAsAT [23], scATAC-pro [11], SCATE [24], SCRAT [25], Signac [26], SnapATAC [27], and STREAM [28].

These tools handle different steps in the analysis workflow using different algorithms. To better understand the strengths and weaknesses of different tools, it is important to benchmark their performance on data sets with known ground-truth. This allows data analysts to infer the optimal methods for each step and to develop a comprehensive pipeline for use in practice.

1.1 Objectives

This project aims to comprehensively evaluate scATAC-seq data clustering tools and gaps in analysis workflows using in-house generated scATAC-seq data as well as optimised universal evaluation metrics.

1.1.1 First Objective

The first part of the project comprised of 1) summarising currently available scATAC-seq analysis tools to construct a systematic benchmarking framework; 2) adapting and

developing evaluation metrics and models for scATAC-seq analysis tools.

1.1.2 Second Objective

The second part of the project focused on benchmarking the clustering step of different scATAC-seq tools using the in-house data set with different filtering stringencies.

1.1.3 Third Objective

The third part of the project focused on briefly comparing single cell RNA sequencing (scRNA-seq) clustering tools and simulating more complex data set using scATAC-seq data. Some representative scRNA-seq clustering tools that have previously been shown to perform well would be selected and benchmarked using our in-house scATAC-seq data. The *in silico* simulation of *pseudo-cells* at the read level, rather than at the feature matrix level was also attempted to increase the complexity of input data to make the clustering problem more challenging.

Chapter 2

Literature Review

2.1 Chromatin accessibility and ATAC-seq

In eukaryotic cells, DNA is packaged into arrays of nucleosomes, each consisting of an octamer of histones wrapped by around 147 base pairs (bp) of DNA, and separated by linker DNA, forming the chromatin structure [29–31]. This beads-on-a-string structure is called euchromatin that undergoes active transcription. Heterochromatin, on the other hand, is more compact and inaccessible, which forms the 30 nano-meter fibre structure and does not undergo active transcription. A chromatin region is open or accessible when chromatin-binding factors can physically contact the region, which is historically characterised by nuclease hypersensitivity *in vivo* [32]. Chromatin accessibility is determined by the occupancy of chromatin by nucleosomes and the occupancy by other chromatin-binding factors, such as transcription factors (TFs), RNA polymerase and architectural proteins. In the human genome, around 2-3% of the DNA sequence is accessible and more than 90% of these sequences can be bound by TFs. The post-translational modification and the composition of nucleosomes change chromatin accessibility through altering the binding of TFs through steric hindrance [33] and changing the affinity of nucleosomes to chromatin re-modellers [34]. There are fewer nucleosomes at regulatory regions including enhancers, insulators and transcribed gene bodies [35, 36] so at these regions more chromatin is accessible. The positioning of nucleosomes throughout a genome affects critical cellular functions such as transcription, DNA repair and replication, as it modifies availability of binding sites to TFs, RNA polymerase and other nuclear proteins

[37]. Therefore, collecting and comparing genome-wide chromatin accessibility is important for locating epigenetic changes that accompany cell differentiation, environmental signalling and disease development [38].

Chromatin accessibility is determined through quantifying the susceptibility of chromatin to either enzymatic cleavage of its constituent DNA or methylation. Some assays that directly isolate accessible regions include DNase I hypersensitive site sequencing (DNase-seq) (Figure 2.1a), Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq) (Figure 2.1b), and Nucleosome Occupancy and Methylome sequencing (NOMe-seq) (Figure 2.1d), while the Micrococcal Nuclease sequencing (MNase-seq) (Figure 2.1c) indirectly measures chromatin accessibility.

DNase-seq uses the endonuclease DNase to cleave DNA within accessible chromatin (Figure 2.1). Boyle *et al.* (2008) [40] used a type II restriction enzyme to make a single cut and then ligate adaptors, whereas Hesselberth *et al.* (2009) [41] used enzymes to make double cuts and applied size selection. **ATAC-seq** uses hyperactive transposases (Tn5) to simultaneously cleave and ligate adaptors to accessible DNA. Cleaved fragments with two different adaptors will be PCR amplified and after size selection, short fragments will be sequenced [5]. **MNase-seq** uses the endo-exonuclease MNase to both cleave and eliminate accessible DNA. Thus, regions occupied by nucleosomes and other chromatin-binding factors are isolated and sequenced [42, 43]. **NOMe-seq** uses GpC methyltransferase to methylate accessible DNA. Non-methylated cytosines in the sequence will be converted to uracil following bisulphite conversion, and only accessible regions are sequenced [44]. Among these methods, ATAC-seq has the advantage of requiring fewer cells. Between 500 - 50,000 cells are used, which is especially beneficial for clinical applications. The time required for sample preparation is also short, and it achieves comparable sensitivity and specificity [5].

2.2 Single-cell ATAC-seq (scATAC-seq)

The rapid development of protocols for single-cell RNA-seq profiling has greatly increased our understanding of existing cells and also led to the discovery of new cell types. However, an important piece of information that is often missing from single-cell RNA-seq is the cell-specific genomic regulatory model. The regulatory model comprises

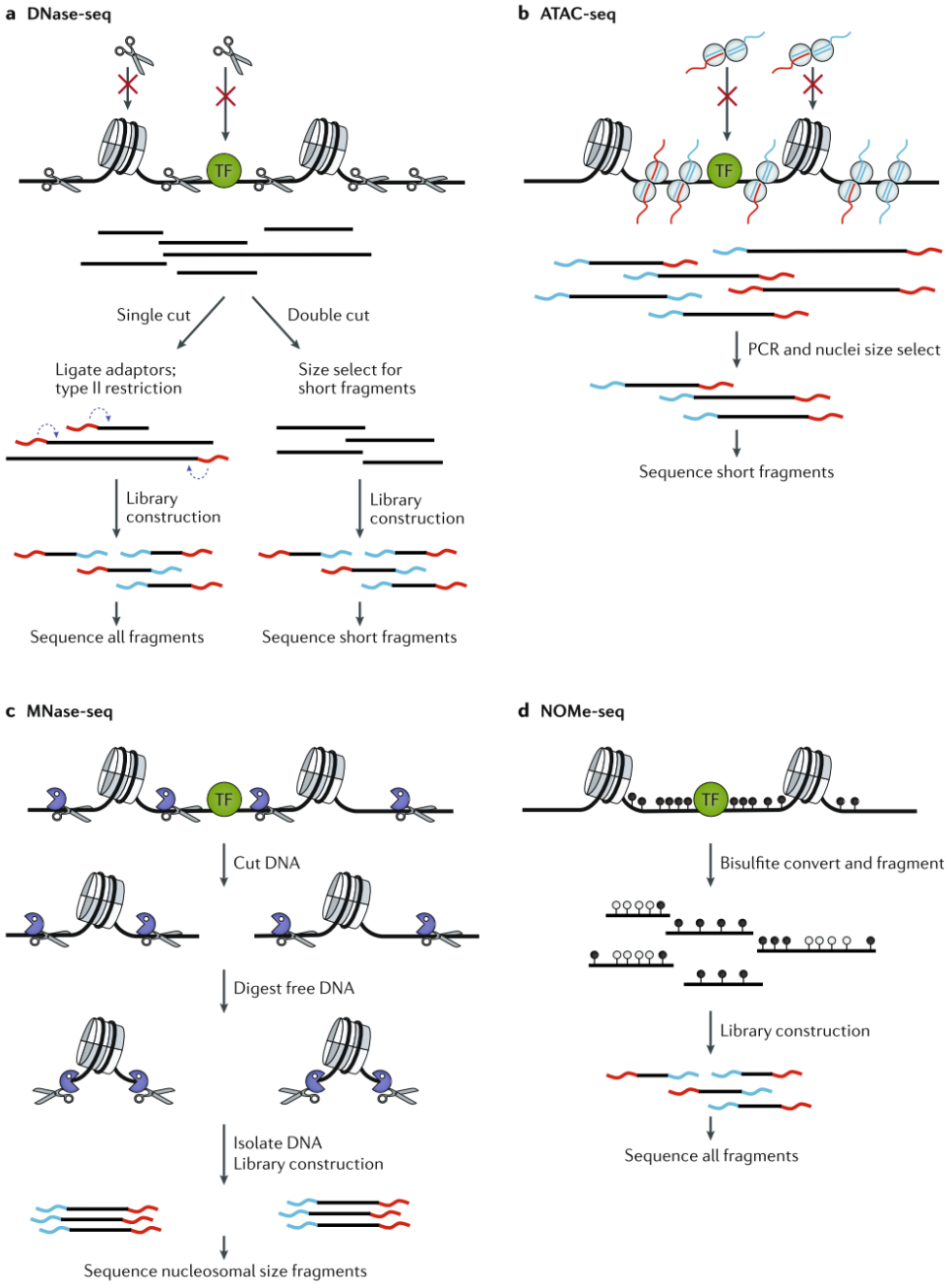


FIGURE 2.1: **Principal methods of measuring chromatin accessibility, adapted from Klemm, Shipony and Greenleaf (2019)** [39]. (a) DNase-seq uses endonuclease DNase to cleave accessible DNA regions (indicated by scissors on the DNA). At the protein-bound position, chromatin is protected against endonuclease cleavage (indicated by the red crosses). Following cleavage, the DNA fragments can be either ligated with adapters at one end after the digestion of type II restriction enzyme (single cut) or digested at both ends and selected for short fragments (double cut). (b) ATAC-seq takes advantage of Tn5 transposases which cleave and ligate adaptors to accessible DNA. (c) In MNase-seq, the MNases, which are both endonuclease and exonuclease, cleave and digest accessible DNA and the inaccessible DNA fragments are sequenced. (d) In NOMe-seq, accessible DNA are methylated (indicated by the black pins) by GpC methyltransferase followed by bisulphite conversion of non-methylated cytosine to uracil. Therefore, the presence of cytosine indicates that the corresponding DNA regions are accessible.

of heterogeneous enhancers, promoters, and insulators that are important for the modulation of single-cell gene expression in a spatiotemporal continuum. Due to its simplicity and sensitivity, in recent years, ATAC-seq has been widely used to measure single-cell chromatin accessibility.

scATAC-seq data can be generated by four major approaches: a) combinatorial indexing through split-pooling [2, 3], b) microfluidics-based methods [8], c) nano-well-based methods [9] and d) droplet microfluidics [1, 2] (Figure 2.2).

In the combinatorial indexing method **(sci-ATAC-seq)** (Figure 2.2a), the isolated nuclei are split into wells and uniquely barcoded with Tn5 transposases for the first round. These nuclei are pooled and redistributed randomly into wells for the second round of barcoding. This method takes advantage of the low probability of having two cells in the same wells in both rounds, thus there is no need for isolation of cells [3]. This method has been successfully used to study the embryonic development in *Drosophila melanogaster* [46], to study transcriptional regulation of developing mouse forebrains [47] and create a single-cell atlas across 13 adult mouse tissues [7]. **scATAC-seq**, developed by Buenrostro *et al.* (2015) enables the capture of single-cell nuclei through a microfluidic device (Fluidigm, C1) [8] (Figure 2.2. This method has been used to profile thousands of cells during hematopoiesis [45, 48]. The nano-well-based **μscATAC-seq** (Figure 2.2c), further increases the throughput of measurements [9]. Individual cells are isolated in nano-wells. In each well, cell-specific transposition barcoding and PCR amplification to build scATAC libraries are performed. Even though the scalability is poor for both scATAC-seq approaches compared to the combinatorial indexing method, the single-cell library complexity is higher, which is critical for reducing the sparsity of scATAC-seq results [3, 8, 39, 45].

Bio-Rad Laboratories [2] (Figure 2.2d) developed droplet-based microfluidic method, which provide similar data quality compared to the previous two microfluidic methods [39]. It also provides an option that allows multiple gel beads captured in the same emulsion drop to increase throughput and a demultiplexing analysis tool was developed to address the problem of one cell having multiple barcodes [2]. To further increase throughput, combinatorial indexing and droplet methods were used together in the **droplet-based single-cell combinatorial indexing for ATAC-seq (dsciATAC-seq)** [2] approach (Figure 2.2e).

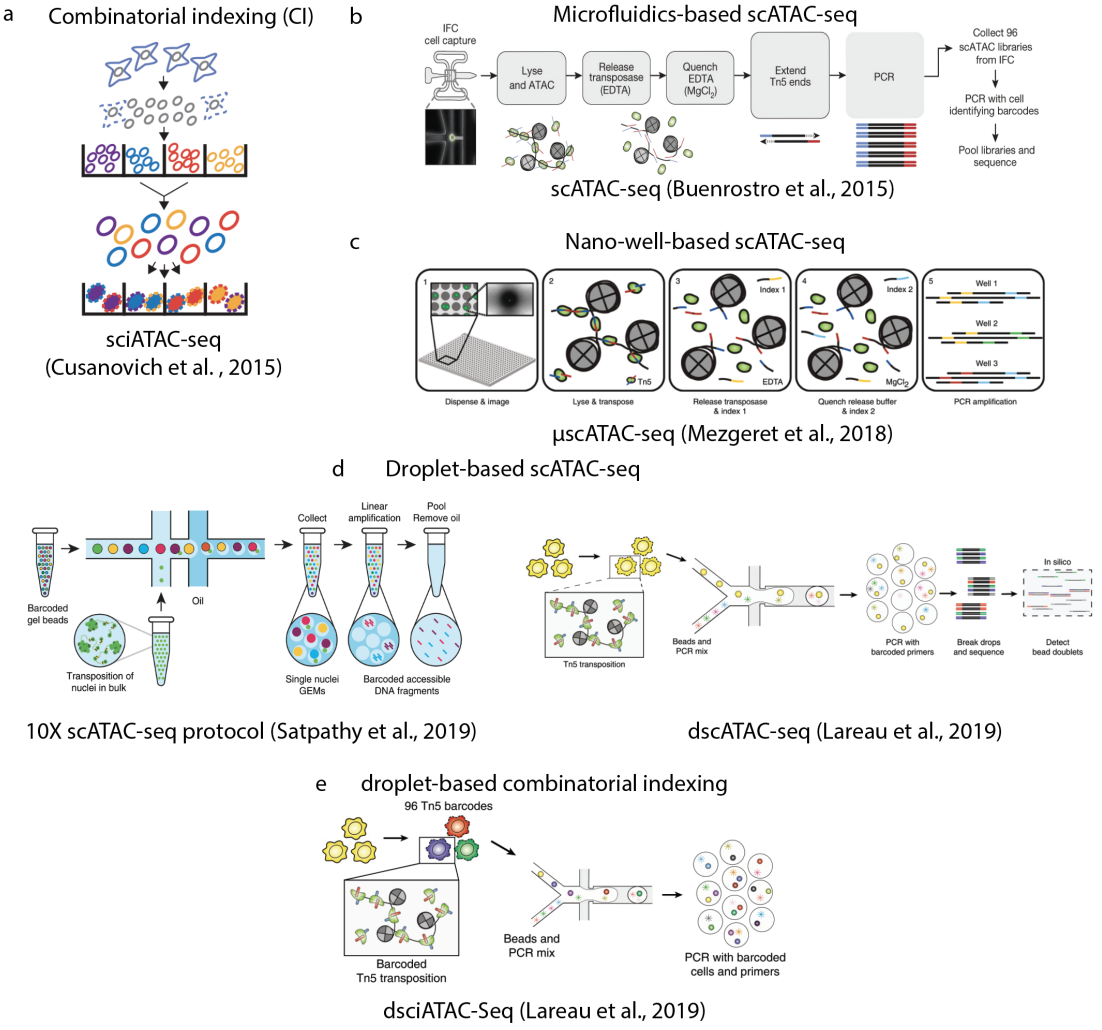


FIGURE 2.2: **Protocols for single-cell ATAC-seq (scATAC-seq)** [1–3, 8, 9, 45]. (a) In combinatorial indexing methods, nuclei are isolated and molecularly tagged in bulk with barcoded Tn5 transposases in wells (different barcodes are represented by the different colours outlining the nuclei). Nuclei are then pooled and redistributed into wells and a second barcode (represented by the colour filling each nucleus) is introduced during PCR (b) In microfluidics-based scATAC-seq, individual cells are captured using microfluidics platform (Fluidigm). The isolated cells are transposed, and fragments are amplified by PCR on the integrated fluidics circuit (IFC). Then in 96-well plates, with each well containing one scATAC library, the scATAC libraries are amplified and barcoded through PCR. (c) In nano-well-based scATAC-seq (μscATAC-seq), individual cells are isolated in nano-wells and the cell-specific transposition barcoding and PCR amplification is carried out to build scATAC libraries within wells (d) In droplet-based 10X scATAC-seq and dscATAC-Seq, a pool of transposed nuclei is loaded into a droplet-microfluidics system and they are simultaneously encapsulated into single droplet emulsion with PCR reagents and barcoded gel beads. Following droplet PCR, droplet-specific barcodes are added to transposed DNA and the scATAC libraries are generated (e) In dsciATAC-seq, nuclei are distributed in wells similar to the combinatorial indexing method and transposed with well specific barcodes. The transposed nuclei are then pooled and loaded in droplet microfluidics device and the rest of processes are the same as dscATAC-seq.

The protocol developed by **10X Genomics** (i.e. 10X-ATAC) [1] is in general similar Bio-Rad dscATAC-seq method. In the 10X method, nuclei are first isolated and transposed with Tn5 transposase. Then each nucleus will be encapsulated by a droplet with a gel bead containing barcodes. After linear amplification, the DNA fragments are barcoded and later emulsion break open the droplets to allow the barcoded DNA to be pooled for PCR amplification and high-throughput sequencing [1]. The main difference between 10X-ATAC and dscATAC-seq is that dscATAC-Seq allows multiple beads in one droplet to increase library complexity, while 10X-ATAC provides larger close-packed hydrogel beads which allow a certain level of control over the number of beads loaded into one droplet.

2.3 Characteristics of scATAC-seq data

scATAC-seq data is highly sparse compared to scRNA-seq data. For an expressed gene, there may be several copies of RNA molecules within a cell to be sequenced while there are only a few copies of DNA (two in diploid organisms) for scATAC-seq assays. This results in the detection of only 1 - 10% of the expected accessible peaks in the Fludigm C1 platform [8]. Therefore, the method used to recover informative features from such sparse data is critical for measuring chromatin accessibility using this technology [10].

The library structure generated by 10X-ATAC [1], dscATAC-seq [2] and sciATAC-seq [3] are shown in Figure 2.3. The 10X-ATAC protocol generates four `fastq` files which are read 1 (50 bp), read 2 (50 bp, reverse direction), the 10X cell barcode (16 bp) and sample index (8 bp). The library structure generated by dscATAC-seq [2] is similar to 10X-ATAC library in general, that is the cell barcode is at one side of the chromatin fragment and the sample index is on the other side. However, the cell barcode is composed of three 7 bp barcode fragments. One `fastq` file is 118 bp containing the cell barcode, read 1 and other sequences. The other two `fastq` files are 40 bp read 2 in the reverse direction and an 8 bp sample index. The library structure of sciATAC-seq [3] is different, with a barcode fragments on each side of the chromatin fragment. Two of the `fastq` files are read 1 and read 2 in a reverse direction. One `fastq` file contains half of the barcode and other sequences. The second `fastq` file contains the other half of the barcode and the sample index.

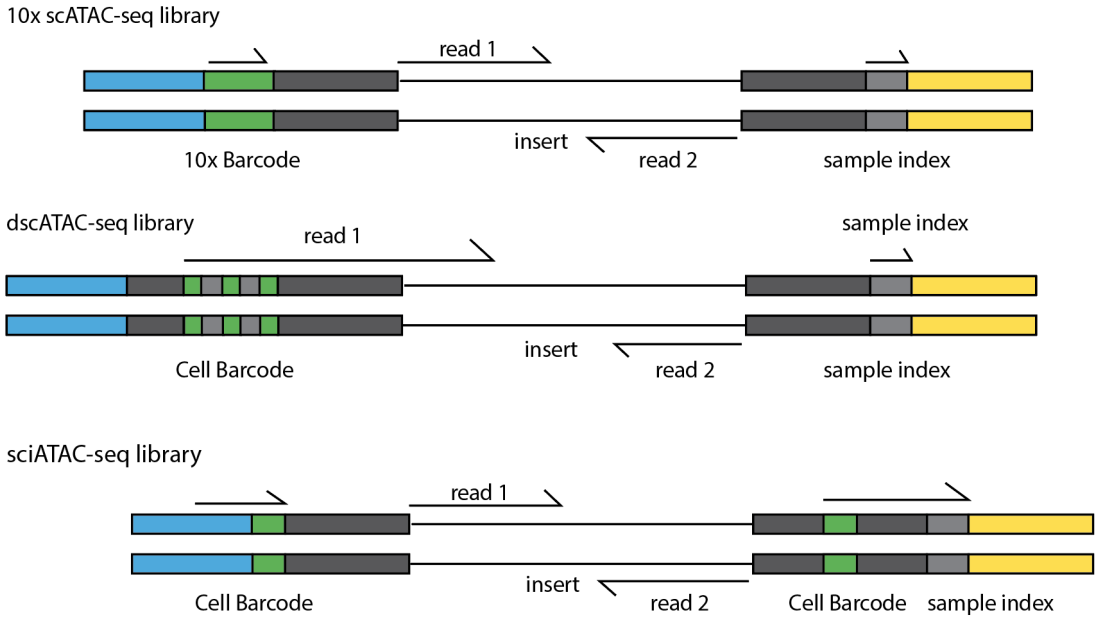


FIGURE 2.3: **The expected library and read structure generated from 10X-ATAC [1], dscATAC-seq [2] and sciATAC-seq [3].** The green blocks are cell barcodes, the light gray blocks are sample indexes and arrows indicate reads generated by sequencing.

2.4 Example applications of scATAC-seq

Single-cell ATAC-seq has been applied to answer a range of different biological questions. Here I summarise several recently published data sets generated from scATAC-seq applications and describe the major findings from each study.

TABLE 2.1: **Examples of publicly available scATAC-seq data sets**

Data set	Methods	Properties	Cell Numbers	Reference
Human hematopoietic lineage	scATAC-seq	continuous	2,034	Buenrostro *et al.*, 2018, GSE96772
***Drosophila* embryos**	sciATAC-seq	continuous	23,085	Cusanovich *et al.*, 2018, GSE101581
Mouse Cell Atlas	sciATAC-seq	large data set	81,173	Cusanovich *et al.*, 2018, GSE111586
Human hematopoeitic cells	10X-ATAC	large data set	61,806	Satpathy *et al.*, 2019, GSE129785, 10X-ATAC PBMC
Basal cell carcinoma tumor microenvironment (TME)	10X-ATAC	cancer	37,818	Satpathy *et al.*, 2019, GSE129785

2.4.1 scATAC-seq of human hematopoietic cells (Buenrostro *et al.*, 2018)

The data set was generated on 2,034 cryopreserved $CD34^+$ cells in bone marrow from 6 human donors. The cells were first isolated by fluorescence-activated cell sorting (FACS) followed by cryopreservation. Then scATAC-seq [8] was performed on each cell type. The cells comprise of hematopoietic stem cells (HSCs), multipotent progenitors (MPPs), lymphoid-primed multipotent progenitors (LMPPs), common-myeloid progenitors (CMPs), granulocyte-macrophage progenitors (GMPs), megakaryocyte-erythrocyte progenitors (MEPs), common-lymphoid progenitor (CLPs), plasmacytoid dendritic cells (pDCs), monocytes, and an uncharacterised $CD34^+CD38^-CD45RA^+CD123^-$ population. The median number of fragments per cell was 8,268 with 76% of them mapping to peaks, resulting in a median of 6,442 fragments in peaks per cell [48].

Buenrostro *et al.* (2018) constructed a chromatin accessibility landscape of human hematopoiesis to characterise differentiation trajectories. They use ChromVAR to infer TF activity by calculating TF motif-associated chromatin accessibility changes of each cell. Based on this analysis, they found that HSCs exhibit low levels of lineage specifying motifs and high levels of motif regulating stem cell activity, but these were reversed in more differentiated cells. They also observe heterogeneity within CMPs and GMPs and develop a strategy to partition GMPs along their differentiation trajectory. Furthermore, they integrated scRNA-seq data with scATAC-seq date to associate transcription factors to chromatin accessibility changes through correlations of expression and regulatory element accessibility, which provided a computational method for integrative exploration of complex regulatory dynamics in a primary human tissue at single-cell resolution [48].

2.4.2 sciATAC-seq of *Drosophila* embryos (Cusanovich *et al.*, 2018)

Using combinatorial indexing assay (sciATAC-seq), Cusanovich *et al.* (2018) profiled chromatin accessibility in 23,085 single nuclei from hundreds of fixed *Drosophila melanogaster* embryos across three landmark embryonic stages: 2 to 4 hours after egg laying (AEL), 6 to 8 hours AEL, and 10 to 12 hours AEL. They revealed the spatial heterogeneity in chromatin accessibility of regulatory genomic regions before gastrulation, which aligns with the future cell fate. During mid embryogenesis, cell types can be inferred by

their chromatin accessibility, while maintaining a signature of their germ layer of origin. They identified over 30,075 distal elements with tissue-specific accessibility. Their work demonstrated the power of scATAC-seq in profiling of embryos to resolve dynamic changes during development, and to uncover the cis-regulatory programs of germ layers and cell types [46].

2.4.3 sciATAC-seq atlas of mouse tissues (Cusanovich *et al.*, 2018)

Cusanovich *et al.* (2018) applied the combinatorial indexing assay, sciATAC-seq, to profile genome-wide chromatin accessibility in 81,173 single cells from adult mouse tissues. The data set was generated on 13 tissues from 5 8-week-old male C57BL/6J mice, including bone marrow, cerebellum, heart, kidney, large intestine, liver, lung, prefrontal cortex, small intestine, spleen, testes, thymus, and whole brain, using sciATAC-seq methods [3]. For tissues from bone marrow, large intestine, lung and whole brain, a replicate sample from a second mouse were collected. The total number of cells profiled per tissue (after filtering) ranged from 2,278 for cerebellum to 9,996 for lung (two samples).

They identified and annotated 85 distinct patterns of chromatin accessibility based on the accessibility score of each cell at the predefined 436,206 potential regulatory elements using t-distributed Stochastic Neighbor Embedding (t-SNE) and Louvain clustering. Besides this, they linked regulatory elements to their target genes using Cicero [16] to define the TF motif specifying each cell type and to identify heterogeneity within cell types. Furthermore, they developed a strategy for mapping scRNA-seq data to sciATAC-seq data, to facilitate the comparison of atlases. Finally, they identified cell-type-specific enrichments of the heritability signal for hundreds of complex traits, by integrating mouse chromatin accessibility with human genome-wide association summary statistics [7].

2.4.4 Human hematopoeitic cell and basal cell carcinoma tumor microenvironment (TME) study (Satpathy *et al.*, 2019)

Satpathy *et al.* (2019) generated scATAC-seq profiles of 61,806 cells from peripheral blood and bone marrow from 16 healthy individuals using the commercial system 10X scATAC-seq. The cell types range from bone marrow progenitor cells to multiple types

of differentiated immune cells including B cells, T cells and NK cell. They identified 31 clusters by performing Latent Semantic Indexing (LSI) and Shared Nearest Neighbor (SNN) clustering [49]. Then to classify these clusters, they performed three strategies: (1) classified clusters based on the neighbouring genes of cluster-specific *cis*-elements; (2) calculated gene activity scores, which are the aggregate accessibility of several enhancers linked to a single gene promoter [16] and (3) used TF motifs, computed from the accessibility of TF binding sites genome-wide in each single-cell [4]. They also demonstrated that using the first strategy can identify cell type-specific cis-elements even for a single gene. All of these strategies did not involve pre-labelling of cell types [1].

In basal cell carcinoma, they generated scATAC-seq profiles of 37,818 cells from biopsies of pre- and post- anti-programmed cell death protein 1 (PD-1) treatment from 7 patients for cell types including T cells, non-T immune cells, stromal cells and tumour cells. After analysis, they revealed the regulatory elements in cancer, stromal and immune cells in the tumour microenvironment. By comparing cells from before and after PD-1 treatment, they identified cell types that were sensitive to the therapy and revealed a shared regulatory program that governs intratumoral CD8+ T cell exhaustion and CD4+ T follicular helper cell development [1].

2.5 A general workflow for scATAC-seq data analysis

The general workflow for scATAC-seq data processing includes three major steps: (1) data pre-processing, (2) feature matrix construction and (3) downstream analysis (Figure 2.4).

2.5.1 Data pre-processing

Before answering the biological questions through the different downstream analysis tools, pre-processing steps are mandatory and essential for data arrangement and cleaning because inappropriate pre-processing steps could lead to inaccurate downstream analysis and therefore, generating misleading results.

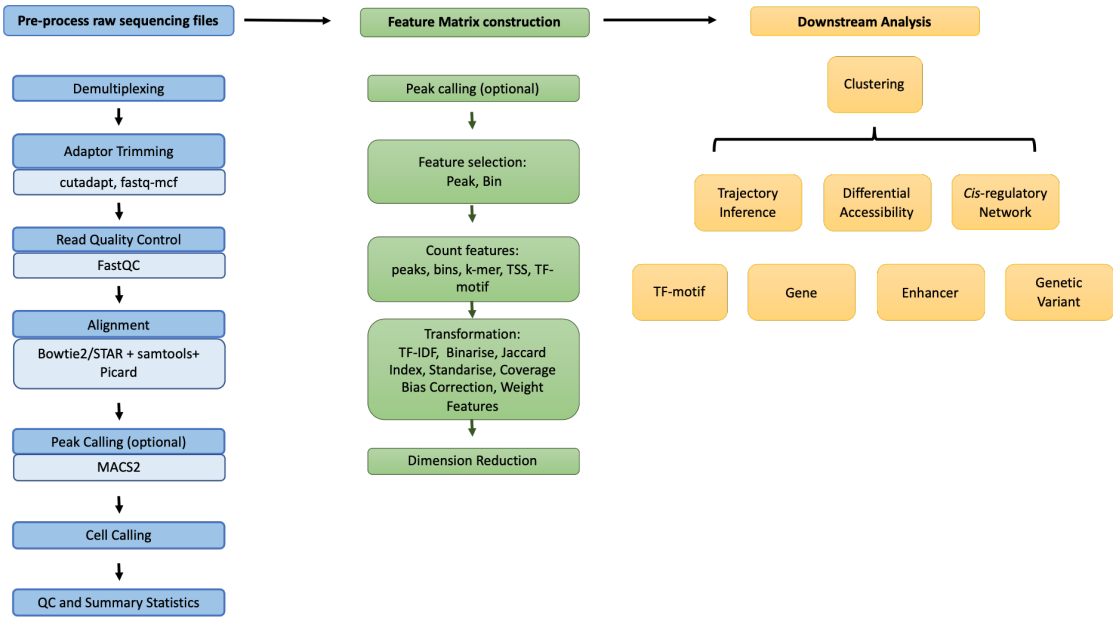


FIGURE 2.4: **A general workflow for scATAC-seq data analysis, adapted from Chen *et al.* (2019) [10] and Baek *et al.* [50].** scATAC-seq data undergo three major steps: pre-processing, feature matrix construction and downstream analysis. The pre-processing step includes demultiplexing, adaptor trimming, reads quality control, mapping, cell calling and optional peak calling before identifying cells. The feature matrix construction step takes mapped reads as input with some tools performing peak calling in advance to construct peak matrix. Other tools may use different features, e.g. fixed size bin. Thes the reads or fragments in each feature for each barcode are counted and filtered. The count matrix then undergoes transformation and dimension reduction for the downstream analysis. For downstream analysis, clustering, trajectory inference, differential accessibility analysis, cis-regulatory network are usually performed. Some biological questions related to TF-motif, gene, enhancer or genetic variation can be answered based on the downstream analysis results.

2.5.1.1 Sequencing read pre-processing

After sequencing, the `BCL` files (Illumina's sequencer base called file format) need to be converted into `fastq` format. One common tool for this is bcl2fastq developed by Illumina. The next step is demultiplexing the sequence files, where the barcode from the barcode `fastq` file is added to the name section of each read to facilitate further analysis. Next, the `fastq` files are generally passed through a general quality control (QC) step and adapter trimming (optional). Lots of tools have been developed for this step such as FastQC [51], cutadapt [52], Trimmomatic [53] and Trim Galore! [54]. The remaining fragments are then aligned to a reference genome using programs such as BWA [55] and bowtie2 [56]. After mapping, samtools [57] can be used to filter out low-quality reads, sort, index and mark duplicates.

Mapping statistics and quality control metrics can then be calculated. Commonly calculated metrics include the total number of reads, the total number of mapped reads, the mapping rate, fraction of reads mapping to the mitochondrial genome, number of duplicate reads, number of high-quality reads, library complexity, the fraction of reads in annotated genomic regions and the TSS enrichment profile [11].

2.5.1.2 Cell calling

Cell calling is the process of identifying barcodes that represent 'real' cells versus those that represent noise barcodes. This step is essential because not all barcodes represent true cells due to cell collisions and/or cell debris. This is still a challenging step and different pipeline or analysis strategies use different methods. The first major method involves model fitting. The cell calling step of Cell Ranger ATAC [58] requires peaking calling in advance, and considering the number of high-quality fragments overlapping peaks. It first removes the 'low-targeting' barcodes that have a lower fraction of fragments in peaks than the fraction of genome in peaks. It also removes repeated minor barcode of the doublets. Then the cell calling step, after subtracting a small amount of fixed contamination count, fits a mixture model of two negative-binomial distributions to distinguish cell barcodes from non-cell barcodes. The second method is based on using summary statistics such as the total number of fragments per barcode or fraction of barcodes in peak regions, thus low-quality barcodes will be filtered out. Signac [26] filters out barcodes by considering the number of fragments in peaks, percentage reads in peaks, blacklist ratio, nucleosome signal and TSS enrichment. This method is also subject to users' setting of the threshold. scATAC-pro [11] also adopted the same strategy, by providing multiple summary statistics including the total number of unique fragments, the fraction of fragments in peaks, fraction of fragments in the mitochondrial genome, and fraction of fragments overlapping with annotated promoters, enhancers, and TSS regions. SnapATAC [27], however, does not require the peak calling step. It considers the number of unique fragments and fragments in promoter ratio to determine the cell barcodes. It keeps the barcodes with high unique fragments and high promoter fragments. The parameters for determining "high" is subject to the user setting.

2.5.2 Peak calling

Peak calling can occur as part of feature matrix construction or during downstream analysis. It is a process that identifies regions that are enriched by the reads or fragments. The currently available scATAC-seq analysis tools use tools for bulk ATAC-seq at this step MACS2 [59] is a commonly used peak calling method that has been adapted by many analysis tools or workflows. ZINBA [60] is another method that is adapted by Cell Ranger ATAC [58].

2.5.3 Feature matrix construction

Generating an accurate feature matrix is crucial for correct downstream analyses of scATAC-seq data. The reads of cell barcodes selected by the Cell Calling step are used to generate the feature matrix. Multiple approaches developed so far to achieve this step are focused on either, binning of reads to regions of the genome [27] or the peaks to regulatory regions [4]. Majority of them employ one feature matrix construction method, while some tools construct multiple types of matrices for different downstream analysis purposes. As summarised in Chen *et al.* (2019) [10] the process of constructing a feature matrix includes 1) defining regions, 2) counting features, 3) transformation and 4) dimension reduction. Different methods are developed for each sub-step and the current tools combine them in different ways (Figure 2.4).

Defining regions

There are 5 main types of regions defined by current scATAC-Seq tools; a) peak regions called on respective bulk ATAC-Seq data, or from public data [4, 16, 17, 21, 23], b) peak regions on aggregated single-cells [4, 16, 17, 21, 23], c) transposon integration sites (BROCKMAN [14]) d) known chromatin regions containing certain TF motifs (SCRAT [25]) e) whole-genome sectioned into uniform bins (SnapATAC [27]) or a subset of it [7]). It has to be noted that defining regions using known regions and/or peaks identified from true bulk or pseudo-bulk regions may fail to call peaks that are only observed in rare cell types.

Counting features

Raw feature counting within the defined regions can be achieved by either by a) counting the reads overlapping the regions (i.e. peaks, bins, TF motifs), b) counting k-mers or gapped k-mers overlapping the regions [4, 14]. Availability of gene annotation is used by some tools to generate a gene enrichment score (i.e. gene activity) based on reads nearby to a specific gene [2, 16, 25].

Transformation

Most methods convert the feature matrix into a binary matrix assuming that each feature could be either "open" or "closed" in accessibility (for diploid organisms) and the resulting count matrix is extremely sparse [7, 16, 17, 23, 27]. An advantage of a binary matrix is it allows one to overlook the technical issues arising from low sequencing coverage and PCR artifacts. Transformations that are currently enforced on these binary matrices by existing tools include the Jaccard index [23, 27], term frequency inverse document frequency transformation (TF-IDF) [7], weighting by co-accessibility [16], weighting by a decaying function based on the distance to gene TSS [2], sample depth correction [4, 27], z-scores to measure the gain and loss of chromatin accessibility across cells [4, 14]. Transformation of the feature-cell matrix is required to partially improve the data sparsity. The TF-IDF strategy is commonly used in text mining, which weights the rarer peaks higher and the common peaks lower [7]. Jaccard distances are used to measure the dissimilarity of accessible matrix between two cells through calculating the ratio of the number of unique peaks of a cell against all peaks in two cells [23].

Dimension reduction

The feature-cell matrix is in high sparsity, therefore the dimension reduction is performed to remove redundant information and potential noise and to be suitable for data representation. Principal component analysis (PCA) is the most widely used linear dimension reduction method, and the number of principal components to be chosen could be specified by users or determined by the elbow of scree plot analysis or Jack-straw test [7, 14, 27]. Topic modelling methods (e.g., cisTopic) chooses the top topics (which are latent features) according to the topic-cell distribution generated by latent Dirichlet allocation (LDA), a method used in natural language processing (NLP) [61]. Latent semantic indexing (LSI) is a simple model usually used in topic modelling, which performs TF-IDF on a binary matrix to normalise sequencing depth and up-weight rarer features, followed by the singular value decomposition (SVD). The TF-IDF transformed

matrix can also be the input of PCA. Multidimensional scaling (MDS) is also used for dimension reduction [23].

2.5.4 Downstream analysis

The general purpose of the single-cell study is to understand the difference between sub-groups of a tested population. The downstream analysis addresses several aspects of the differences. The downstream analysis could include clustering, visualisation, peak calling (optional), trajectory inference, differential accessibility analysis, motif enrichment analysis, motif foot-printing and cis-regulatory network construction [50].

The goal of clustering is to classify cells into sub-groups based on the feature matrix obtained in the previous step. Each cluster could represent a cell type or cell states in the tested population and the clustering helps us identifying new cell types or states and separate heterogeneous cells. Common methods include Model-Based Clustering (mclust) [62], DBSCAN, Hierarchical Clustering, K-means, weighted K-medoids [63], Louvain [64] etc. Then the clusters need to be annotated to identify the cell types.

After obtaining distinct cell sub-groups, peak calling is sometimes performed again to identify the cluster-specific accessible chromatin regions, which are then used for differential accessibility analysis.

For visualisation, high dimensional data are projected onto a 2D plane. t-Distributed Stochastic Neighbor Embedding (tSNE) [65] and Uniform Manifold Approximation and Projection (UMAP) [66] are commonly used in most of the analysis tools for this task.

Trajectory inference is to infer the cell differentiation trajectory based on the accessibility changes of the cells. Differential accessibility analysis aims to identify cell-type-specific regulatory elements. *Cis*-regulatory network analysis identifies annotated and potential enhancer regions of certain gene promoters, which is also known as co-accessibility analysis.

2.6 Existing scATAC-seq data analysis tools

In this thesis, 22 scATAC-seq specific analysis tools are summarised in Figure 2.5 through reading corresponding publications and their `GitHub` pages. The summary is in general

form, thus many specific functions or features of each tool are not summarised in details. Most tools can perform pre-processing and generate a feature matrix but some of them only focus on the downstream analysis (e.g. Garnett). The pre-processing steps could involve the data cleaning pipelines by using wrapper functions of command-line tools or just involve cell and/or feature filtering. The most obvious difference among the tools is the feature matrix construction methods. Figure 2.5 only summarised the feature type on the resulting matrix, however, the algorithms for constructing the feature matrix are not shown. The clustering algorithms for each tool are also listed. Although some tools use the same algorithm (e.g. APEC, Cicero, scATAC-pro etc use Lovain clustering), the implementation may not be the same. Only APEC, BROCKMAN, CellRanger ATAC, Destin, SnapATAC, scATAC-pro can take fastq files as input to perform the full workflow of pre-processing steps. Some tools can take the output of CellRanger ATAC as input (the 10X column), including ArchR, Destin, APEC, SnapATAC, scATAC-pro, Signac, epiConv, cisTopic, cicero, SCALE and STREAM. Cicero, APEC, STREAM and ArchR are also able to perform trajectory analysis. The feature matrix generated by other tools can also be used to perform trajectory analysis using these tools. Only SnapATAC and Cicero can analyse co-accessibility.

The following section explains some scATAC-seq tools in detail.

Cusanovich *et al.* [7] generated an analysis workflow that has been used to analyse a mouse cell atlas data set and a *drosophila* embryonic development data set. Before peak calling, it clusters cells using latent semantic indexing (LSI). A bin by cell binary matrix is first constructed as mentioned in Section 2.4 (2). Then commonly used bins are selected and cells with low read counts are filtered out. The normalisation of reads is performed using the term frequency-inverse document frequency transformation (TF-IDF) followed by dimensionality reduction using singular value decomposition (SVD). After these steps, the first-round of clustering is performed (referred to as *'in silico cell sorting'*) to generate clades. To identify specific regulatory elements within each clade, peaks are called using MACS2 within each cluster and combined into one .bed file. Finally, the clusters are refined with a second-round of clustering after TF-IDF and SVD based on read counts from the peaks called previously.

SCRAT [25] was designed for analysing single-cell regulome data, with an online web

graphical user interface and an R package. It covers the pre-processing, feature summarisation, clustering, differential accessibility analysis and infers cell identity. The input is `bam` files and an option is provided to exclude any signal from the ENCODE blacklist regions [67]. It combines read counts on different regulatory features such as TF binding motifs, gene TSS regions and user-defined features. It provides PCA and t-SNE as dimension reduction methods and multiple clustering methods including Model-Based Clustering (mclust) [62], DBSCAN, Hierarchical Clustering and k-means. It can also perform differential accessibility analysis. For each feature, statistical tests including parametric (t-, ANOVA F-) or non-parametric (Wilcoxon rank-sum, Kruskal-Wallis or permutation) tests can be performed between the clusters, to identify differential features. Features that pass a particular false discovery rate threshold are then reported.

chromVAR [4] takes the input of aligned sequences, a peak file (determined from either bulk reference or aggregated single-cell data) and then estimates the dispersion of chromatin accessibility within peaks sharing the same feature, e.g. TF motifs or k-mers. It sampling 'background' peaks from all peaks for each defined feature, and the background peaks have the same GC content and fragment count as the observed peaks. The 'background' peaks are used to compute bias-corrected accessibility deviation or z-score among all cells for each defined feature. This accessibility deviation or z-score is used for downstream clustering. For downstream analysis, firstly it has an interactive web application that can show clusters and deviation score based colouring in all clusters for the selected gene. Secondly, it can analyse the correlation and potential cooperativity between TF binding sites. Thirdly, it can sort features based on their variability across cells.

scABC [21] takes `bam` file and peak file obtained from aggregation of all cells. It first calculates a global weight for each cell based on the number of distinct reads in the peak regions. Based on these weights, it then uses weighted k-medoids [63] to cluster cells based on the read counts within peaks. Then to improve clustering, it calculates landmarks for each cluster, which are the P peaks with the highest read count, where P are user-defined. The assignment of cells to sets of landmarks (i.e. clusters) is based on Spearman correlation. Further, differential accessibility is obtained using an empirical Bayes regression-based hypothesis testing procedure.

Cicero [16] also first groups similar cells by calculating a gene activity score based on

accessibility at a promoter region and the regulatory potential of peaks nearby. The special feature of **Cicero** is that it identifies all co-accessible regions to build a *cis*-regulatory map.

BROCKMAN [14] represents genomic sequences by gapped k-mers within transposon integration sites and infers the variation in k-mer occupancy using principal component analysis (PCA). It is designed for identifying differentially active TFs and TF-TF interactions.

Cell Ranger ATAC is a set of analysis pipelines for Chromium scATAC-seq data developed by 10X Genomics. It also uses peak information for clustering with a theoretical disadvantage of not being able to identify rare peaks appearing only in very rare cell populations. It supports multiple dimension reduction methods, including PCA, LSA and PLSA and multiple clustering methods, such as k-means, graph-clustering for PCA and Spherical k-means, graph-clustering for LSA and PLSA.

Destin [18] first generates two-weight peak matrices. The first one up-weights the distal regulatory regions over proximal regulatory regions based on their distances to the TSS. The second up-weights the less shared accessibility peaks with reference to chromatin accessibility peaks using DNase I hypersensitive site (DHS) data [67]. The two matrices are multiplied and weighted PCA is performed followed by k-means clustering to group cells with similar chromatin accessibility profiles.

The Gene Scoring method [2] assigns each gene an accessibility score by summarising peaks near its TSS and weighting them by an exponential decay function based on their distances to the TSS.

SnapATAC [27] is a pipeline for analysing scATAC-seq data. Before peak calling, it constructs a cell by bin matrix, converts it into a Jaccard index matrix by measuring similarities between cells and adjusts for differences in library size using a regression-based normalisation method. The normalised matrix is used for clustering and peak calling is performed on each cluster to identify *cis*-regulatory elements.

cisTopic [17] uses Latent Dirichlet Allocation (LDA), which is a Bayesian topic modelling approach commonly used in natural language processing, with collapsed Gibbs sampler to classify chromatin regions into regulatory topics and classify and cluster cells based on their contributions to regulatory topics. Clustering of cells is achieved

through optimising topic-cell distribution and classifying regions into topics is through region-topic distribution.

2.7 Benchmarking scATAC-seq analysis tools

2.7.1 Previous benchmarking efforts

Currently, the interest in exploring the potential of scATAC-Seq booming than ever before. Furthermore, technologies including the commercially available ones such as 10X-ATAC make scATAC-Seq technologies readily available for even small laboratories. Therefore, a comprehensive assessment of existing analysis tools of scATAC-seq data and the development of pipelines that can suit the wide range of sequencing techniques will be critical. The only available benchmarking effort to-date is by Chen *et al.* (2019) [10].

Chen *et al.* (2019) [10] benchmarked the performance of 10 computational methods for scATAC-seq analysis on different data sets in terms of feature matrix generation and clustering results. They concluded that SnapATAC [27] performs consistently well across data sets with varying numbers of cells (from 2,034 cells to 80,000 cells) while CisTopic [17] and methods developed by Cusanovich *et al.* (2018) [7] performed well in comparatively smaller data sets (fewer than 6,000 cells).

Chen *et al.* (2019) [10] also assessed the impact of keeping or removing the first principal component (PC) had when performing PCA dimension reduction. They presumed that the first PC only captures variation in sequencing depth. They found that methods that do not specifically address sequencing depth saw improved clustering results upon removal of the first PC. On the other hand, methods that implement binarisation (e.g. Cusanovich *et al.*(2018) and SnapATAC) or that implement cell coverage bias correction (e.g. chromVAR and SnapATAC), tend to be less affected by sequencing depth, hence the removal of the first PC did not have a major effect on clustering. Secondly, for methods using peaks as input, there was no clear difference between using bulk ATAC-seq peaks and peaks obtained from aggregated single cells. Only cisTopic, Cusanovich *et al.*(2018), and Cicero performed better did aggregating cell peaks perform better. Thirdly, after peak calling, cisTopic, Scasat, SCRAT, and SnapATAC filters out the

ENCODE blacklist regions. The authors conclude that including this step does not have a major benefit on performance. Finally, for rare cell types, calling peaks using the pseudo-bulk of all cells may miss peaks specific for these rare cells. Without correction by pre-labelling these rare cells may not be identified [10].

2.7.2 Benchmark evaluation metrics

Chen *et al.* (2019) [10] used three commonly used metrics to evaluate the clustering results: a) the Adjusted Rand Index (ARI), b) Adjusted Mutual Information (AMI) and cluster homogeneity when ground truth (i.e. known cell group labels for the data, e.g. FACS labels) was available c) Residual Average Gini Index (RAGI) when ground truth was not available.

The ARI score is an adjusted version of Rand Index (RI) which measures similarity between two clusters. Mutual Information (MI) measures the mutual dependence between two clusters. Homogeneity measures whether the clustering algorithm assigns cells of the same class to each cluster. The Gini Index (GI) of a marker gene measures the imbalance in gene accessibility across clusters. GI is between 0 and 1, with 1 meaning imbalance i.e. a gene is accessible in only one cluster and 0 meaning a gene is accessible to an equal extent in both clusters. A positive RAGI value indicates that the marker gene separates the clusters better than a housekeeping gene.

2.7.3 Benchmarking platform

Recently, colleagues at WEHI developed CellBench [68], an R package containing functions and data structures that simplify the benchmarking of combinations of analysis methods without duplicating code. When using CellBench, methods are modified by wrapper functions to allow different methods taking a common input format and producing a common output format. This simplifies the process by generating many combinations from lists of methods in different analysis stages and allows individual combinations to fail without affecting the execution of other method combinations. It also contains a function to time methods which can be useful for comparisons of running time.

2.8 Other data sets

2.8.0.1 Publicly available scATAC-seq data sets

Previously mentioned publicly available data sets could potentially be used for benchmarking, however, the ground truth for cell types are not clearly determined using other methods. For example, Chen *et al.* used the clusters determined by CellRanger ATAC as ground truth. The continuity of the Human hematopoietic lineage data set can be used for benchmarking trajectory analysis methods and the Mouse Cell Atlas data set can be used to benchmark the efficiency and scalability of different analysis methods.

2.8.0.2 Simulated Data: generating single cell ATAC-seq data from bulk ATAC-seq data

To simulate single-cell ATAC-seq data from bulk data sets, previous studies [10] have used the peak $\times$ cell type count matrix from bulk ATAC-seq and a binomial model. In detail, for a simulated single cell j having cell type k and peak i, the peak counts for this single cell is generated by $c_{i,j} \sim binom(2, p_i^t)$ where $p_i^t = (r_i^t)(\frac{1}{2}n)(1-q) + (\frac{1}{k})(\frac{1}{2}n)q$, with $q \in [0, 1]$ defining the noise level and n defining the number of simulated fragments. Some potential sources of publicly available bulk ATAC-seq data sets include: (1) human bone marrow bulk ATAC-seq (GSE119453), which contains FACS-sorted data from 6 cell types (HSCs, CMPs, erythroid cells (Ery), and three lymphoid cell types: NK, CD4 and CD8 T-cells) and (2) human erythropoiesis bulk ATAC-seq (GSE115672), which includes HSCs, CMPs, MEPs, MPPs, myeloid progenitors (MyP), colony forming unit-erythroid (CFU-E), proerythroblasts (ProE1), proerythroblasts (ProE2), basophilic erythroblasts (BasoE), polychromatic erytrhoblasts (PolyE), orthochromatic erythroblasts (OrthoE) and OrthoE and reticulocytes (Orth/Ret).

	Method/Tool	Year	Platform	10X	Input	Pre-prossessing	Matrix Generation	Clustering	Differential Accessibiliy	TF-motif	Gene Activity	Co-accessibility	Trajectory	Pathway	Enrichment	Documentation
2	APEC	2019	Python	Y	fastq, matrix + peak		Peak	Louvain, KNN								Detailed
3	ArchR	2020	R	Y	BAM, fragments		Bin	LSI				Cicero				Very Detailed
4	BROCKMAN	2018	R	N	fastq		k-mer									Brief
5	Cell Ranger ATAC	2018	Command line	Y	fastq		Peak	k-means								Very Detailed
6	ChromSCape	2019	R Shiny	N	Various		Bin, Peak, TSS	Hierachical								Detailed
7	chromVAR	2017	R	N	bam+peak		Motif, k-mer									Detailed
8	Cicero	2018	R	Y	sparse matrix		TSS	Louvain								Very Detailed
9	cisTopic_v3	2019	R	Y	bam+regions, matrix		Peak	Hierachical								Detailed
10	Cusanovich 2018	2015	Scripts	N	bam		Peak									Detailed
11	Destin	2019	R	Y	fastq,bam+peak		Peak	k-means								Detailed
12	epiConv	2020	R source	Y	matrix		Peak	Louvain								Detailed
13	Garnett	2019	R	N	CellDataSet			Classify cells								Very Detailed
14	Gene scoring	2019	Scripts	N	fastq		TSS									Detailed
15	scABC	2018	R	N	fastq		Peak	k-means +								Brief
16	SCALE	2019	Python	Y	matrix		Peak	Various								Detailed
17	scasat	2018	JupyterNotebook	N	list bams		Peak	k-medoids								Brief
18	scATAC-pro	2019	Command line	Y	fastq, various		Peak	Louvain				Cicero				Very Detailed
19	SCATE	2020	R	N	list bams		Peak	k-means + hierachical								Brief
20	SCRAT	2017	R,WEB	N	list bams		Various	Various								Detailed
21	Signac	2019	R	Y	matrix		Peak	Hierachical								Very Detailed
22	SnapATAC	2019	R	Y	fastq		Peak	Lovain-Jaccard								Very Detailed
23	STREAM	2019	Python/Docker	Y	various		Various	k-means								Detailed

FIGURE 2.5: **Computational tools for scATAC-seq data analysis.** 22 computational tools specific for scATAC-seq analysis are summarised in terms of publication year, available platform, adapted for 10X output, input data type, availability of various downstream analysis steps and documentation quality. The resulting feature matrix type, clustering algorithms are also stated.

Chapter 3

Methodology

3.1 Data set overview

In this project, an in-house scATAC-seq data set generated using the 10x Chromium platform was used for benchmarking. It contains an equal mixture of nuclei from five different human lung adenocarcinoma cell lines (i.e. HCC827, H1975, A549, H838 and H2228) (Figure 3.1) and contained a total of 1,500 nuclei. This data set was generated by Luyi Tian.

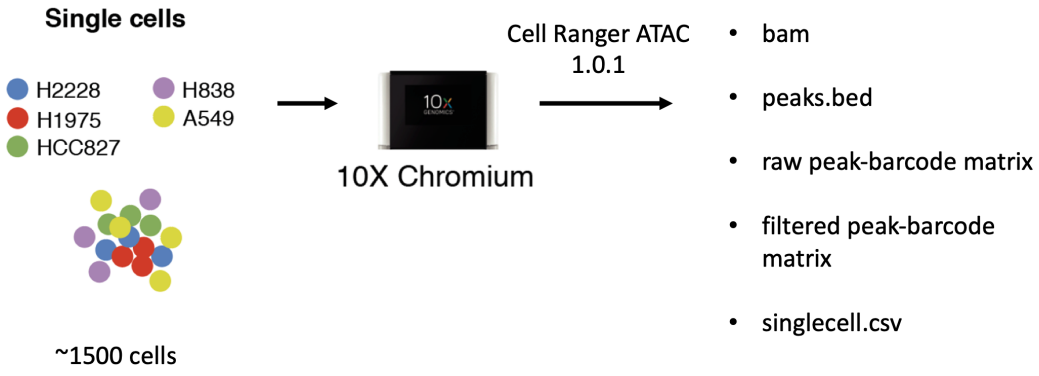


FIGURE 3.1: **Summary of the in-house scATAC-seq data used in the benchmark analysis.** Figure adapted from Tian *et al.* [69]. Five human lung adenocarcinoma cell lines were mixed in equal proportions ($\sim$ 1,500 cells). 10x scATAC-seq was performed on these cells as described in the Section 3.1.1 and the preliminary analysis using Cell Ranger ATAC (version 1.0.1) generated many outputs with five of them: bam file, raw peak-barcode matrix, filtered peak-barcode matrix, peaks.bed and singlecell.csv, used in the benchmarking for this project.

3.1.1 Cell culture and library preparation

The cell lines were obtained from the ATCC (https://www.atcc.org/) and the cell culture. Cells were cultured in Roswell Park Memorial Institute (RPMI) 1640 medium with 10% fetal calf serum and 1% penicillin-streptomycin. The cells were grown independently at 37 degree Celsius with 5% carbon dioxide until near 100% confluency.

According to the Nuclei Isolation for Single Cell ATAC sequencing (10x Genomics) protocol, cell nuclei were isolated and washed to prepare for library generation. Sequencing libraries were generated according to the Chromium Single Cell ATAC Reagent Kits User Guide (10x Genomics; CG000168 Rev B) [70], which were then loaded on an Illumina NextSeq500 sequencer with 2 x 75 paired-end kits using the following read length: 72 bp read 1N (read 1), 8 bp i7 index (gray region), 16 bp i5 index (green region) and 72 bp read 2N (read2) as illustrated in the top panel of Figure 2.3. Only reads 1N and 2N contain the insert sequences, while the index reads, i5 and i7, capture cell barcodes and sample indices, respectively. The sequencing run yielded around 300 million read pairs in total.

3.1.2 Basic quality control (QC) using Cell Ranger ATAC

A preliminary analysis (by Luyi Tian) was carried out using the 10x Genomics recommended Cell Ranger ATAC (version 1.0.1) workflow that included barcode processing, aligning reads to the human genome (GRCh38), marking duplicates, peak calling, cell calling, quality control, generation of a peak by cell barcode matrix, dimensionality reduction, clustering and t-SNE projection.

This analysis identified 7 clusters from 1,320 cells. The cluster number obtained was greater than expected (five clusters were expected since the design involved mixing cells from 5 distinct cell lines) which may suggest epigenetic heterogeneity within some cell lines, and more detailed analysis may be needed.

Cell Ranger ATAC output that were used in the downstream analysis include the following files (Figure 3.1):

- `bam` file: contains aligned reads to the human genome (GRCh38)

- `peaks.bed` file: contains peaks identified from aggregated single cell data
- `raw peak-barcode matrix`: contains the counts of fragment ends (or cut sites) within each peak region for each barcode
- `filtered peak-barcode matrix`: contains peak-barcode matrix of only cell barcodes
- `singlecell.csv`: contains the metadata of mapping and cell calling results for each barcode

In this project, the `bam` file was used for simulating pseudo-cells that contain a specific proportion of reads from different single cells. The raw peak-barcode matrix was filtered using different stringencies and the resulting matrices were used as input for the benchmarking. The `singlecell.csv` file is used for cell calling by some tools.

3.1.3 Ground truth

As the input cell line identity is known in our data set, it could be used as ground truth for our benchmarking study. The cell type of the barcodes were determined using genetic variation amongst the cells through Demuxlet (version 0.0.1) [71] with the `bam` file as input.

3.2 Benchmarking the clustering step of scATAC-seq tools

3.2.1 Selected scATAC-seq clustering tools

Clustering in single-cells is used to identify cell types and cell states. It is also the prerequisite for trajectory analysis. Clustering may include multiple sub-steps such as filtering of peaks and barcodes, normalisation and dimension reduction. This thesis includes benchmarking results for five R based scATAC-Seq specific clustering tools: Cicero (version 1.4.4) [16], Destin (version 1.0.1) [18] , epiConv [19], scABC (version 0.99.0) [21] and Signac (version 0.2.5) [72].

These tools implemented different types of clustering algorithms, normalisation methods and dimensionality reduction methods as summarised in Table 3.1. Cicero, Destin and

epiConv use the binarised peak-barcode matrix, i.e. if the number of fragments ends for a certain peak and barcode is above one, the count is treated as one. Signac and epiConv used the TF-IDF normalisation and Cicero provides either log or size only normalisation before dimensionality reduction. scABC and epiConv do not determine the optimal number of clusters from the data, instead of requiring the user to provide this information.

TABLE 3.1: **Summary of selected scATAC-seq clustering tools.**

Tool	**Binary**	**Normalisation**	**Dimension Reduction**	**Clustering Algorithm**	**Parameters**	**K**
Cicero	Y	log or size_only or none	PCA or LSI or UMAP or tSNE	Louvain or Leiden	k in kNN graph	N
Destin	Y	-	PCA or tSNE	k-means + Elbow	k range	N
epiConv	Y	TF-IDF	UMAP	densityClust	k	Y
Signac	N	TF-IDF	LSI	Louvain or Leiden or Multilevel refinement Louvain or SLM	resolution	N
scABC	N	-	-	k-medoids + correlation analysis	k	Y

3.2.2 Input data manipulation

To benchmark the five selected clustering tools, a peak-by-cell matrix was used as the input for all tools. The `raw peak-barcode matrix` generated by Cell Ranger ATAC contains the fragment counts for all detectable barcodes (both cell and non-cell barcodes). The raw matrix file in `hdf5` format together with `peak.bed` are converted to the `SingeCellExperiment` class. The `singlecell.csv` file has different summary statistics as columns and all detectable barcodes as rows. The 16 summary statistics for each barcode include total read-pairs, number of duplicate read-pairs, number of chimerically mapped read-pairs, number of read-pairs with at least one end not mapped, number of read-pairs with < 30 MAPQ on at least one end, number of read-pairs mapping to mitochondria and non-nuclear contigs, number of non-duplicate, usable read-pairs i.e. "fragments", a binary indicator of whether the barcode is associated with a cell, number of fragments overlapping with TSS regions, number of fragments overlapping with DNase sensitive regions, number of fragments overlapping enhancer regions, number of fragments overlapping promoter regions, number of fragments overlapping any of TSS, enhancer, promoter and DNase hypersensitivity sites, number of fragments overlapping blacklisted regions, number of fragments overlapping peaks, number of ends of fragments in peak regions. This information together with the Cell Ranger ATAC assigned cluster

number were added to the `SingeCellExperiment` class as part of `ColData` describing the barcodes.

Next, the barcodes were selected using two filtering methods to obtain five data sub-sets. The first step kept only the barcodes that can be labelled by demuxlet using the genotypic information as mentioned in Section 3.1.3. Next, two filters were applied to generate five data sub-sets to achieve data of varying noise levels, some of which will contain barcodes that are may not be real cells. The first filter (Filter 1) was the number of unique fragments, and only the barcodes with $> 5,000$ unique fragments were kept. This data set was labelled as `Set-1`. The second filter (Filter 2) was based on the binary library size, obtained by first binarising the peak-cell matrix and calculating the column sum of the resulting library size. This filter was equivalent to counting the number of unique peaks for each barcode. The thresholds set were above 1000, 100, 10 and 0 to create increasingly noisy sub-sets of the data, which were labelled as Set-2-1, Set-2-2, Set-2-3 and Set-2-4 respectively.

3.2.3 Evaluation metrics

To evaluate the clustering results, the Adjusted Rand Index (ARI) [73], Adjusted Mutual Information (AMI), Normalised Mutual Information (NMI), homogeneity, completeness and v-measure were used to measure the similarity between two clusters and assess whether each cluster contains only cells belonging to a single class (cell type/line). ARI was calculated using the `adjustedRandIndex` function from mclust (5.4.6) R package [62]. NMI and AMI were calculated using the aricode (1.0.0) R package. Homogeneity, completeness and v-measure were calculated using the `vmeasure` function from the sabre (0.3.2) R package [74].

3.2.3.1 ARI

The Rand Index measures the similarity between two partitions by considering all pairs of samples. It is the ratio of the number of pairs assigned in the same or different clusters in either partition against the total number of pairs. The Adjusted Rand Index is ensured to have a value close to 0 for random labelling and close to 1 when the partitions are very similar.

The Rand Index (RI) is defined as:

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{3.1}$$

for a set S with two partitions X and Y (e.g. cell type labels and clusters defined by a given method),

- a is the number of pairs of elements in S that are in the same subset in X and in the same subset in Y
- b is the number of pairs of elements in S that are in different subsets in X and in different subsets in Y
- c is the number of pairs of elements in S that are in the same subset in X and in different subsets in Y
- d is the number of pairs of elements in S that are in different subsets in X and in the same subset in Y

The denominator is the total number of pairs, which can also given by $\binom{n}{2}$.

The ARI is corrected-for-chance version of the RI:

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)} \tag{3.2}$$

and for a contingency table of number of items in X and Y partitions, ARI is defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{3.3}$$

where n_{ij} is the numbers in the contingency table and a_i is the row summation and b_j is the column summation.

3.2.3.2 AMI

The Mutual Information measures the similarity between two labels of the same data. However, MI will be larger for a larger number of clusters, regardless of whether there

is more information shared. Adjusted Mutual Information (AMI) is an adjustment of the MI to account foe chance. Considering two clusters U and V with $|U|$ and $|V|$ being the number of samples in the clusters, the mutual information is given by:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|}\frac{|U_i \cap V_j|}{N} log\frac{N|U_i \cap V_j|}{|U_i||V_j|} \tag{3.4}$$

The AMI is given by:

$$AMI(U,V) = \frac{MI(U,V) - E(MI(U,V))}{avg(H(U),H(V)) - E(MI(U,V))} \tag{3.5}$$

where $H(.)$ represents the entropy function which is given by:

$$H(U) = -\sum_{i=1}^{R} P_U(i)\log P_U(i) \tag{3.6}$$

3.2.3.3 NMI

Normalised Mutual Information is a normalised form of NMI to keep the range between 0 and 1 but does not adjust for the chance. NMI is given by:

$$NMI(U,V) = \frac{MI(U,V)}{max(H(U),H(V))} \tag{3.7}$$

3.2.3.4 Homogeneity, completeness and v-measure

Homogeneity, completeness and v-measure are three important metrics for evaluation of the clustering method. The higher homogeneity means that the predicted clusters contain mainly samples from one true class. Completeness considers the other way that is the higher completeness means more samples from one class are predicted as the same cluster. V-measure is the combination of these two metrics.

The homogeneity score is defined as:

$$h = 1 - \frac{H(Y_{true}|Y_{pred})}{H(Y_{true})} \tag{3.8}$$

The completeness score is defined as:

$$c = 1 - \frac{H(Y_{pred}|Y_{true})}{H(Y_{pred})} \tag{3.9}$$

where $H(.)$ represents the entropy function.

The v-measure score is give by :

$$v = \frac{(1+\beta) \times homogeneity \times completeness}{\beta \times homogeneity + completeness} \tag{3.10}$$

, where the β was set to 1.

All of these metrics are bounded between 0 and 1, with 1 indicating the best results and 0 indicating the worst.

3.2.4 Benchmarking with CellBench

The Bioconductor package CellBench [68] was used in the benchmarking analysis to help streamline the processing and organise the results. The package simplifies the analysis by providing user-friendly functions to reduce the amount of duplicating code required. In this project, the CellBench mainly took several inputs data sets, and the wrapper functions for the clustering tools were written and multiple evaluation metrics were applied to the predicted and true clusters. If any error happened during the analysis, the function was still able to run and carried the error message. The time consumption was also determined using CellBench.

The wrapper functions for selected clustering tools (Section 3.2.1) were written with some tools using multiple parameters. The parameters for Cicero [16] were set to four combinations. The normalisation method was set to "size" or "Log" and the dimension reduction method was set to PCA or LSI. Many of the tools have their filters for barcodes and peaks regions. The tool-specific filters were not applied or a dummy parameter was

provided. For scABC and epiConv, the cluster number expected in the data set (5) was provided as required.

3.2.5 Clustering cells using selected scRNA-seq clustering tools

Three clustering methods from the scRNA-seq tools were selected and applied to the scATAC-seq data. SC3 [75], RaceID [76] and Seurat [77] were selected [78] as they were determined to perform well on scRNA-seq clustering analysis [69, 79] and they were all R based that is comparable to the scATAC-seq clustering tools benchmarked. The Set-1 was used as input for the three tools and the ARI of the clustering results were calculated using the `adjustedRandIndex` function from the mclust R package [62].

3.3 Simulation of pseudo-cells

To increase the complexity of input data sets, simulation with different noise levels or sequencing depths is often applied. For example, previous studies [10] have used the peak × cell type count matrix from bulk ATAC-seq and a binomial model to simulate single cell matrix data as mentioned in Section 2.8.0.2. This approach simulates data at the matrix level. In this project, we considered simulating cells using the sequencing data by mixing reads from different cells in different proportions to create 'pseudo-cells'.

Three cell lines were selected for simulation: A549, HCC827 and H2228. There are 130 A549 cells, 142 HCC827 cells and 91 H2228 cells. The original `bam` file generated from Cell Ranger ATAC was first split into multiple `bam` files, with each `bam` file representing one barcode. Only the cells (barcodes) belonging to these three cell lines were selected from the `bam` files. Next, for each pseudo-cell, a `bam` file from one of each of the three cell types was randomly selected. From these three `bam` files, a corresponding proportion of reads were randomly sampled using Samtools [57] and all sampled reads were merged into a new `bam` that represented the pseudo-cell. As illustrated in Figure 3.2, Mix A, B and C were designed to have 68% of reads from the major cell type and 13% from the two other minor cell types, with Mix A dominated by A549, Mix B dominated by H2228 and Mix C dominated by HCC827. Mix M was designed to have an even number (33% of reads) from each cell type to mimick an equal mixture of DNA from three types of single cells.

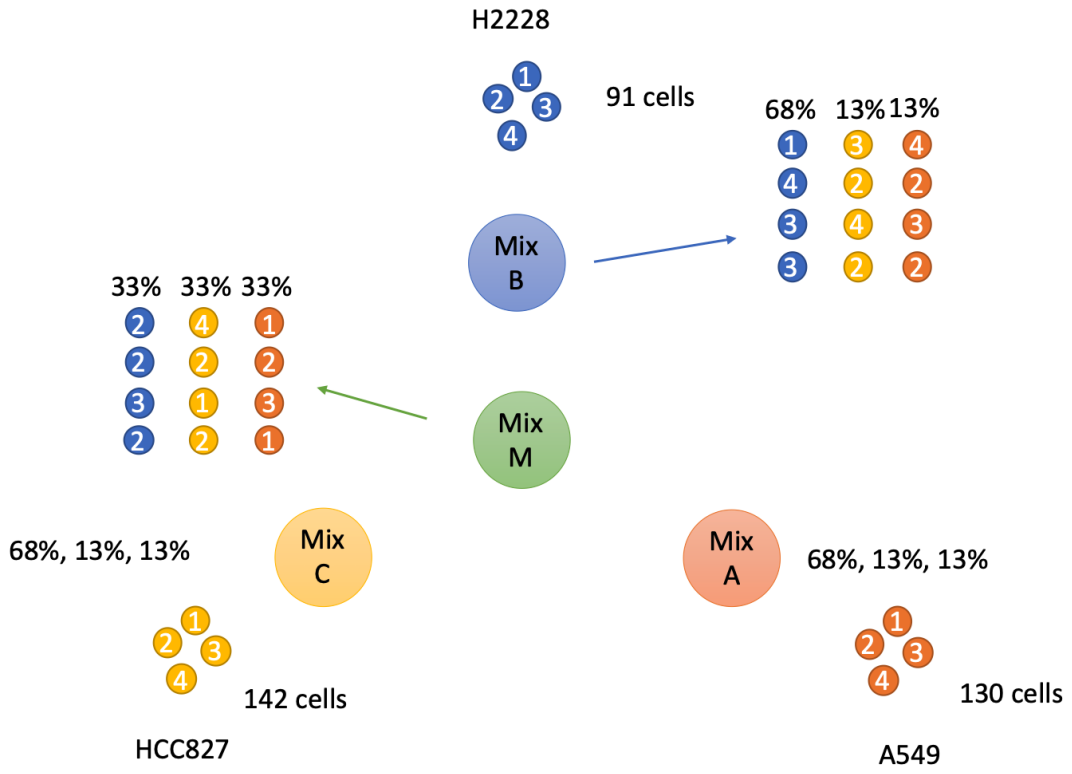


FIGURE 3.2: **Illustration of simulation of reads to create pseudo-cells.** Singe cell data from A549, HCC827 and H2228 cell lines were selected, which are represented as circles with numbers. One cell was selected from each of the three types. Mix A, B and C were designed to have 68% of reads from the single cell of one cell line and 13% reads each from other two single cells. In Mix A, the dominant cell type was A549. In Mix B, the dominant cell type was H2228. In Mix C, the dominant cell type was HCC827. In Mix M, 33% of reads from each of the three single-cells were merged to form the simulated pseudo-cell.

The simulated single-cell `bam` files were analysed using cisTopic version 3 [61] and the Signac [26]. The peak used was generated from the whole data set mentioned in 3.1 using MACS2 from Cell Ranger ATAC. The method used to generate the feature matrix was `WarpLDA`, with the optimum number of topics being 50. UMAP was used to show the structure of the simulated data set in two dimensions.

Chapter 4

Results

4.1 Usability of available scATAC-seq tools

As summarised in Figure 2.5, several scATAC-seq specific tool-kits or pipelines have extremely detailed documentation, which include ArchR [13], Cell Ranger ATAC, Cicero [16], Garnett [20], scATAC-pro [11], Signac [26] and SnapATAC [27]. The documentation not only provides multiple vignettes to help users understand the workflow but also explains the reasons for selecting certain parameters or explanation of algorithms. They covered nearly all the important steps for scATAC-seq analysis. Some other tools also have detailed documentation, which also greatly helps users to understand the tool/-package quickly and perform the correct analysis.

4.2 Quality control of the data set

4.2.1 Basic quality control

The basic quality control (QC) plots and summary statistics generated by Cell Ranger ATAC or Samtools [57] indicated that our in-house generated data was of high quality.

The pre-processing and basic quality control (QC) of the scATAC-seq data were firstly performed using Cell Ranger ATAC (1.0.1). The data was estimated to contain 1,320 cells with the median fragments per cell being 6,820 and the total read pairs number was around 288 million. As shown in Figure 4.1, the insert size distribution plot showed a

periodicity of 150 bp corresponding to the number of nucleosomes that the transposase accessible fragments could span, including nucleosome free (i.e. < 150 bp), mononucleosome (i.e.150-300 bp), and dinucleosome (i.e. > 300 bp) fragments. It also captured the sawtooth pattern in the < 200 bp region, which corresponded to the helical pitch of DNA. The TSS enrichment plot showed the cutting sites had large enrichment around the TSS within the 2,000 bp range of annotated TSS regions. TSS regions were known to have a high degree of chromatin accessibility, therefore the large enrichment indicated that the data set was of high quality overall.

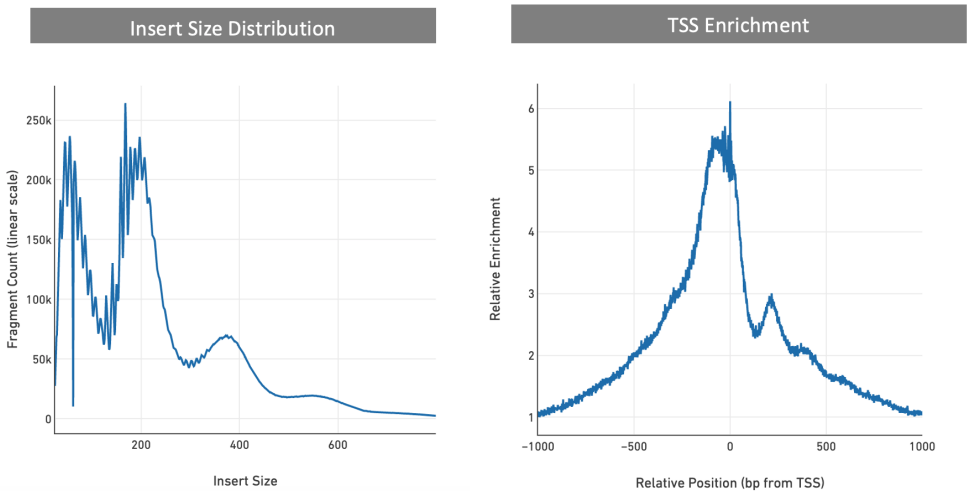


FIGURE 4.1: **QC of the in-house scATAC-seq data using Cell Ranger ATAC.** The insert size distribution plot on the left captured periodicity and sawtooth structure of the transposase accessible fragments distribution. The TSS enrichment plot on the right showed that the transposase cutting sites were enriched around the TSS regions.

After using Samtools [57] to obtain summary statistics, it was found that the `bam` file included duplicates (marked with the sam flag 1024), mitochondrial alignments and chimera alignments etc. There were 369,240 unique barcodes in the `bam` file, with 597,146,352 reads in total. 96.23% of the reads were mapped, 74.57% of the reads were duplicates and 94.79% of the reads were properly mapped. Looking at the distribution of read number across all barcodes, we found 13,833 barcodes with more than 1,000 reads and 1,463 barcodes with more than 10,000 reads, which is shown in Figure 4.2. For the 1,463 barcodes with hign numbers of reads, the average number of reads per barcode was 365,471, which represented high read depths for these cells. The `raw peak-barcode matrix` contained the counts of fragment ends within each peak region for each barcode and had 112,754 peaks and 197,479 barcodes. These barcodes were

generated by retaining reads that were pair-aligned with both reads having mapping quality (MAPQ)> 30, non-mitochondrial and non-chimerically mapped. Any 'non-cell' barcodes determined by the Cell Ranger ATAC (1.0.1) were then filtered out and the `filtered peak-barcode matrix` contains information from 1,320 'true' barcodes (i.e. cells).

After cell calling, Cell Ranger ATAC (1.0.1) determined 1,320 cells by fitting the mixture model of two negative binomial distributions to the number of fragments overlapping peaks (Figure 4.3 Cell Calling Plot). Besides, most of the determined cells had relatively large number of fragments per barcode and a high percentage of fragments overlapping peaks as shown in the single cell targeting plot of Figure 4.3 . A total of 1,283 barcodes were assigned a cell type label using the genotypic variation present in the five cell lines used in the experiment via the demuxlet tool [71].

4.2.2 Barcode selection

The input for the benchmarking study was the `raw peak-barcode matrix`. Before clustering, barcodes need to be selected to achieve a clean peak by barcode matrix for use in benchmarking. Because CellRanger ATAC determined 1,320 cell barcodes and demuxlet assigned 1,283 barcodes with a cell type label, we were interested in the degree of overlap between the two barcode lists. As shown in Figure 4.4a, 970 barcodes had both a demuxlet cell type label and a high quality cell barcode. As shown in Figure 4.4b, clusters 1 and 7 determined by CellRanger ATAC were shown to be H1975 cells and cluster 3 was not assigned a cell type label. Furthermore, for each cell type the reads count exhibited a bi-modal distribution with means that were relatively close between the cell types. However, the distribution was different for read counts of barcodes without a cell type label. Moreover, most of the H838 cells were not classified as genuine cell barcodes by Cell Ranger ATAC as they had low read counts. Therefore, we applied a more stringent filter to remove barcodes with relatively fewer reads to clean up the data set and improve data quality.

Firstly, a filter based on the number of high quality unique fragments (Filter 1) was applied to retain only high quality barcodes for analysis (Figure 4.5). High quality means that both reads of the fragment have MAPQ $>$ 30 and they are non-mitochondrial and non-chimerically mapped. Barcodes with less than or equal to 5,000 unique fragments

and without a cell type label were filtered out. The resulting number of barcodes remaining after applying this filter were 669 and the number of barcodes that remained for each cell type was summarised in Table 4.1.

TABLE 4.1: **Number of cells for each cell type after applying Filter 1**

Filter 1	**A549**	**H1975**	**H2228**	**H838**	**HCC827**	**Total**
Set-1 Unique Fragments >5,000	130	212	91	94	142	669

The second filter considered the binary library sizes, which is the binarised column sum of the peak-barcode matrix and also the binarised counts of fragment ends in the called peaks (Filter 2). As for Filter 1, the barcodes without cell type label were removed. Multiple filtering thresholds were considered because we were interested in the ability of the different clustering tools to cope as data quality varied. The threshold for filtering was set to binary library sizes of at least 1,000, 100 and 10 for each barcode. The selection of 1,000 was based on the distribution of the binary library size in each cell type, i.e. select only those barcodes from each group with relatively high binary library sizes. After applying this filter with multiple thresholds, the number of remaining barcodes for each cell type was summarised in Table 4.2. With a lower threshold, more barcodes with low library size were included, which allows testing of the ability of different clustering methods to cope in the presence of more background noise.

TABLE 4.2: **Number of cells for each cell type after applying different level of Filter 2**

Filter 2	**A549**	**H1975**	**H2228**	**H838**	**HCC827**	**Total**
Set-2-1 Binary Lib Size >1000	132	222	101	92	151	698
Set-2-2 Binary Lib Size >100	164	256	128	155	190	893
Set-2-3 Binary Lib Size >10	204	288	162	384	221	1259
Set-2-4 Cell Type Label only	207	290	164	397	225	1,283

4.3 Benchmarking scATAC-seq clustering tools

4.3.1 Comparing clustering methods with a clean data set

To benchmark the clustering step for the five selected tools (Section 3.2.1), the Set-1 data set with five known cell type labels as ground-truth and high counts of unique fragments per cell was used as input. CellBench [68] was used to allow combinations of multiple clustering methods and evaluation methods to be run on the same input data set with clear and easy to follow R code. For Cicero, four combinations of two normalisation methods (log or size only) and two dimension reduction methods (LSI or PCA) with the Leiden clustering method were assessed. The resolution parameter was determined automatically. For Destin, the required range for the true number of clusters was set to 2 to 20 and PCA was used for dimension reduction. For signac, the small local moving (SLM) algorithms for clustering was selected, with resolution set to 0.8. For scABC and epiConv the number of clusters (k) was provided as 5.

Figures 4.7 and 4.8 present the performance of each of the clustering methods. All methods achieved high ARI with the lowest at 0.7 for Destin (Figure 4.8a). The UMAP representations of the clustering results were shown in Figure 4.7. The top plot was coloured by the cell type label determined by demuxlet. Using this plotting method, the H1975 cells were shown as two populations but all other cell lines were shown as distinct clusters indicating that the UMAP representation was informative and correctly captured the cell type differences. The remaining plots were coloured by the clustering results with ARI values labelled on the right. From the figure, all methods captured the cell type differences by correctly clustering the cells. Destin achieved lower ARI (i.e. 0.7) as it identified more clusters by further dividing the H1975 and HCC827 cells , however, the overall differences were correctly captured. For the methods that are able to determine the optimal number of clusters, none determined 5 clusters. Signac and three of the Cicero methods determined six clusters (the log transformation and PCA option determined 7) while Destin determined eight clusters. As shown in Figure 4.8b, all methods achieved high homogeneity scores which were close to 1, which indicated that each of the cluster determined contained nearly one cell type. The AMI, NMI, completeness and v-measure all showed the same pattern as the ARI (data not shown),

which again confirmed that all the clustering tools performed well on this clean, well separated data set.

It was not surprising to see six clusters determined by many of the tools, as the tSNE plot of the input data set shows a spread of H1975 cells that could well be separated into two sub-clusters (Figure 4.9). The two sub-clusters may be due to experimental variation related to cell cycle or different mutations (sub-clones) present in this cancer cell line. When provided with the expected number of clusters, both scABC and epiconv correctly recapitulated the ground-truth labels. When this extra information was not provided, which is common in general practice, the performance of methods like Cicero and Signac are still reasonably good. Destin also performed well on this clean data set, but generated more clusters than expected.

4.3.1.1 Comparing scRNA-seq clustering methods

To understand whether clustering tools for scRNA-seq could perform well on scATAC-seq data, three recommended clustering methods [69] for scRNA-seq were selected and the clustering was performed using these tools on Set-1. The tools are RaceID [76], SC3 [75] and Seurat [49]. The UMAP representation of the clustering results were shown in Figure 4.10. For RaceID, the highest ARI (i.e. 0.99) was achieved when the cluster number was equal to five. For SC3, when the cluster number set to between four and seven clusters, the ARI was the highest, i.e. 0.97 for seven. For Seurat, using the default resolution parameter i.e. 0.8, the ARI achieved was 0.54, however, the results could be improved by reducing the resolution parameter. These results suggested that clustering tools for scRNA-seq data performed well on scATAC-seq data, achieving a high ARI value.

4.3.2 Comparing clustering methods with variable quality data

Next, the performance of clustering methods when the input data quality varied were compared. When the threshold of Filter 2 decreased from 1,000 to 0, more barcodes or cells with smaller library sizes were included in the data set and the quality decreased. As shown in Figure 4.11, the ARI decreased as the data quality decreased for all methods tested. Even methods that were supplied with the expected number of clusters (i.e. 5),

such as scABC and epiConv, were unable to recapitulate the cell type labels as well as they did for the clean data set. Performance generally suffered when cells with lower library sizes were included with the clean data set, i.e. the tools were not able to cluster the cells correctly.

Interestingly, as shown in Figure 4.12, Destin correctly identified five clusters in Set-2-1, while in Set-1 it over-clustered. However, Cicero with size only normalisation and PCA identified fewer clusters than expected in this Set-1 which lead to a reduction in ARI. Other methods generated similar clustering results between these two data sets. The two data sets contain a similar number of barcodes (669 vs 698) with 664 barcodes in common. This suggested that Destin and Cicero_size_PCA were unstable.

Based on the ARI (Figure 4.11, the performance of all tools decreased dramatically as the data changed from Set-2-2 to Set-2-3. By showing the UMAP plots for the clustering results on Set-2-3, we could see that the Cicero_LSI methods performed badly as they assigned multiple cell types to one cluster. Cicero_PCA, Destin, Signac and epiconv achieved low ARI at around 0.40, however, the differences of the distinct cell types were mostly captured. The low library size population reduced the ARI value a lot as the cluster population was hard to determine. scABC started to randomly assign the clusters but it achieved a relatively high ARI value. Some cells were not able to be assigned with a cluster label by scABC, suggesting the limited ability of scABC.

The results suggested that when performing the clustering step during scATAC-seq analysis, the filtering step can have a critical role in clustering performance, with most tools unable to cope with low quality barcodes even when the expected number of clusters is provided. This step also involves a lot of parameter tuning and inspection of the dimension reduced plot of the data (e.g. tSNE, PCA or UMAP) can help with this process.

4.3.3 Comparing the run-time of different clustering methods

Finally, the time spent to run selected clustering methods was compared. For the tested data set with a feature matrix of 12,754 peaks by 1,283 cells, the longest time for clustering is less than 3 minutes. Signac achieved the shortest median run time and the two LSI methods of Cicero achieved the longest median time (Figure 4.14). Our data

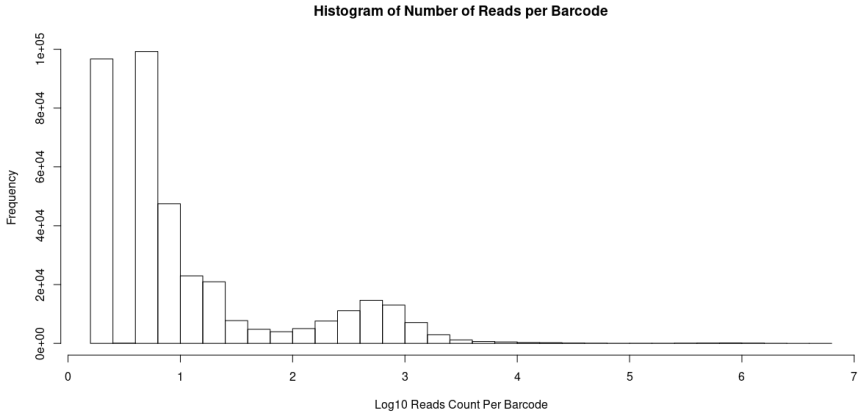
set is relatively small, and for scalability of the tools to be assessed more thoroughly, larger data sets will need to be used in the future.

4.4 Simulation of pseudo-cells containing reads mixed in varying proportions

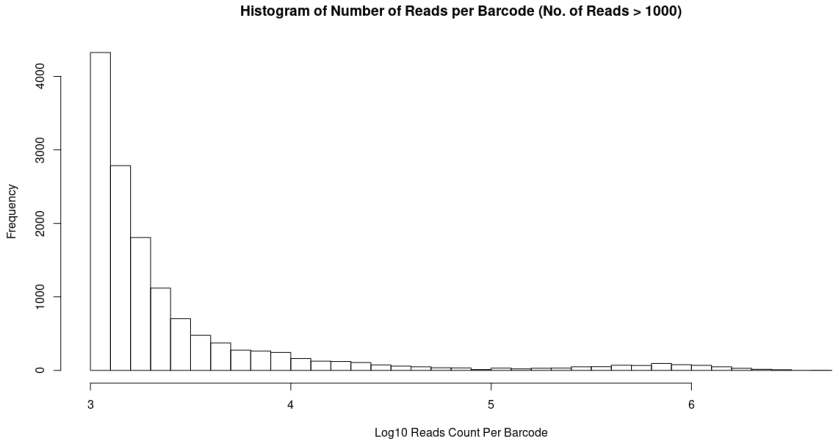
The designed in-house data set has simple structure of five distinct cell types, which could be too simple for the benchmarking study to observe differences between the clustering tools. All tools perform well on the carefully filtered clean data set (Set-1). One approach to increase the complexity of the data set is adding more intermediate cell types or more similar cell types make the clustering problem more difficult. As mentioned in Section 3.3, the simulation was designed to generate pseudo-cells that contained different proportions of reads mixed from single cells from three cell lines. The pseudo-cell was thought to be more similar to the cell type from which the cell had most proportion of DNA. However, as shown in Figure 4.15, the simulated pseudo-cells did not show a "gradient" of similarity to the three cell types, instead, the simulated cells were separated from A549 cells and randomly distributed between the HCC827 cells and H2228 cells without forming any distinct clusters as expected.

In summary, the simulation method did not achieve the designed structure, therefore, could not be used as input of the benchmarking study. One hypothesis to explain why the method did not work is due to the binary nature of scATAC-seq data. Each cell type was characterised by certain open chromatin regions. When the random sampling was performed, we were selecting a small sub-set of the open regions and then merging these regions from different cells to create new combinations of open regions. However, even though we were sampling from the same cell type, the random sample might only capture small subsets of open regions with large variation. To check this approach, we can apply it to scRNA-seq data which has matching lab mixed pseudo-cells to check that our simulations agree with what is obtained from physically mixing the RNA (or DNA in the case of scATAC-seq). For further simulation on the scATAC-seq data, other methods may need to be considered such as the recently published simATAC [80] or SCAN-ATAC Sim [81]. However, both methods simulate scATAC-seq data at a count

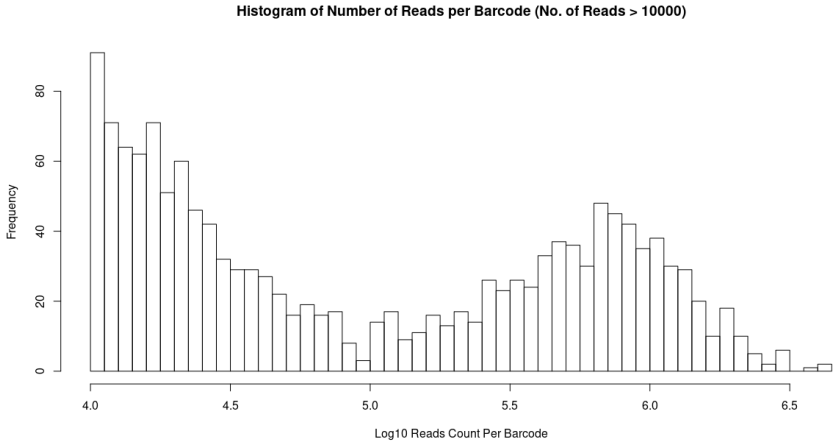
matrix level, therefore, methods to simulate scATAC-seq at the read-level still need to be explored.



(A) Distribution of read-counts in all barcodes.



(B) Distribution of read-counts in barcodes with more than 1,000 reads.



(C) Distribution of read-counts in barcodes with more than 10,000 reads.

FIGURE 4.2: **Distribution of read-counts of barcodes in `bam` file.**

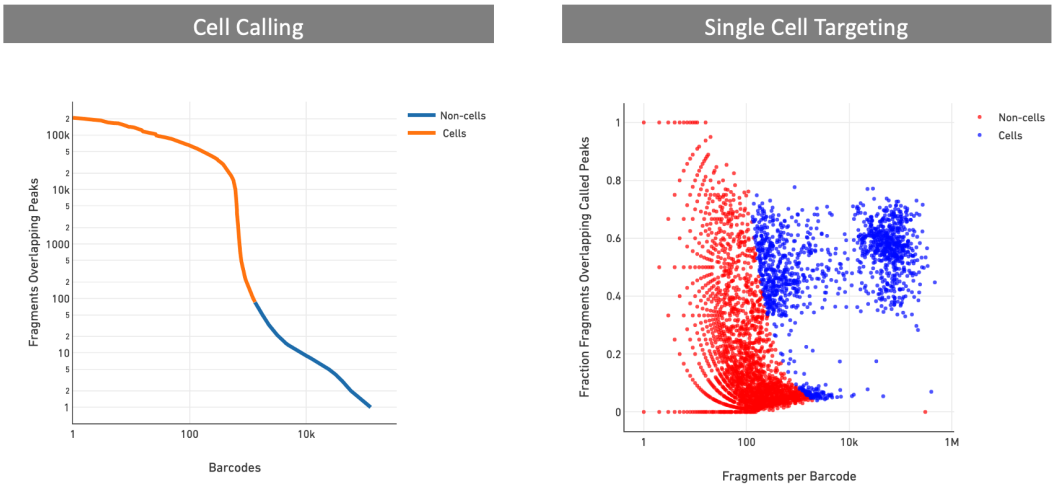
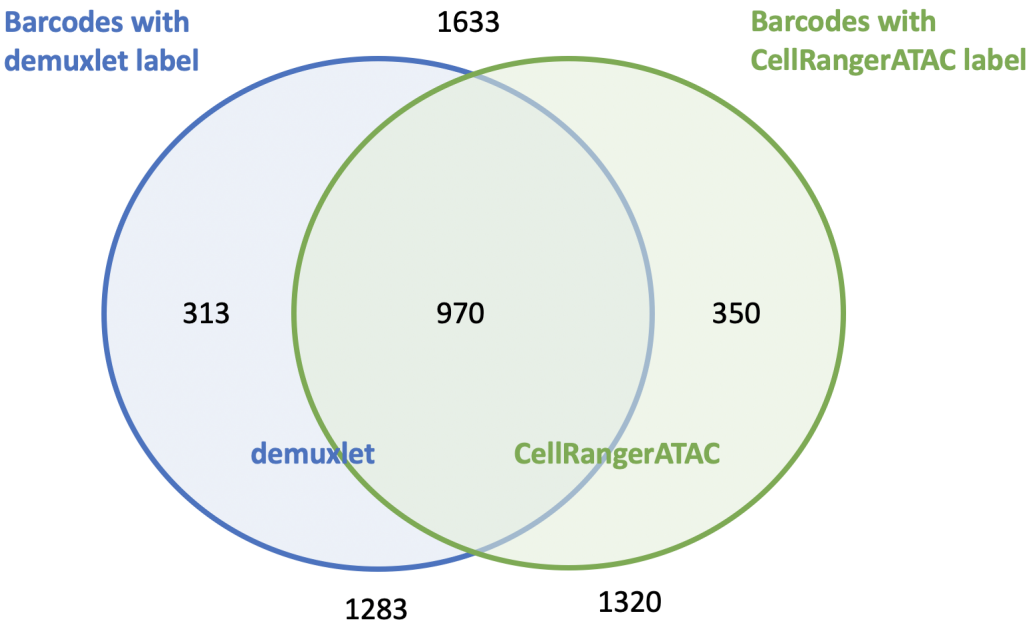
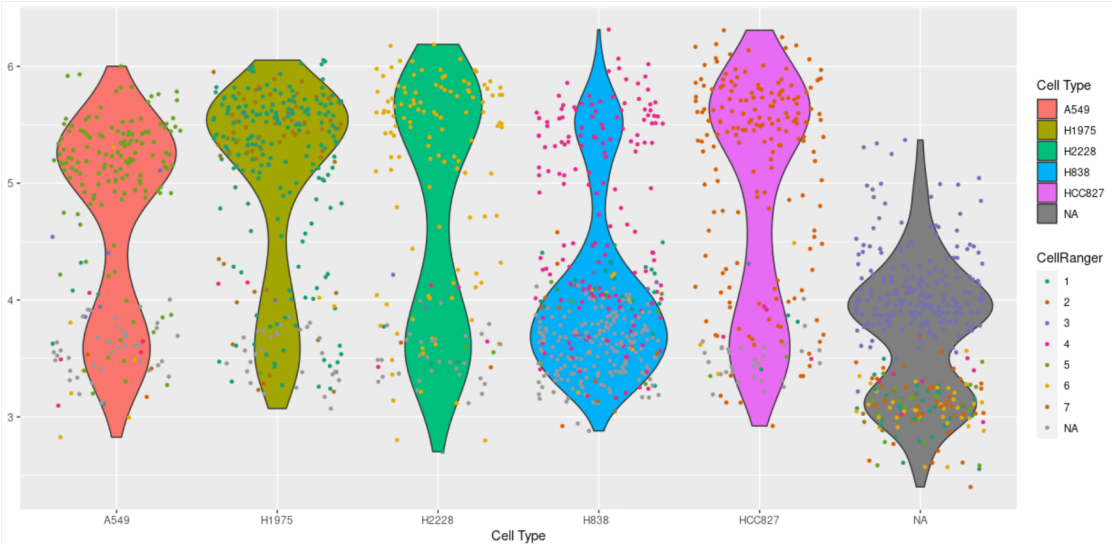


FIGURE 4.3: **Cell calling results using Cell Ranger ATAC.** Based on the number of fragments overlapping peaks by fitting a mixture model Cell Ranger ATAC identified 1,320 cells from the in-house data set (left plot). The single cell targeting plot on the right is a scatter plot of the fraction of fragments overlapping peaks versus the number of fragments per barcode. Most of the determined cells (coloured in blue) have a relatively large number of fragments per barcode and a high percentage of fragments overlapping peaks.



(A) Relationship between the number of CellRanger ATAC determined cell barcodes and the cells that demuxlet provided cell type (line) labels for.



(B) The distribution of $\log_{10}$(read-pair numbers) (y-axis) for each cell type (x-axis). Each violin plot is coloured according to cell types with grey indicating barcodes not labelled by demuxlet. Each dot represents a cell (i.e. barcode). The dots are coloured by CellRanger ATAC predicted clusters.

FIGURE 4.4: **Relationship between Cell Ranger ATAC determined cell barcodes and demuxlet labelled barcodes.**

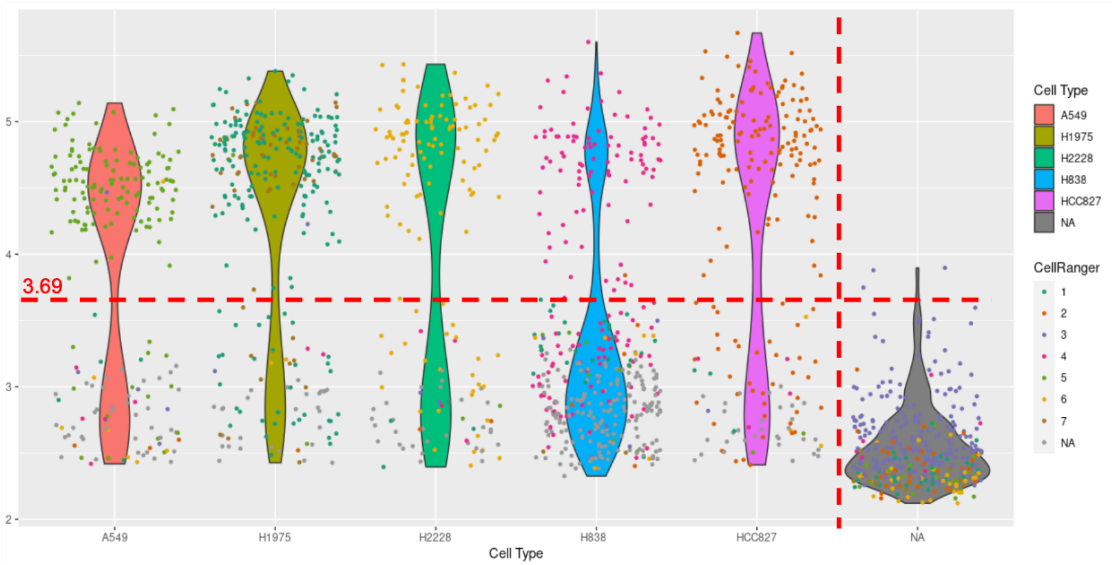


FIGURE 4.5: **Filter 1: filter of barcodes based on $\log_{10}$ of the number of high quality unique fragments per barcode for each cell type.** The plot shows the distribution of $\log_{10}$(read-pair numbers) in y-axis for each cell type (x-axis). Each violin plot is coloured according to cell types with grey indicating barcodes not labelled by demuxlet. Each dot represents a cell (i.e. barcode). The dots are coloured by CellRanger ATAC predicted clusters. The horizontal red line indicates the cutoff point for Filter 1 (i.e. barcodes with more than 5,000 unique fragments were retained). The vertical red line is to shows that barcodes without a cell type label through demuxlet were removed. Only the top-left section was retained.

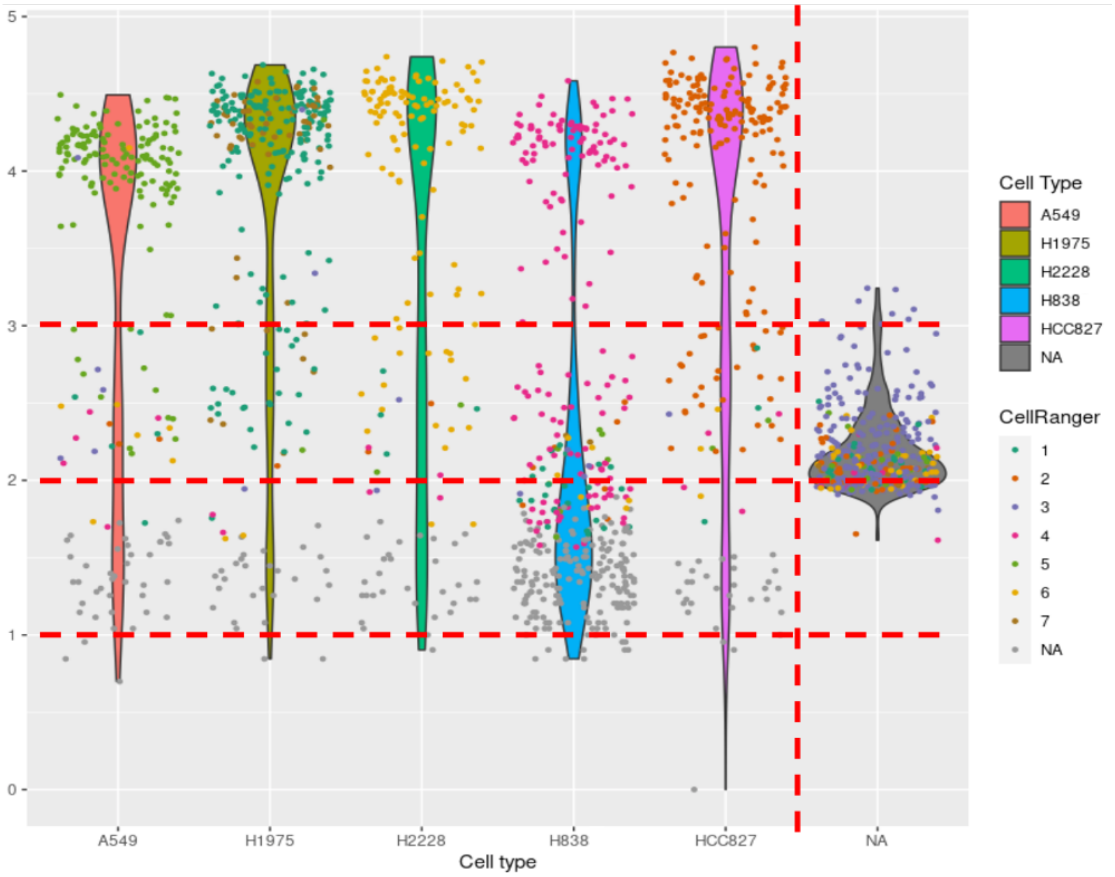


FIGURE 4.6: **Filter 2: filtering barcodes based on $\log_{10}$ of the binary library size.**
The plot shows the distribution of $\log_{10}$(read-pair numbers) in y-axis for each cell type (x-axis). Each violin plot is coloured according to cell types with grey indicating barcodes not labelled by demuxlet. Each dot represents a cell (i.e. barcode). The dots are coloured by Cell Ranger ATAC predicted clusters. The horizontal red lines indicates multiple threshold levels for filtering, representing binary library sizes of 1,000, 100, and 10 from top to bottom, respectively. The vertical red line is to show that barcodes without a cell type label through demuxlet were removed.

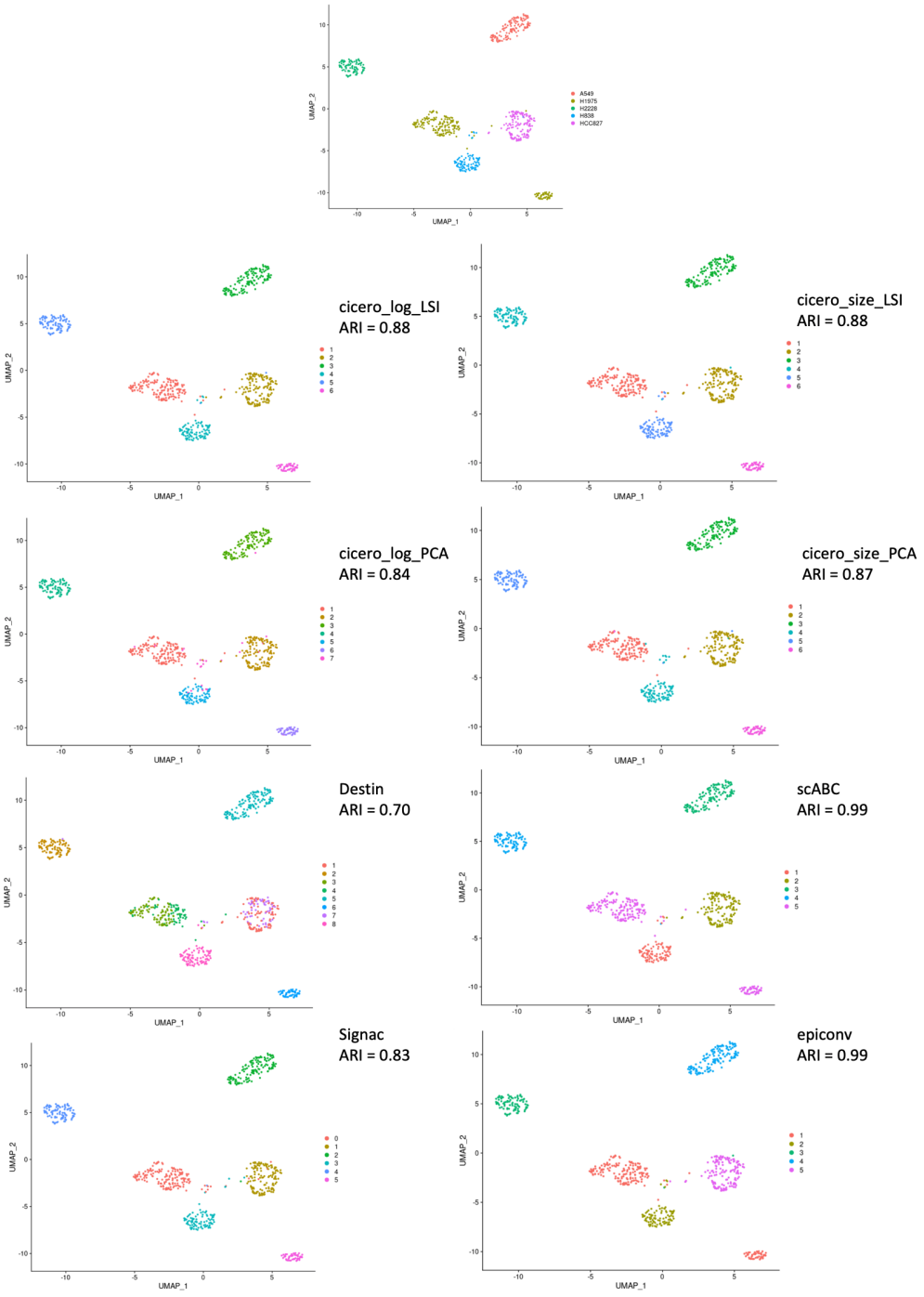
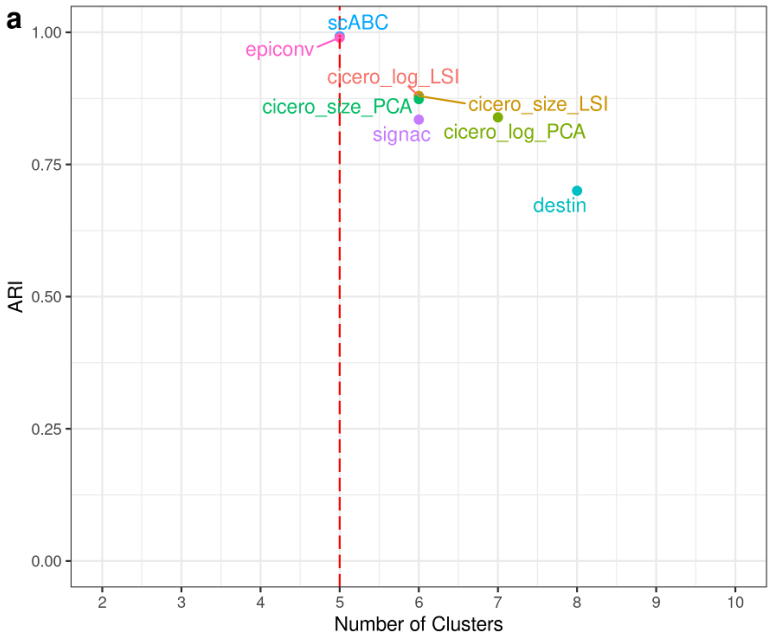
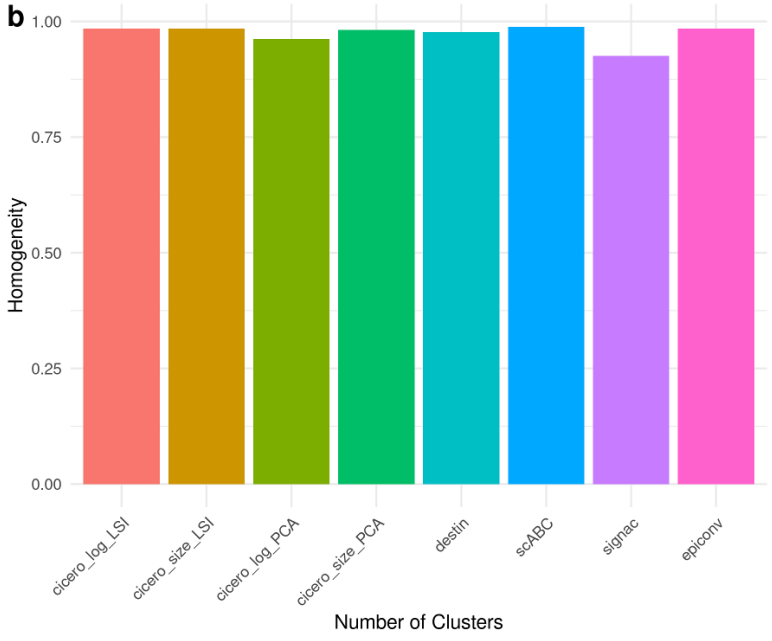


FIGURE 4.7: **The UMAP plots of clustering results on the Set-1 data.** The UMAP plots were generated by Signac using Set-1 as input. The top UMAP plot was labelled by the cell type and the H1975 showed two populations. The other eight plots were coloured by the clustering labels generated by each tool and the ARI values were also indicated on the right.



(A)



(B)

FIGURE 4.8: **Evaluation of the clustering step of five selected scTAC-seq data analysis tools.** Two normalisation methods (log or size only) and two dimension reduction methods (LSI or PCA) were applied for `Cicero`. (A) The Adjusted Rand Index ARI and number of clusters generated for each clustering method. (B) The homogeneity of the clusters generated by each tool.

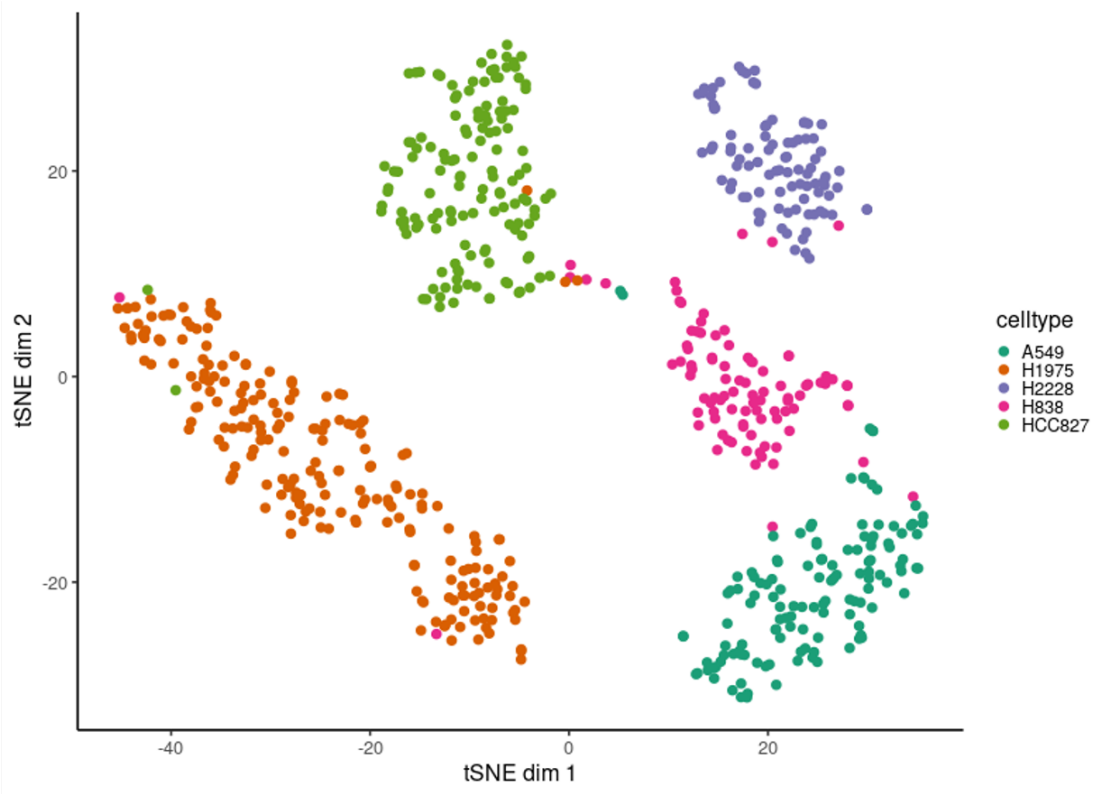


FIGURE 4.9: **The tSNE plot of the Set-1 data generated by chromVar [4].**

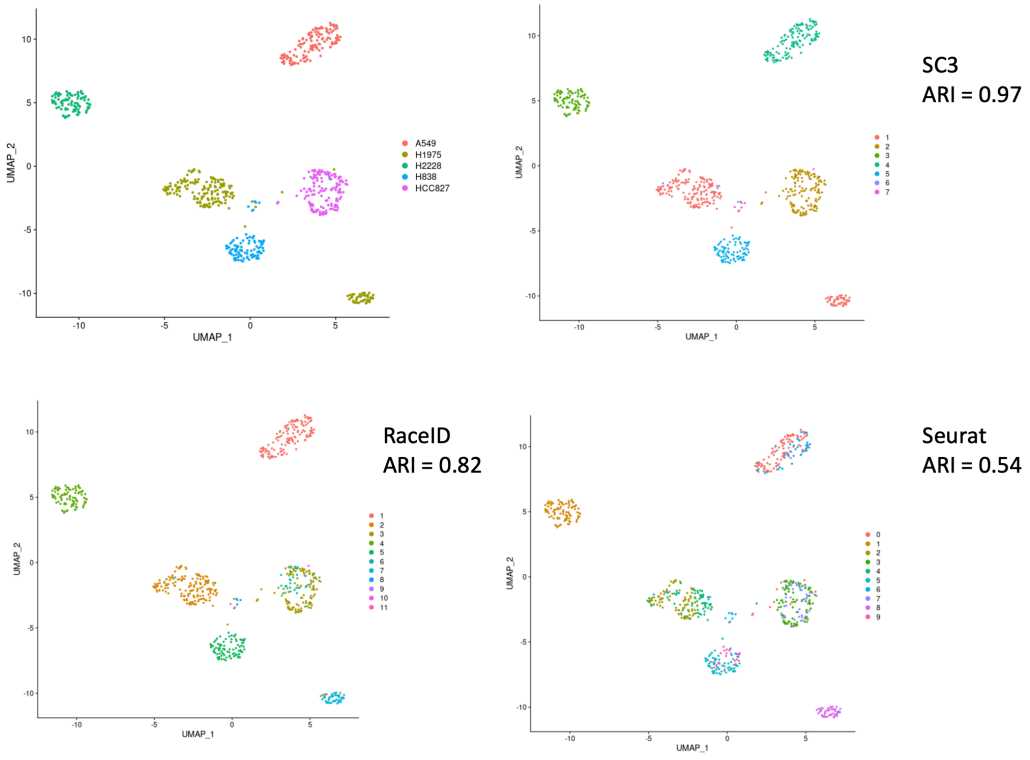


FIGURE 4.10: **The UMAP plots for clustering results using scRNA-seq clustering tools.** The plots were obtained using Signac. The ARI value achieved by each tool was labelled.

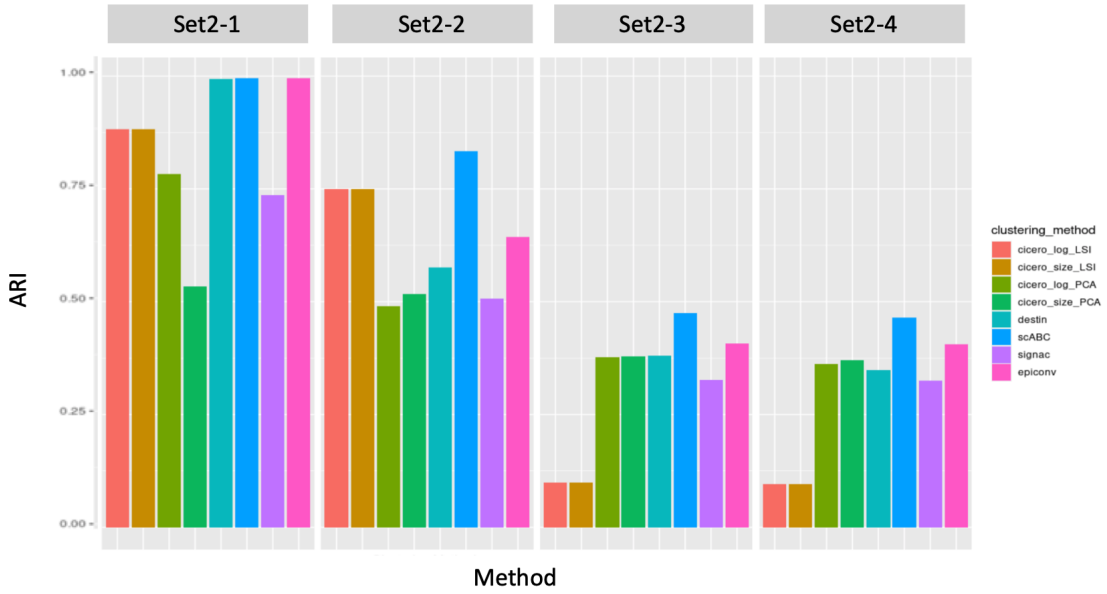


FIGURE 4.11: **Performance of clustering methods assessed using ARI on data of variable quality.** From left to right, each plot represents the clustering performance in terms of ARI on different sub-sets.

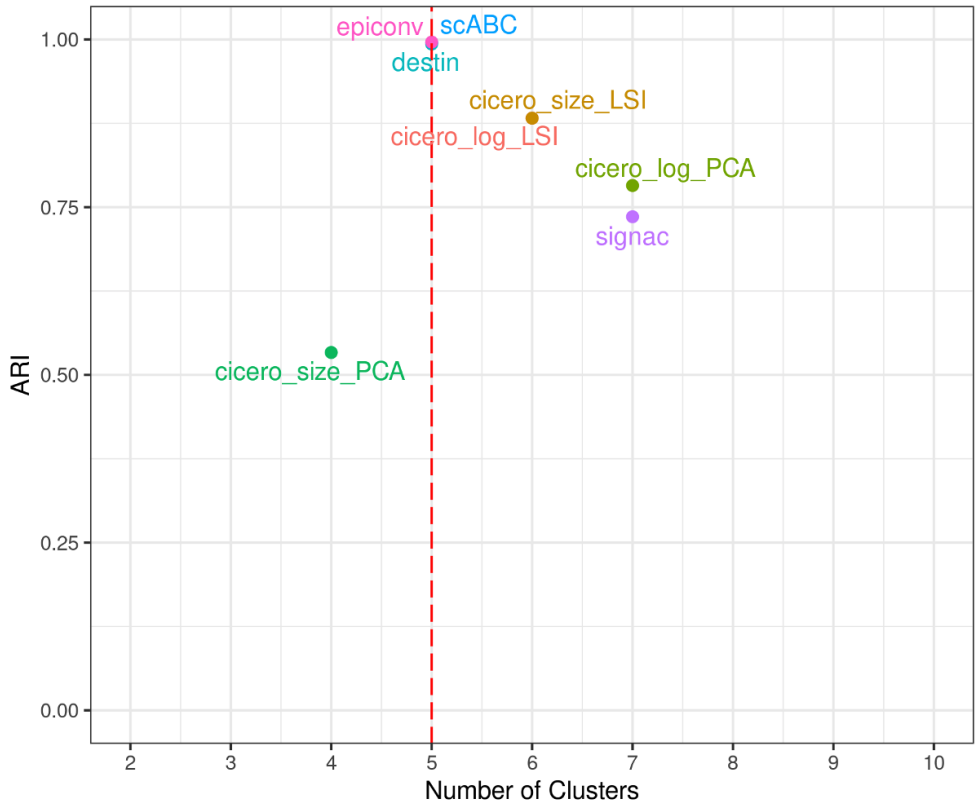


FIGURE 4.12: **The ARI against the number of clusters on the Set-2-1 data.** The designed cell type number was five and is indicated by the red dashed line. For Cicero with size normalisation and PCA, only four clusters were identified. In contrast, Signac and Cicero using log normalisation and PCA each generated seven clusters.

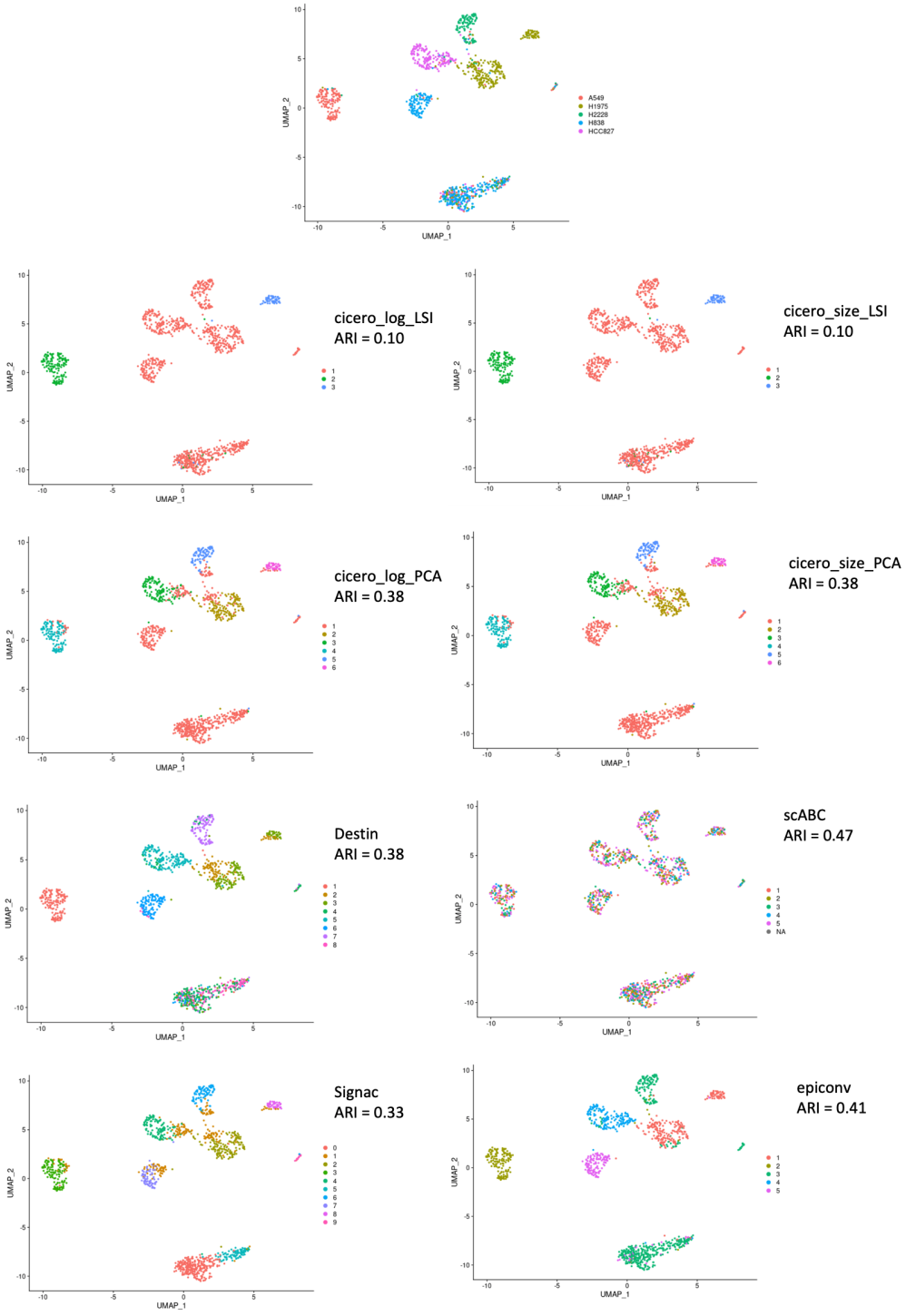


FIGURE 4.13: **The UMAP plots of clustering results on the Set-2-3 data**. The UMAP plots were generated by Signac using the Set-2-3 data as input. The top UMAP plot was labelled by cell type and the H1975 cells can be seen to form two distinct clusters. The other eight plots were coloured by the clustering labels generated by each tool and the ARI values were also indicated on the right.

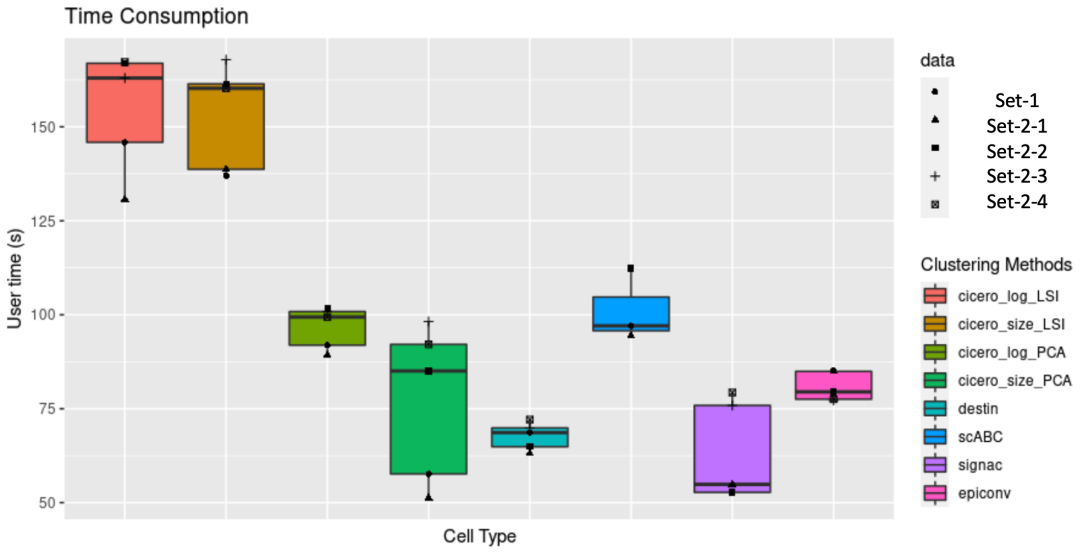


FIGURE 4.14: **The run-time of clustering methods.** The shape of the point indicates the different data sets (more or less stringent barcode filtering). The colour indicates different clustering methods.

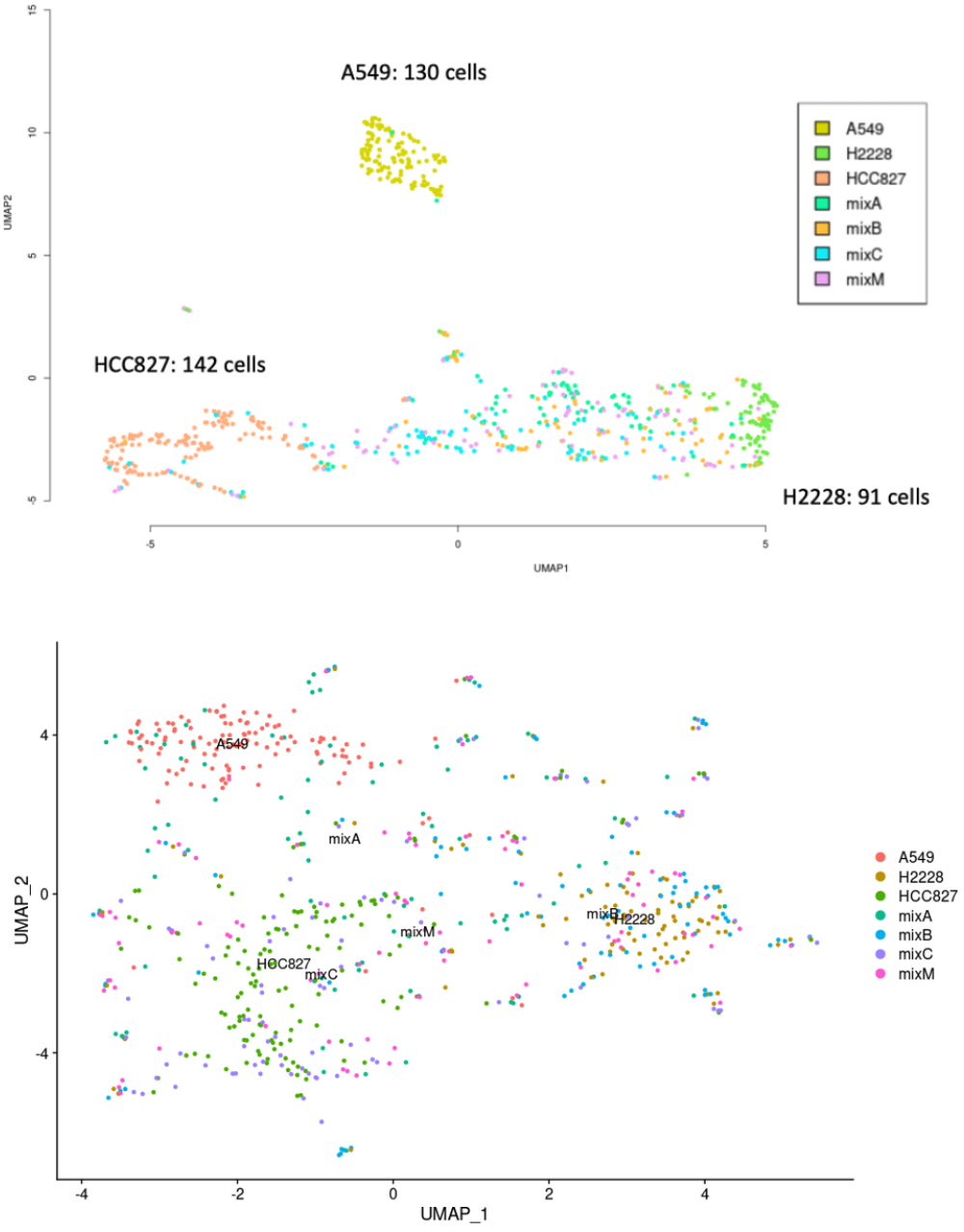


FIGURE 4.15: **The UMAP representation of simulated data.** The top UMAP was plotted using cisTopic [61] and the bottom UMAP was plotted using Signac [26] . Three cell types were well separated, however, the simulated cells randomly located between HCC827 cells and H3338 cells. Mix A, B and C were designed to have 68% of reads from the single cell of one cell line and 13% reads each from other two single cells. In Mix A, the dominant cell type was A549. In Mix B, the dominant cell type was H2228. In Mix C, the dominant cell type was HCC827. In Mix M, 33% of reads from each of the single-cell was merged to form the single cell.

Chapter 5

Discussion

This study benchmarked the clustering tools for scATAC-seq data using an in-house data set which contained an equal proportion of cells from five cell lines. We first summarised the computational pipelines, tools or packages specifically designed for scATAC-seq analysis in terms of their desired input data format, feature matrix construction methods, clustering algorithms, available downstream analysis steps and documentation. We emphasised the importance of documentation because these analysis pipelines involve multiple steps, including data manipulation, package installation and statistical and machine learning knowledge, which can present challenges for new users. The availability of detailed documentation helps users to understand the reasons behind performing certain steps and can explain parameter selection criteria and caveats during analysis. We recommended ArchR [13], Cell Ranger ATAC, Cicero [16] and Garnett [16], scATAC-pro [11], Signac [26] and SnapATAC [27], because these tools have very detailed documentation and a wide range of downstream analysis tools.

Next, we benchmarked the five selected R-based scATAC-seq specific clustering tools which covered two of the three most common clustering algorithms, i.e. Louvain and k-means, using our contrl data set with cell labels assigned via demuxlet [71] based on cell line specific genotypic variation. The clustering step was composed of several steps: binarisation, normalisation, dimension reduction and clustering. The selected tools captured variable combinations of these intermediate steps. We found that all the scATAC-seq clustering tools benchmarked performed well on a carefully filtered clean sub-set of these data for which each cell was required to have more than 5,000

insertion fragments. All tools achieved Adjusted Rand Index (ARI) above 0.7 and high AMI, NMI, homogeneity, completeness and v-measures, even though some tools identified fewer clusters than the designed five cell lines. Apart from that, two of the three selected clustering tools for scRNA-seq data, i.e. SC3 [75] and RaceID [76], were also able to cluster these scATAC-seq data correctly, achieving ARI above 0.8 using the default parameters. Seurat [49] was also able to cluster accurately when the input resolution was adjusted. In summary, the designed data set had a simple structure (i.e. five distinct clusters) for a clustering task and the clustering tools all performed well with acceptable accuracy compared to the ground truth labels.

Finally, we compared the performance of clustering tools on four sub-sets of the mixture benchmarking data set that varied in quality. We found that the performance of all tools declined, as measured by reduced ARI values, as more and more low-quality barcodes were included in the analysis. This suggested that careful filtration of barcodes or cell calling at the pre-processing step was very important; otherwise the presence of noisy low-quality data will interfere with the clustering tool's ability to capture the real clusters present in the data. As for the run-time, each tool took less than 3 minutes on the provided data set that had a maximum 12,754 peaks and 1,283 cells. Cicero using Latent Semantic Indexing normalisation methods would took longer to run that the other methods, however, the time differences were small, suggesting that data sets with more cells were required to properly benchmark the scalability of the tools. As the designed data set showed a simple structure, more complicated data sets are also required to comprehensively benchmark the tools. Therefore we attempted to simulate pseudo-cells to create new sub-clusters that were closer to one another by mixing the reads from difference single cells in varying fixed proportions. However, our method was unsuitable probably due to the binary nature of scATAC-seq data; thus, other methods to increase data complexity were suggested to consider for future benchmarking studies.

This project successfully built up a benchmarking framework for scATAC-seq, which will greatly benefit future benchmarking investigations. We emphasised on the selection of independent ground truth for the data set, the use of tools like CellBench [68] to simplify code required and generate more combinations of intermediate steps, and the complexity of the data set. All these considerations would be valuable for the next more comprehensive benchmarking study.

The single-cell sequencing field, especially scATAC-seq, develops very fast. During this project, many new computational tools were developed, such as epiConv [19] and ArchR [13]. In the future we expect to see more complex large-scale scATAC-seq data sets on different organisms. Benchmarking studies such as this play an essential role in providing researchers with information on the most suitable tools for their analysis requirements and feedback to areas for further development of the field.

5.1 Improved barcode selection of Cell Ranger ATAC

During the course of this project, Cell Ranger ATAC improved its barcode selection criteria. The version (1.0.1) used in this project identified 1,320 cell barcodes on our data set. However, the recent version (1.2.0) identified 678 cell barcodes, which was more close to the number of cells we identified after more stringent filtering, with 547 of the 678 barcodes overlapping with our clean data set. This again emphasises the importance of the pre-processing and the cell calling steps.

5.2 Comparing to other benchmarking efforts on scATAC-seq data

As mentioned in Section 2.7.1, the only independent benchmarking research on scATAC-seq tools [10] focused on the variable feature matrix construction methods. In contrast, our study focused on the part of the feature matrix construction, including transformation and dimension reduction, and the clustering. Both studies evaluated the clustering accuracy using the ARI, AMI and homogeneity metrics. In Chen *et al.*'s study, the performance of feature matrix construction tools were reflected on the average of clustering performance using three commonly used clustering methods. In our study, the evaluation reflected the performances of different clustering tools. They suggested cisTopic [61] and SnapATAC [27] based on the ability to generate feature matrix. At the same time, we observed relative homogeneous performance of clustering tools, which indicates that more comprehensive benchmarking study was required for the future.

In our study, we used genetic variation information of the cell lines and demuxlet [71] to determine the ground truth of our data set, which was proved to be reliable based on

our results. In Chen *et al.* 's study [10], the 10x data set they analysed was assumed to contain 8 clusters based on other studies on that data set, thus they used the Cell Ranger ATAC output based on 8 clusters, without any further annotation, as the ground truth. The advantages of this previous benchmarking work that could be adopted in future benchmarking studies include the use of multiple data sets covering nearly all scATAC-seq protocols and the use of large data sets to benchmark the scalability. In the next section, we will discuss some strategies to improve our benchmarking study.

5.3 Comprehensive Benchmarking scATAC-seq Tools

5.3.1 Real large scale data sets

Due to the time constrains, this study only used our in-house scATAC-seq data set, which did not have many cells and had only included five distinct cell lines. Real biological data tends to be more complex with more "intermediate" cell states and more similar cell types. Therefore, the in-house data set was over-simplified, and future benchmarking work should consider the inclusion of more realistic or complex data sets. Some large scale complex data sets that may be useful for such purposes are listed below.

1. **Human cell atlas of fetal chromatin accessibility (Nov, 2020):** recently, a human cell atlas of fetal chromatin accessibility [82] (http://descartes.brotmanbaty.org was established. The data set sequenced around ∼800,000 single cells from 59 human fetal samples between 89 to 125 days postconception using sci-ATAC-seq3 protocol representing 15 organs. Cell types were annotated with the help from gene expression data, which could be used as the ground truth for benchmarking study.

2. **Mouse cell atlas of chromatin Accessibility (Aug, 2018):** The mouse cell atlas [7] (http://descartes.brotmanbaty.org or (https://atlas.gs.washington.edu/mouse-atac/) profiled chromatin accessibility of ∼100,000 single cells from 13 adult mouse tissues of 17 8-week old mice. Most of cells were annotated with a cell type using the chromatin accessibility data.

3. **Chromatin cell atlas of developing fly embryo (Mar, 2018):** The mouse cell atlas [46] (http://descartes.brotmanbaty.org or (https://shendurelab.

github.io/fly-atac/) profiled chromatin accessibility of ∼20,000 single cells from fixed *Drosophila melanogaster* embryos at 3 different stages of development (2-4, 6-8, and 10-12 hours after egg laying) using sci-ATAC-seq protocol, representing 18 cell types. This data set was developed to understand the dynamics of chromatin accessibility during embryo-genesis.

4. **Multiple PBMC data sets from 10x Genomics:** 10x Genomics provides several peripheral blood mononuclear cells (PBMCs) data sets from healthy donors. The number of cells profiled include 500, 1,000, 5,000 and 10,000. The cells were annotated through immuno-phenotyping using flow cytometry.

These data sets, especially the atlas data sets, capture great complexity of cell types and sub-types. The number of cells profiled is also very large, which would require significant computational resources in order to analyse them together. These data sets could also capture the developing states of particular cell sub-types, which could be used to benchmark the trajectory inference methods. The large cell number could also be exploited to benchmark the computational run-time and scalability of different tools by inputting data sets of varying cell number to the analysis. Scalability is required for future scATAC-seq computational tools as more and more large data set will be generated. The complex structure of these data sets will also challenge the clustering tools.

5.3.2 Simulated data sets

Another method to obtain complex data sets is through `in silico` simulation using either the bulk ATAC-seq data set or the single-cell data set to increase data complexity. Previous benchmarking work [10] simulated scATAC-seq data sets using bulk ATAC-seq data to generate different library sizes and noise levels. The recently published simATAC [80] method takes a bin-by-cell matrix of scATAC-seq data and simulates synthetic data sets in a bin-by-cell or feature-by-cell matrix that are similar to the input data.

Interestingly, SCAN-ATAC sim [81] simulates scATAC-seq data by down-sampling the reads from bulk ATAC-seq `bam` files, which is not at matrix level but at the read level. It firstly defines cell-type-specific foreground peaks and unified background regions. For the simulation of the single-cell, the regions from foreground and background are sampled

separately without replacement, and the probability is related to the average of the region. Reads are sampled from the selected regions, and the foreground reads and background reads are merged. However, the paper only introduced the method and presented run-time efficiency and did not show the quality of the simulated reads. Its strategy of considering the foreground and background regions during sampling could potentially be used to improve our simulation method mentioned in Section 3.3.

5.3.3 Benchmarking framework

This study could also be improved by adding more aspects of the analysis steps. We could benchmark pre-processing steps for scATAC-seq as they are critical and dramatically affect the downstream analysis results. The input could be `fastq` files or `bam` files, and the endpoint for pre-processing could be generating a filtered feature matrix that is able to perform clustering. The angles for evaluation of benchmarking pre-processing steps include cell calling results, run-time and storage, steps performed and the similarity to ground truth cell type after clustering. In this case, the clustering method will not change, and the purpose of clustering is to evaluate the performance of the pre-processing steps. Similar to the Chen *et al.* paper [81], several clustering tools could be used, and the evaluation is based on average performance using the selected clustering tool. The cell calling steps also need to be benchmarked because the tools use different methods as mentioned in Section 2.5.1.2 and can generate different numbers of cells. The run-time and space requirement for pre-processing should also be compared as these steps are the most time-consuming part of the analysis workflow, and the raw data is usually very large and can be hard to manage.

In our study, only five clustering tools for scATAC-seq were selected. In the future benchmarking work, more clustering tools across all computational languages could be added. As the hyper-parameter tuning is important, we could also add a parameter selection grid for tools to select the optimal hyper-parameter. The run-time of the tools could be evaluated using one of the larger data sets mentioned in Section 5.3 to measure the scalability of the tools, as some of the tool developed early on when data sets tended to be small may scale poorly when applied to newer, larger data sets that are becoming increasingly common in the scATAC-seq literature.

To benchmark more comprehensively, we could select several representative feature matrices as input to benchmark the downstream analysis tools including clustering, trajectory analysis and data integration. Using CellBench [68] will greatly reduce the code replication and simplify parameter tuning. To benchmark the trajectory inference methods, we need to use well-annotated data sets showing the development or differentiation of cells, and compare the inferred trajectory time points and developing sequences of the cells with the true developing sequences by calculating the correlations between true and proposed pseudotime and proportion of cells correctly assigned to the trajectory.

Chapter 6

Conclusion

In conclusion, this study benchmarked the selected scATAC-seq and scRNA-seq clustering tools on a custom mixture data set that comprised of five cell lines. We found that all the tools clustered the cells correctly by achieving high ARI score, and the run-time was very short. The performance of the tools reduced dramatically as the data set became noisier as more low-quality cells were included. None of the tools was stable enough to perform well with decreasing data quality. This suggested the importance of filtering during pre-processing and the importance of parameter tuning during clustering. Different data sets may suit different clustering algorithms.

This study created a benchmarking framework for scATAC-seq clustering tools, using the CellBench [68] package to simplify coding work and choosing representative evaluation metrics. This framework could easily be adapted to handle larger data sets, more complex parameter tuning and expanded to evaluate more clustering methods. This framework can also be extended to include additional analysis steps, for example pre-processing and other downstream analysis tasks, to create a more comprehensive benchmarking study.

The field of single-cell genomics, especially scATAC-seq, has rapidly evolved over the past five years. More sequencing protocols have been developed alongside new computational tools which have led to the release of more large-scale data sets. Benchmarking studies play an important role as they provide researchers with a detailed summary of the available tools and help guide the selection of the best performing analysis method in different analysis settings. This saves a researcher time, as the benchmarking results

can be used to select a short-list of tools best suited to the biological question they wish to answer instead of having to run all the methods on their data to try and determine the best tool themselves. Apart from that, benchmarking studies also advance the development of new computational tools.

Bibliography

[1] Ansuman T. Satpathy, Jeffrey M. Granja, Kathryn E. Yost, Yanyan Qi, Francesca Meschi, Geoffrey P. McDermott, Brett N. Olsen, Maxwell R. Mumbach, Sarah E. Pierce, M. Ryan Corces, Preyas Shah, Jason C. Bell, Darisha Jhutty, Corey M. Nemec, Jean Wang, Li Wang, Yifeng Yin, Paul G. Giresi, Anne Lynn S. Chang, Grace X. Y. Zheng, William J. Greenleaf, and Howard Y. Chang. Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion. *Nature Biotechnology*, 37(8):925–936, 2019. ISSN 1087-0156. doi: 10.1038/s41587-019-0206-z. URL http://www.nature.com/articles/s41587-019-0206-z.

[2] Caleb A. Lareau, Fabiana M. Duarte, Jennifer G. Chew, Vinay K. Kartha, Zach D. Burkett, Andrew S. Kohlway, Dmitry Pokholok, Martin J. Aryee, Frank J. Steemers, Ronald Lebofsky, and Jason D. Buenrostro. Droplet-based combinatorial indexing for massive-scale single-cell chromatin accessibility. *Nature Biotechnology*, 37(August), 2019. ISSN 1087-0156. doi: 10.1038/s41587-019-0147-6. URL http://www.nature.com/articles/s41587-019-0147-6.

[3] Darren A. Cusanovich, Riza Daza, Andrew Adey, Hannah A. Pliner, Lena Christiansen, Kevin L. Gunderson, Frank J. Steemers, Cole Trapnell, and Jay Shendure. Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science*, 348(6237):910–914, 5 2015. ISSN 10959203. doi: 10.1126/science.aab1601.

[4] Alicia N. Schep, Beijing Wu, Jason D. Buenrostro, and William J. Greenleaf. ChromVAR: Inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nature Methods*, 14(10):975–978, 2017. ISSN 15487105. doi: 10.1038/nmeth.4401.

[5] Jason D. Buenrostro, Paul G. Giresi, Lisa C. Zaba, Howard Y. Chang, and William J. Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods*, 10(12):1213–1218, 2013. ISSN 15487091. doi: 10.1038/nmeth.2688.

[6] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Computational Biology*, 2018. ISSN 15537358. doi: 10.1371/journal.pcbi.1006245.

[7] Darren A. Cusanovich, Andrew J. Hill, Delasa Aghamirzaie, Riza M. Daza, Hannah A. Pliner, Joel B. Berletch, Galina N. Filippova, Xingfan Huang, Lena Christiansen, William S. DeWitt, Choli Lee, Samuel G. Regalado, David F. Read, Frank J. Steemers, Christine M. Disteche, Cole Trapnell, and Jay Shendure. A Single-Cell Atlas of In Vivo Mammalian Chromatin Accessibility. *Cell*, 174(5): 1309–1324, 2018. ISSN 10974172. doi: 10.1016/j.cell.2018.06.052. URL https://doi.org/10.1016/j.cell.2018.06.052.

[8] Jason D. Buenrostro, Beijing Wu, Ulrike M. Litzenburger, Dave Ruff, Michael L. Gonzales, Michael P. Snyder, Howard Y. Chang, and William J. Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523 (7561):486–490, 2015. ISSN 14764687. doi: 10.1038/nature14590.

[9] Anja Mezger, Sandy Klemm, Ishminder Mann, Kara Brower, Alain Mir, Magnolia Bostick, Andrew Farmer, Polly Fordyce, Sten Linnarsson, and William Greenleaf. High-throughput chromatin accessibility profiling at single-cell resolution. *Nature Communications*, 9(1):6–11, 12 2018. ISSN 20411723. doi: 10.1038/s41467-018-05887-x. URL http://www.nature.com/articles/s41467-018-05887-xhttp://dx.doi.org/10.1038/s41467-018-05887-x.

[10] Huidong Chen, Caleb Lareau, Tommaso Andreani, Michael E. Vinyard, Sara P. Garcia, Kendell Clement, Miguel A. Andrade-Navarro, Jason D. Buenrostro, and Luca Pinello. Assessment of computational methods for the analysis of single-cell ATAC-seq data. *bioRxiv*, page 739011, 8 2019. doi: 10.1101/739011.

[11] Wenbao Yu, Yasin Uzun, Qin Zhu, Changya Chen, and Kai Tan. scATAC-pro: a comprehensive workbench for single-cell chromatin accessibility sequencing data. *bioRxiv*, page 824326, 10 2019. doi: 10.1101/824326.

[12] Bin Li, Young Li, Kun Li, Lianbang Zhu, Qiaoni Yu, Pengfei Cai, Jingwen Fang, Wen Zhang, Pengcheng Du, Chen Jiang, Jun Lin, and Kun Qu. APEC: An accesson-based method for single-cell chromatin accessibility analysis. *Genome Biology*, 21 (1):116, 5 2020. ISSN 1474760X. doi: 10.1186/s13059-020-02034-y. URL https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02034-y.

[13] Jeffrey M. Granja, M. Ryan Corces, Sarah E. Pierce, S. Tansu Bagdatli, Hani Choudhry, Howard Y. Chang, and William J. Greenleaf. ArchR: An integrative and scalable software package for single-cell chromatin accessibility analysis. *bioRxiv*, page 2020.04.28.066498, 4 2020. doi: 10.1101/2020.04.28.066498. URL https://www.biorxiv.org/content/10.1101/2020.04.28.066498v1.

[14] Carl G. de Boer and Aviv Regev. BROCKMAN: Deciphering variance in epigenomic regulators by k-mer factorization. *BMC Bioinformatics*, 19(1):1–13, 2018. ISSN 14712105. doi: 10.1186/s12859-018-2255-6.

[15] Pacôme Prompsy, Pia Kirchmeier, and Céline Vallot. ChromSCape : a Shiny/R application for interactive analysis of single-cell chromatin profiles. *bioRxiv*, page 683037, 7 2019. doi: 10.1101/683037. URL https://www.biorxiv.org/content/early/2019/07/22/683037.

[16] Hannah A. Pliner, Jonathan S. Packer, José L. McFaline-Figueroa, Darren A. Cusanovich, Riza M. Daza, Delasa Aghamirzaie, Sanjay Srivatsan, Xiaojie Qiu, Dana Jackson, Anna Minkina, Andrew C. Adey, Frank J. Steemers, Jay Shendure, and Cole Trapnell. Cicero Predicts cis-Regulatory DNA Interactions from Single-Cell Chromatin Accessibility Data. *Molecular Cell*, 71(5):858–871, 2018. ISSN 10974164. doi: 10.1016/j.molcel.2018.06.044.

[17] Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nature Methods*, 16(5):397–400, 5 2019. ISSN 15487105. doi: 10.1038/s41592-019-0367-1. URL http://dx.doi.org/10.1038/s41592-019-0367-1.

[18] Eugene Urrutia, Li Chen, Haibo Zhou, and Yuchao Jiang. Destin: toolkit for single-cell analysis of chromatin accessibility. *Bioinformatics*, 3 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz141.

[19] Li Lin and Liye Zhang. Single-cell ATAC-seq clustering and differential analysis by convolution-based approach. *bioRxiv*, page 2020.02.13.947242, 2 2020. doi: 10.1101/2020.02.13.947242. URL https://www.biorxiv.org/content/10.1101/2020.02.13.947242v1https://www.biorxiv.org/content/10.1101/2020.02.13.947242v1.abstract.

[20] Hannah A. Pliner, Jay Shendure, and Cole Trapnell. Supervised classification enables rapid annotation of cell atlases. *Nature Methods*, 16(10):983–986, 10 2019. ISSN 15487105. doi: 10.1038/s41592-019-0535-3. URL https://www.nature.com/articles/s41592-019-0535-3.

[21] Mahdi Zamanighomi, Zhixiang Lin, Timothy Daley, Xi Chen, Zhana Duren, Alicia Schep, William J. Greenleaf, and Wing Hung Wong. Unsupervised clustering and epigenetic classification of single cells. *Nature Communications*, 9(1):1–8, 2018. ISSN 20411723. doi: 10.1038/s41467-018-04629-3. URL http://dx.doi.org/10.1038/s41467-018-04629-3.

[22] Lei Xiong, Kui Xu, Kang Tian, Yanqiu Shao, Lei Tang, Ge Gao, Michael Zhang, Tao Jiang, and Qiangfeng Cliff Zhang. SCALE method for single-cell ATAC-seq analysis via latent feature extraction. *Nature Communications*, 10(1), 12 2019. ISSN 20411723. doi: 10.1038/s41467-019-12630-7.

[23] Syed Murtuza Baker, Connor Rogerson, Andrew Hayes, Andrew D. Sharrocks, and Magnus Rattray. Classifying cells with Scasat, a single-cell ATAC-seq analysis tool. *Nucleic acids research*, 47(2):e10, 2019. ISSN 13624962. doi: 10.1093/nar/gky950.

[24] Zhicheng Ji, Weiqiang Zhou, and Hongkai Ji. Single-cell ATAC-seq Signal Extraction and Enhancement with SCATE. *bioRxiv*, page 795609, 2019. doi: 10.1101/795609. URL http://biorxiv.org/content/early/2019/10/07/795609.abstract.

[25] Zhicheng Ji, Weiqiang Zhou, and Hongkai Ji. Single-cell regulome data analysis by SCRAT. *Bioinformatics*, 33(18):2930–2932, 2017. ISSN 14602059. doi: 10.1093/bioinformatics/btx315.

[26] Tim Stuart, Avi Srivastava, Caleb Lareau, and Rahul Satija. Multimodal single-cell chromatin analysis with Signac. *bioRxiv*, page 2020.11.09.373613, 11 2020. doi: 10.1101/2020.11.09.373613. URL https://doi.org/10.1101/2020.11.09.373613.

[27] Rongxin Fang, Sebastian Preissl, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, Andrew K. Shiau, Eran A. Mukamel, Yanxiao Zhang, M. Margarita Behrens, Joseph Ecker, and Bing Ren. Fast and Accurate Clustering of Single Cell Epigenomes Reveals Cis-Regulatory Elements in Rare Cell Types. *bioRxiv*, pages 1–41, 2019. doi: 10.1101/615179. URL https://www.biorxiv.org/content/10.1101/615179v1.

[28] Huidong Chen, Luca Albergante, Jonathan Y. Hsu, Caleb A. Lareau, Giosuè Lo Bosco, Jihong Guan, Shuigeng Zhou, Alexander N. Gorban, Daniel E. Bauer, Martin J. Aryee, David M. Langenau, Andrei Zinovyev, Jason D. Buenrostro, Guo Cheng Yuan, and Luca Pinello. Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nature Communications*, 10(1):1–14, 12 2019. ISSN 20411723. doi: 10.1038/s41467-019-09670-4. URL https://www.nature.com/articles/s41467-019-09670-4.

[29] Karolin Luger, Armin W. Mäder, Robin K. Richmond, David F. Sargent, and Timothy J. Richmond. Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature*, 389(6648):251–260, 1997. ISSN 00280836. doi: 10.1038/38444.

[30] Timothy J. Richmond and Curt A. Davey. The structure of DNA in the nucleosome core. *Nature*, 423(6936):145–150, 5 2003. ISSN 00280836. doi: 10.1038/nature01595.

[31] Roger D. Kornberg. Chromatin structure: A repeating unit of histones and DNA. *Science*, 184(4139):868–871, 1974. ISSN 00368075. doi: 10.1126/science.184.4139.868.

[32] Steven Henikoff. Nucleosome destabilization in the epigenetic regulation of gene expression. *Nature Reviews Genetics*, 9(1):15–26, 1 2008. ISSN 14710056. doi: 10.1038/nrg2206.

[33] C. David Allis and Thomas Jenuwein. The molecular hallmarks of epigenetic control. *Nature Reviews Genetics*, 17(8):487–500, 8 2016. ISSN 14710064. doi: 10.1038/nrg.2016.59.

[34] Geoffrey P. Dann, Glen P. Liszczak, John D. Bagert, Manuel M. Müller, Uyen T.T. Nguyen, Felix Wojcik, Zachary Z. Brown, Jeffrey Bos, Tatyana Panchenko, Rasmus Pihl, Samuel B. Pollock, Katharine L. Diehl, C. David Allis, and Tom W. Muir. ISWI chromatin remodellers sense nucleosome modifications to determine substrate

preference. *Nature*, 548(7669):607–611, 8 2017. ISSN 14764687. doi: 10.1038/nature23671.

[35] Cheol Koo Lee, Yoichiro Shibata, Bhargavi Rao, Brian D. Strahl, and Jason D. Lieb. Evidence for nucleosome depletion at active regulatory regions genome-wide. *Nature Genetics*, 36(8):900–905, 8 2004. ISSN 10614036. doi: 10.1038/ng1400.

[36] Robert E. Thurman, Eric Rynes, Richard Humbert, Jeff Vierstra, Matthew T. Maurano, Eric Haugen, Nathan C. Sheffield, Andrew B. Stergachis, Hao Wang, Benjamin Vernot, Kavita Garg, Sam John, Richard Sandstrom, Daniel Bates, Lisa Boatman, Theresa K. Canfield, Morgan Diegel, Douglas Dunn, Abigail K. Ebersol, Tristan Frum, Erika Giste, Audra K. Johnson, Ericka M. Johnson, Tanya Kutyavin, Bryan Lajoie, Bum Kyu Lee, Kristen Lee, Darin London, Dimitra Lotakis, Shane Neph, Fidencio Neri, Eric D. Nguyen, Hongzhu Qu, Alex P. Reynolds, Vaughn Roach, Alexias Safi, Minerva E. Sanchez, Amartya Sanyal, Anthony Shafer, Jeremy M. Simon, Lingyun Song, Shinny Vong, Molly Weaver, Yongqi Yan, Zhancheng Zhang, Zhuzhu Zhang, Boris Lenhard, Muneesh Tewari, Michael O. Dorschner, R. Scott Hansen, Patrick A. Navas, George Stamatoyannopoulos, Vishwanath R. Iyer, Jason D. Lieb, Shamil R. Sunyaev, Joshua M. Akey, Peter J. Sabo, Rajinder Kaul, Terrence S. Furey, Job Dekker, Gregory E. Crawford, and John A. Stamatoyannopoulos. The accessible chromatin landscape of the human genome. *Nature*, 489(7414):75–82, 9 2012. ISSN 00280836. doi: 10.1038/nature11232.

[37] Marta Radman-Livaja and Oliver J. Rando. Nucleosome positioning: How is it established, and why does it matter? *Developmental Biology*, 339(2):258–266, 3 2010. ISSN 1095564X. doi: 10.1016/j.ydbio.2009.06.012.

[38] Maria Tsompana and Michael J. Buck. Chromatin accessibility: A window into the genome. *Epigenetics and Chromatin*, 7(1), 2014. ISSN 17568935. doi: 10.1186/1756-8935-7-33.

[39] Sandy L. Klemm, Zohar Shipony, and William J. Greenleaf. Chromatin accessibility and the regulatory epigenome. *Nature Reviews Genetics*, 20(4):207–220, 4 2019. ISSN 1471-0064. doi: 10.1038/s41576-018-0089-8. URL https://doi.org/10.1038/s41576-018-0089-8.

[40] Alan P Boyle, Sean Davis, Hennady P Shulha, Paul Meltzer, Elliott H Margulies, Zhiping Weng, Terrence S Furey, and Gregory E Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–22, 1 2008. ISSN 1097-4172. doi: 10.1016/j.cell.2007.12.014. URL http://www.ncbi.nlm.nih.gov/pubmed/18243105http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2669738.

[41] Jay R Hesselberth, Xiaoyu Chen, Zhihong Zhang, Peter J Sabo, Richard Sandstrom, Alex P Reynolds, Robert E Thurman, Shane Neph, Michael S Kuehn, William S Noble, Stanley Fields, and John A Stamatoyannopoulos. Global mapping of protein-DNA interactions in vivo by digital genomic footprinting. *Nature methods*, 6(4):283–9, 4 2009. ISSN 1548-7105. doi: 10.1038/nmeth.1313. URL http://www.ncbi.nlm.nih.gov/pubmed/19305407http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2668528.

[42] Jakub Mieczkowski, April Cook, Sarah K Bowman, Britta Mueller, Burak H Alver, Sharmistha Kundu, Aimee M Deaton, Jennifer A Urban, Erica Larschan, Peter J Park, Robert E Kingston, and Michael Y Tolstorukov. MNase titration reveals differences between nucleosome occupancy and chromatin accessibility. *Nature communications*, 7:11485, 2016. ISSN 2041-1723. doi: 10.1038/ncomms11485. URL http://www.ncbi.nlm.nih.gov/pubmed/27151365http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4859066.

[43] Britta Mueller, Jakub Mieczkowski, Sharmistha Kundu, Peggy Wang, Ruslan Sadreyev, Michael Y Tolstorukov, and Robert E Kingston. Widespread changes in nucleosome accessibility without changes in nucleosome occupancy during a rapid transcriptional induction. *Genes & development*, 31(5):451–462, 2017. ISSN 1549-5477. doi: 10.1101/gad.293118.116. URL http://www.ncbi.nlm.nih.gov/pubmed/28356342http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5393060.

[44] Theresa K. Kelly, Yaping Liu, Fides D. Lay, Gangning Liang, Benjamin P. Berman, Peter A. Jones, Martin Müller, Theresia Rieser, Marcello Köthe, Bernd Keßler, Marcela Brissova, and Klaus Lunkwitz. Genome-wide mapping of nucleosome positioning and DNA methylation within individual DNA molecules. *Genome Research*, 22(12), 12 2012.

[45] M. Ryan Corces, Jason D. Buenrostro, Beijing Wu, Peyton G. Greenside, Steven M. Chan, Julie L. Koenig, Michael P. Snyder, Jonathan K. Pritchard, Anshul Kundaje, William J. Greenleaf, Ravindra Majeti, and Howard Y. Chang. Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nature Genetics*, 2016. ISSN 15461718. doi: 10.1038/ng.3646.

[46] Darren A. Cusanovich, James P. Reddington, David A. Garfield, Riza M. Daza, Delasa Aghamirzaie, Raquel Marco-Ferreres, Hannah A. Pliner, Lena Christiansen, Xiaojie Qiu, Frank J. Steemers, Cole Trapnell, Jay Shendure, and Eileen E.M. Furlong. The cis-regulatory dynamics of embryonic development at single-cell resolution. *Nature*, 555(7697):538–542, 3 2018. ISSN 14764687. doi: 10.1038/nature25981.

[47] Sebastian Preissl, Rongxin Fang, Hui Huang, Yuan Zhao, Ramya Raviram, David U. Gorkin, Yanxiao Zhang, Brandon C. Sos, Veena Afzal, Diane E. Dickel, Samantha Kuan, Axel Visel, Len A. Pennacchio, Kun Zhang, and Bing Ren. Single-nucleus analysis of accessible chromatin in developing mouse forebrain reveals cell-type-specific transcriptional regulation. *Nature Neuroscience*, 21 (3):432–439, 3 2018. ISSN 15461726. doi: 10.1038/s41593-018-0114-4. URL http://www.ncbi.nlm.nih.gov/pubmed/29434377http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5862073http://www.nature.com/articles/s41593-018-0079-3.

[48] Jason D. Buenrostro, M. Ryan Corces, Caleb A. Lareau, Beijing Wu, Alicia N. Schep, Martin J. Aryee, Ravindra Majeti, Howard Y. Chang, and William J. Greenleaf. Integrated Single-Cell Analysis Maps the Continuous Regulatory Landscape of Human Hematopoietic Differentiation. *Cell*, 173(6):1535–1548, 2018. ISSN 10974172. doi: 10.1016/j.cell.2018.03.074. URL https://doi.org/10.1016/j.cell.2018.03.074.

[49] Rahul Satija, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5):495–502, 5 2015. ISSN 15461696. doi: 10.1038/nbt.3192.

[50] Seungbyn Baek and Insuk Lee. Single-cell ATAC sequencing analysis: from data preprocessing to hypothesis generation. *Computational and Structural Biotechnology Journal*, 18:1429–1439, 6 2020. ISSN 20010370. doi: 10.1016/j.csbj.2020.06.012.

[51] Simon Andrews. FastQC: A quality control tool for high throughput sequence data., 2010.

[52] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10, 5 2011. doi: 10.14806/ej.17.1.200.

[53] Anthony M. Bolger, Marc Lohse, and Bjoern Usadel. Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 8 2014. ISSN 14602059. doi: 10.1093/bioinformatics/btu170.

[54] Felix Krueger. Trim Galore! 2012. URL https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/.

[55] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 7 2009. ISSN 1367-4803. doi: 10.1093/bioinformatics/btp324. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp324.

[56] Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 4 2012. ISSN 15487091. doi: 10.1038/nmeth.1923.

[57] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 8 2009. ISSN 13674803. doi: 10.1093/bioinformatics/btp352.

[58] 10X Genomics. Cell Ranger ATAC. 2018.

[59] Yong Zhang, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nussbaum, Richard M Myers, Myles Brown, Wei Li, and X Shirley Liu. Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):R137, 2008. ISSN 1465-6906. doi: 10.1186/gb-2008-9-9-r137. URL http://genomebiology.biomedcentral.com/articles/10.1186/gb-2008-9-9-r137.

[60] Naim U. Rashid, Paul G. Giresi, Joseph G. Ibrahim, Wei Sun, and Jason D. Lieb. ZINBA integrates local covariates with DNA-seq data to identify broad and narrow regions of enrichment, even within amplified genomic regions. *Genome Biology*, 12 (7), 7 2011. ISSN 14747596. doi: 10.1186/gb-2011-12-7-r67.

[61] Carmen Bravo González-Blas, Liesbeth Minnoye, Dafni Papasokrati, Sara Aibar, Gert Hulselmans, Valerie Christiaens, Kristofer Davie, Jasper Wouters, and Stein Aerts. Cis-topic modelling of single-cell epigenomes. *bioRxiv*, page 370346, 2018. doi: 10.1101/370346. URL https://www.biorxiv.org/content/10.1101/370346v1.

[62] Luca Scrucca, Michael Fop, T. Brendan Murphy, and Adrian E. Raftery. Mclust 5: Clustering, classification and density estimation using Gaussian finite mixture models. *R Journal*, 8(1):289–317, 2016. ISSN 20734859. doi: 10.32614/rj-2016-021.

[63] Matthias Studer. WeightedCluster Library Manual: A practical guide to creating typologies of trajectories in the social sciences with R. *LIVES Working Papers*, 24, 2013. doi: http://dx.doi.org/10.12682/lives.2296-1658.2013.24.

[64] Vincent D. Blondel, Jean Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), 10 2008. ISSN 17425468. doi: 10.1088/1742-5468/2008/10/P10008.

[65] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579 – 2605, 2008. URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[66] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2 2018. URL http://arxiv.org/abs/1802.03426.

[67] Ian Dunham, Anshul Kundaje, Shelley F. Aldred, Patrick J. Collins, Carrie A. Davis, Francis Doyle, Charles B. Epstein, Seth Frietze, Jennifer Harrow, Rajinder Kaul, Jainab Khatun, Bryan R. Lajoie, Stephen G. Landt, Bum Kyu Lee, Florencia Pauli, Kate R. Rosenbloom, Peter Sabo, Alexias Safi, Amartya Sanyal, Noam Shoresh, Jeremy M. Simon, Lingyun Song, Nathan D. Trinklein, Robert C. Altshuler, Ewan Birney, James B. Brown, Chao Cheng, Sarah Djebali, Xianjun Dong, Jason Ernst, Terrence S. Furey, Mark Gerstein, Belinda Giardine, Melissa Greven, Ross C. Hardison, Robert S. Harris, Javier Herrero, Michael M. Hoffman, Sowmya Iyer, Manolis Kellis, Pouya Kheradpour, Timo Lassmann, Qunhua Li, Xinying Lin, Georgi K. Marinov, Angelika Merkel, Ali Mortazavi, Stephen C.J.

Parker, Timothy E. Reddy, Joel Rozowsky, Felix Schlesinger, Robert E. Thurman, Jie Wang, Lucas D. Ward, Troy W. Whitfield, Steven P. Wilder, Weisheng Wu, Hualin S. Xi, Kevin Y. Yip, Jiali Zhuang, Bradley E. Bernstein, Eric D. Green, Chris Gunter, Michael Snyder, Michael J. Pazin, Rebecca F. Lowdon, Laura A.L. Dillon, Leslie B. Adams, Caroline J. Kelly, Julia Zhang, Judith R. Wexler, Peter J. Good, Elise A. Feingold, Gregory E. Crawford, Job Dekker, Laura Elnitski, Peggy J. Farnham, Morgan C. Giddings, Thomas R. Gingeras, Roderic Guigó, Timothy J. Hubbard, W. James Kent, Jason D. Lieb, Elliott H. Margulies, Richard M. Myers, John A. Stamatoyannopoulos, Scott A. Tenenbaum, Zhiping Weng, Kevin P. White, Barbara Wold, Yanbao Yu, John Wrobel, Brian A. Risk, Harsha P. Gunawardena, Heather C. Kuiper, Christopher W. Maier, Ling Xie, Xian Chen, Tarjei S. Mikkelsen, Shawn Gillespie, Alon Goren, Oren Ram, Xiaolan Zhang, Li Wang, Robbyn Issner, Michael J. Coyne, Timothy Durham, Manching Ku, Thanh Truong, Matthew L. Eaton, Alex Dobin, Andrea Tanzer, Julien Lagarde, Wei Lin, Chenghai Xue, Brian A. Williams, Chris Zaleski, Maik Röder, Felix Kokocinski, Rehab F. Abdelhamid, Tyler Alioto, Igor Antoshechkin, Michael T. Baer, Philippe Batut, Ian Bell, Kimberly Bell, Sudipto Chakrabortty, Jacqueline Chrast, Joao Curado, Thomas Derrien, Jorg Drenkow, Erica Dumais, Jackie Dumais, Radha Duttagupta, Megan Fastuca, Kata Fejes-Toth, Pedro Ferreira, Sylvain Foissac, Melissa J. Fullwood, Hui Gao, David Gonzalez, Assaf Gordon, Cédric Howald, Sonali Jha, Rory Johnson, Philipp Kapranov, Brandon King, Colin Kingswood, Guoliang Li, Oscar J. Luo, Eddie Park, Jonathan B. Preall, Kimberly Presaud, Paolo Ribeca, Daniel Robyr, Xiaoan Ruan, Michael Sammeth, Kuljeet Singh Sandhu, Lorain Schaeffer, Lei Hoon See, Atif Shahab, Jorgen Skancke, Ana Maria Suzuki, Hazuki Takahashi, Hagen Tilgner, Diane Trout, Nathalie Walters, Huaien Wang, Yoshihide Hayashizaki, Alexandre Reymond, Stylianos E. Antonarakis, Gregory J. Hannon, Yijun Ruan, Piero Carninci, Cricket A. Sloan, Katrina Learned, Venkat S. Malladi, Matthew C. Wong, Galt P. Barber, Melissa S. Cline, Timothy R. Dreszer, Steven G. Heitner, Donna Karolchik, Vanessa M. Kirkup, Laurence R. Meyer, Jeffrey C. Long, Morgan Maddren, Brian J. Raney, Linda L. Grasfeder, Paul G. Giresi, Anna Battenhouse, Nathan C. Sheffield, Kimberly A. Showers, Darin London, Akshay A. Bhinge, Christopher Shestak, Matthew R. Schaner, Seul Ki Kim, Zhuzhu Z. Zhang, Piotr A. Mieczkowski, Joanna O. Mieczkowska, Zheng Liu, Ryan M. McDaniell, Yunyun Ni, Naim U. Rashid, Min Jae Kim, Sheera Adar, Zhancheng Zhang,

Tianyuan Wang, Deborah Winter, Damian Keefe, Vishwanath R. Iyer, Meizhen Zheng, Ping Wang, Jason Gertz, Jost Vielmetter, E. Christopher Partridge, Katherine E. Varley, Clarke Gasper, Anita Bansal, Shirley Pepke, Preti Jain, Henry Amrhein, Kevin M. Bowling, Michael Anaya, Marie K. Cross, Michael A. Muratet, Kimberly M. Newberry, Kenneth McCue, Amy S. Nesmith, Katherine I. Fisher-Aylor, Barbara Pusey, Gilberto DeSalvo, Stephanie L. Parker, Sreeram Balasubramanian, Nicholas S. Davis, Sarah K. Meadows, Tracy Eggleston, J. Scott Newberry, Shawn E. Levy, Devin M. Absher, Wing H. Wong, Matthew J. Blow, Axel Visel, Len A. Pennachio, Hanna M. Petrykowska, Alexej Abyzov, Bronwen Aken, Daniel Barrell, Gemma Barson, Andrew Berry, Alexandra Bignell, Veronika Boychenko, Giovanni Bussotti, Claire Davidson, Gloria Despacio-Reyes, Mark Diekhans, Iakes Ezkurdia, Adam Frankish, James Gilbert, Jose Manuel Gonzalez, Ed Griffiths, Rachel Harte, David A. Hendrix, Toby Hunt, Irwin Jungreis, Mike Kay, Ekta Khurana, Jing Leng, Michael F. Lin, Jane Loveland, Zhi Lu, Deepa Manthravadi, Marco Mariotti, Jonathan Mudge, Gaurab Mukherjee, Cedric Notredame, Baikang Pei, Jose Manuel Rodriguez, Gary Saunders, Andrea Sboner, Stephen Searle, Cristina Sisu, Catherine Snow, Charlie Steward, Electra Tapanari, Michael L. Tress, Marijke J. Van Baren, Stefan Washietl, Laurens Wilming, Amonida Zadissa, Zhengdong Zhang, Michael Brent, David Haussler, Alfonso Valencia, Nick Addleman, Roger P. Alexander, Raymond K. Auerbach, Suganthi Balasubramanian, Keith Bettinger, Nitin Bhardwaj, Alan P. Boyle, Alina R. Cao, Philip Cayting, Alexandra Charos, Yong Cheng, Catharine Eastman, Ghia Euskirchen, Joseph D. Fleming, Fabian Grubert, Lukas Habegger, Manoj Hariharan, Arif Harmanci, Sushma Iyengar, Victor X. Jin, Konrad J. Karczewski, Maya Kasowski, Phil Lacroute, Hugo Lam, Nathan Lamarre-Vincent, Jin Lian, Marianne Lindahl-Allen, Renqiang Min, Benoit Miotto, Hannah Monahan, Zarmik Moqtaderi, Xinmeng J. Mu, Henriette O'Geen, Zhengqing Ouyang, Dorrelyn Patacsil, Debasish Raha, Lucia Ramirez, Brian Reed, Minyi Shi, Teri Slifer, Heather Witt, Linfeng Wu, Xiaoqin Xu, Koon Kiu Yan, Xinqiong Yang, Kevin Struhl, Sherman M. Weissman, Luiz O. Penalva, Subhradip Karmakar, Raj R. Bhanvadia, Alina Choudhury, Marc Domanus, Lijia Ma, Jennifer Moran, Alec Victorsen, Thomas Auer, Lazaro Centanin, Michael Eichenlaub, Franziska Gruhl, Stephan Heermann, Burkhard Hoeckendorf, Daigo Inoue, Tanja Kellner, Stephan Kirchmaier, Claudia Mueller, Robert Reinhardt, Lea Schertel, Stephanie Schneider, Rebecca Sinn, Beate Wittbrodt, Jochen Wittbrodt, Gaurav

Jain, Gayathri Balasundaram, Daniel L. Bates, Rachel Byron, Theresa K. Canfield, Morgan J. Diegel, Douglas Dunn, Abigail K. Ebersol, Tristan Frum, Kavita Garg, Erica Gist, R. Scott Hansen, Lisa Boatman, Eric Haugen, Richard Humbert, Audra K. Johnson, Ericka M. Johnson, Tattyana V. Kutyavin, Kristen Lee, Dimitra Lotakis, Matthew T. Maurano, Shane J. Neph, Fiedencio V. Neri, Eric D. Nguyen, Hongzhu Qu, Alex P. Reynolds, Vaughn Roach, Eric Rynes, Minerva E. Sanchez, Richard S. Sandstrom, Anthony O. Shafer, Andrew B. Stergachis, Sean Thomas, Benjamin Vernot, Jeff Vierstra, Shinny Vong, Hao Wang, Molly A. Weaver, Yongqi Yan, Miaohua Zhang, Joshua M. Akey, Michael Bender, Michael O. Dorschner, Mark Groudine, Michael J. MacCoss, Patrick Navas, George Stamatoyannopoulos, Kathryn Beal, Alvis Brazma, Paul Flicek, Nathan Johnson, Margus Lukk, Nicholas M. Luscombe, Daniel Sobral, Juan M. Vaquerizas, Serafim Batzoglou, Arend Sidow, Nadine Hussami, Sofia Kyriazopoulou-Panagiotopoulou, Max W. Libbrecht, Marc A. Schaub, Webb Miller, Peter J. Bickel, Balazs Banfai, Nathan P. Boley, Haiyan Huang, Jingyi Jessica Li, William Stafford Noble, Jeffrey A. Bilmes, Orion J. Buske, Avinash D. Sahu, Peter V. Kharchenko, Peter J. Park, Dannon Baker, James Taylor, and Lucas Lochovsky. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, 9 2012. ISSN 14764687. doi: 10.1038/nature11247.

[68] Shian Su, Luyi Tian, Xueyi Dong, Peter F Hickey, Saskia Freytag, and Matthew E Ritchie. CellBench: R/Bioconductor software for comparing single-cell RNA-seq analysis methods. doi: 10.1093/bioinformatics/btz889. URL https://academic.oup.com/bioinformatics/advance-article-abstract/doi/10.1093/bioinformatics/btz889/5645177.

[69] Luyi Tian, Xueyi Dong, Saskia Freytag, Kim Anh Lê Cao, Shian Su, Abolfazl JalalAbadi, Daniela Amann-Zalcenstein, Tom S. Weber, Azadeh Seidi, Jafar S. Jabbari, Shalin H. Naik, and Matthew E. Ritchie. Benchmarking single cell RNA-sequencing analysis pipelines using mixture control experiments. *Nature Methods*, 16(6):479–487, 2019. ISSN 15487105. doi: 10.1038/s41592-019-0425-8. URL http://dx.doi.org/10.1038/s41592-019-0425-8.

[70] 10x Genomics. Chromium Single Cell ATAC Reagent Kits. URL https://support.10xgenomics.com/single-cell-atac/library-prep/doc/user-guide-chromium-single-cell-atac-reagent-kits-user-guide-v1-chemistry.

[71] Hyun Min Kang, Meena Subramaniam, Sasha Targ, Michelle Nguyen, Lenka Maliskova, Elizabeth McCarthy, Eunice Wan, Simon Wong, Lauren Byrnes, Cristina M. Lanata, Rachel E. Gate, Sara Mostafavi, Alexander Marson, Noah Zaitlen, Lindsey A. Criswell, and Chun Jimmie Ye. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology*, 36(1): 89–94, 1 2018. ISSN 15461696. doi: 10.1038/nbt.4042.

[72] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7): 1888–1902, 2019. ISSN 10974172. doi: 10.1016/j.cell.2019.05.031. URL https://doi.org/10.1016/j.cell.2019.05.031.

[73] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 12 1985. ISSN 01764268. doi: 10.1007/BF01908075. URL https://link.springer.com/article/10.1007/BF01908075.

[74] J. Nowosad and T. F. Stepinski. Spatial association between regionalizations using the information-theoretical V-measure. *International Journal of Geographical Information Science*, 2018. ISSN 13623087. doi: 10.1080/13658816.2018.1511794.

[75] Vladimir Yu Kiselev, Kristina Kirschner, Michael T. Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N. Natarajan, Wolf Reik, Mauricio Barahona, Anthony R. Green, and Martin Hemberg. SC3: Consensus clustering of single-cell RNA-seq data. *Nature Methods*, 2017. ISSN 15487105. doi: 10.1038/nmeth.4236.

[76] Josip S. Herman, Sagar, and Dominic Grün. FateID infers cell fate bias in multipotent progenitors from single-cell RNA-seq data. *Nature Methods*, 2018. ISSN 15487105. doi: 10.1038/nmeth.4662.

[77] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 2018. ISSN 15461696. doi: 10.1038/nbt.4096.

[78] Angelo Duò, Mark D. Robinson, and Charlotte Soneson. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, 7: 1141, 2018. ISSN 20461402. doi: 10.12688/f1000research.15666.2.

[79] Saskia Freytag, Luyi Tian, Ingrid Lönnstedt, Milica Ng, and Melanie Bahlo. Comparison of clustering tools in R for medium-sized 10x Genomics single-cell RNA-sequencing data. *F1000Research*, 7:1297, 2018. ISSN 20461402. doi: 10.12688/f1000research.15809.2.

[80] Zeinab Navidi Ghaziani, Lin Zhang, and Bo Wang. simATAC: A Single-cell ATAC-seq Simulation Framework. *bioRxiv*, 2020.

[81] Zhanlin Chen, Jing Zhang, Jason Liu, Zixuan Zhang, Jiangqi Zhu, Min Xu, and Mark Gerstein. SCAN-ATAC Sim: a scalable and efficient method to simulate single-cell ATAC-seq from bulk-tissue experiments. *bioRxiv*, page 2020.05.29.123638, 5 2020. doi: 10.1101/2020.05.29.123638. URL https://doi.org/10.1101/2020.05.29.123638.

[82] Silvia Domcke, Andrew J Hill, Riza M Daza, Junyue Cao, Diana R O'Day, Hannah A Pliner, Kimberly A Aldinger, Dmitry Pokholok, Fan Zhang, Jennifer H Milbank, Michael A Zager, Ian A Glass, Frank J Steemers, Dan Doherty, Cole Trapnell, Darren A Cusanovich, and Jay Shendure. A human cell atlas of fetal chromatin accessibility. *Science (New York, N.Y.)*, 370(6518), 11 2020. ISSN 1095-9203. doi: 10.1126/science.aba7612. URL http://www.ncbi.nlm.nih.gov/pubmed/33184180.