BINF90009 Bioinformatics Research Project

Progress Report

Comprehensive benchmarking of scATAC-Seq data analysis tools

Haoyu Yang (743501)

Supervisors: Matthew E. Ritchie, Shanika L. Amarasinghe

Word Count: 2,825

June 2020

University of Melbourne

Master of Science (Bioinformatics)

Contents

1 Research Overview

Single Cell Assay for Transposase Accessible Chromatin Sequencing (scATAC-seq) is a technology that allows the detection of chromatin accessibility at single cell resolution. The timeline for scATAC-seq protocol development and data production has been summarised by the first author of `SnapATAC` [1], as shown in Figure 1. The assay has been widely used to define cell types and states, identify master regulators, discover cis-regulatory regions and characterise gene regulatory networks.

The general workflow for scATAC-seq data analysis comprises of (1) preprocessing: demultiplexing, adaptor trimming (optional), read mapping, quality control, cell calling, peak calling (optional) and mutiplet removal (optional); (2) feature matrix construction: defining regions via peak calling or genome binning, counting defined features, transformation and dimensionality reduction; (3) downstream analysis: cell clustering, visualisation, differential accessibility analysis and cis-regulatory network analysis etc. [2, 3] (Figure 2). Currently there are around 24 scATAC-seq specific analysis tools, summarised in Figure 3. These tools can handle different steps in the analysis, with some taking care of the entire workflow from beginning to end, while others focus on specific downstream tasks. To better understand the strengths and weaknesses of different tools, it is important to benchmark their performance on datasets 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.

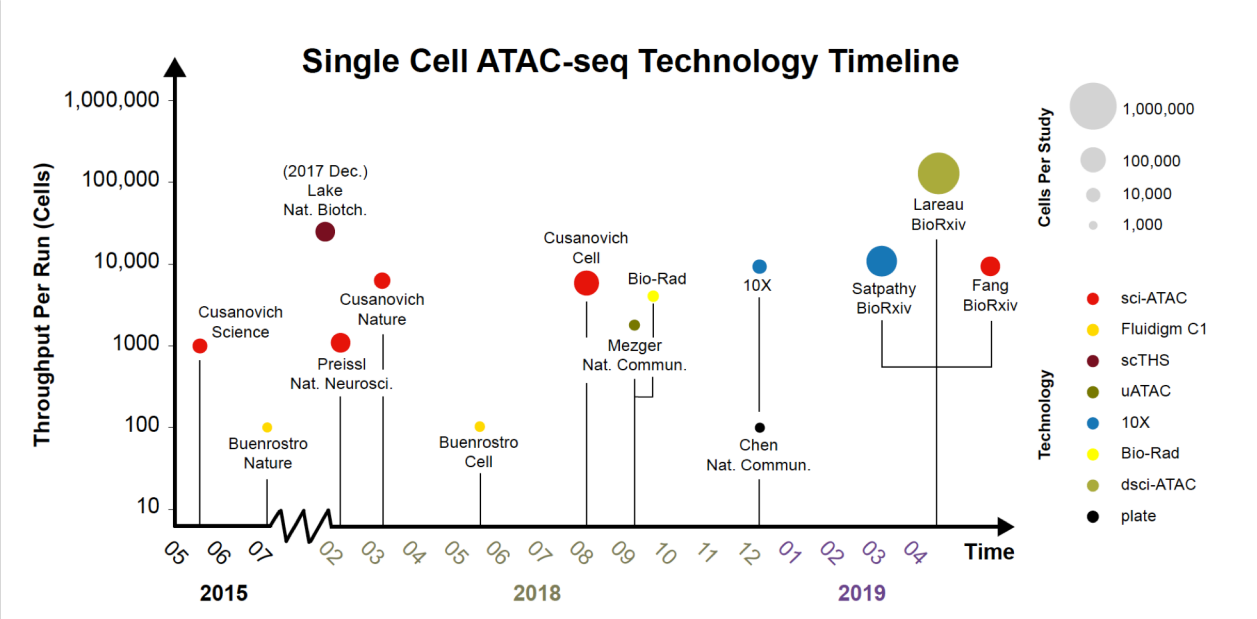


Figure 1: **Timetine of single cell ATAC-seq technology development and data generation.**
Source: https://github.com/r3fang/SnapATAC/blob/master/notebooks/experiemnt_timeline.md

This project aims to comprehensively evaluate computational tools for scATAC-seq data analysis. The aims stated in the research proposal include:

1. developing universal performance indicators to evaluate the performance of current scATAC-seq analysis tools
2. evaluating the performance of current scATAC-seq analysis tools using developed evaluation metrics:

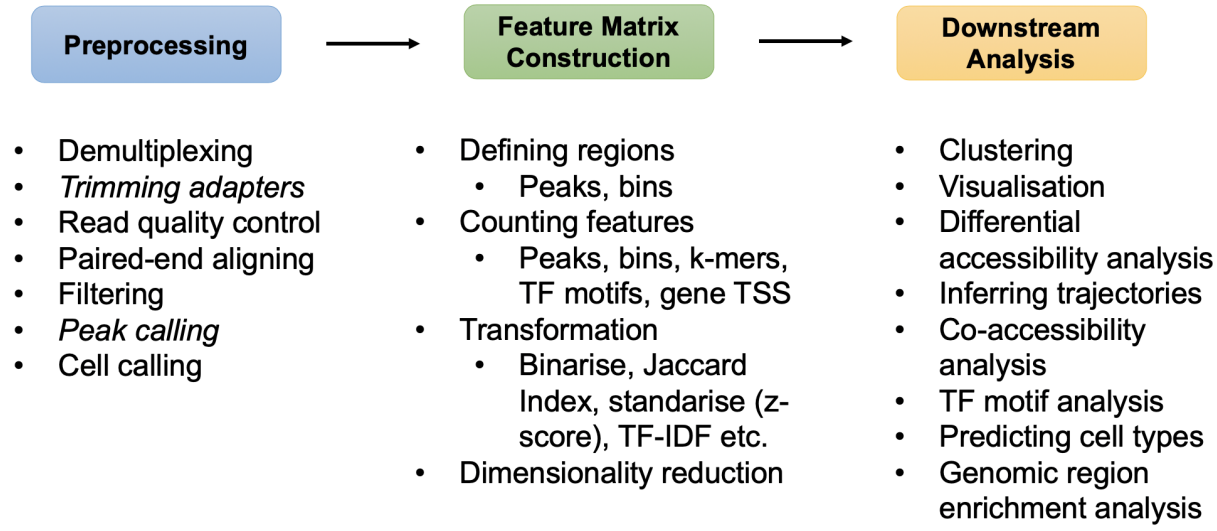


Figure 2: **A general workflow for scATAC-seq data analysis, adapted from Chen *et al.* (2019) [3]**

(a) on sequencing depth variation

(b) on data quality variation

(c) using default and customised parameters

(d) in terms of the ease of use of the tool

using both experimental and simulated datasets which have known cell group labels as ground-truth.

This progress report will mainly address Aims 1, Aim 2(b) and 2(c) above. To summarise what I have been doing in the past semester; first I summarised the available computational tools for scATAC-seq data analysis, focusing on the steps they are able to analyse and the input requirements of the tools. Next I performed several exploratory analyses on an in-house dataset, which included understanding the quality of the data, running example pipelines of some tools (results not shown) and running the scripts of another recent benchmarking paper [3] (results not shown). This was done with the objective to improve my `Linux` and `R` skills such as writing scripts to modify data formats and visualise the resulting datasets. Thereafter, I performed benchmarking on scATAC-seq data clustering tools using `CellBench` [4] package.

The input dataset for clustering benchmarking was pre-processed by the `CellRanger ATAC` software and the resulting peak-barcode matrix passed a stringent self designed filter of barcodes to produce a clean dataset. The cell type labels determined by `demuxlet` [5] were used as ground-truth for evaluation of the performance. Five `R` based clustering methods were selected covering multiple algorithms for normalisation, dimensionality reduction and clustering. Furthermore, I also assessed the performance of the clustering methods as data quality varied, which was achieved by applying several different filtering thresholds to the binary library size of the barcode (or cell). In addition, the amount of time each of these methods took to run was compared.

2 Work to Date

2.1 Updated summary of currently available scATAC-seq data analysis tools

Extensive work has been done in summarising scATAC-seq specific analysis tools, through reading corresponding publications and their `GitHub` entries. The updated summary mainly focus on the steps performed by each tool and the inputs they take (Figure 3).

Most tools are able to generate a feature matrix but some of them only focus on downstream analysis (e.g. `Signac` [6], `APEC` [7], `chromSCape`[8], `Garnett` [9], `STREAM` [10] and `epiConv` [11]). `Cicero` [12], `APEC` 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` are able to analyse co-accessibility. Some tools are able to take the output of `CellRanger ATAC` as input, including `ArchR`, `chromSCape`, `Destin`, `APEC`, `SnapATAC`, `scATAC-pro` [2], `Signac`, `epiConv`, `cisTopic` [13], `cicero`, `SCALE` and `STREAM`. Only `CellRanger ATAC`, `BROCKMAN` [14], `Destin`, `APEC`, `SnapATAC`, `scATAC-pro` are able to take fastq files as input.

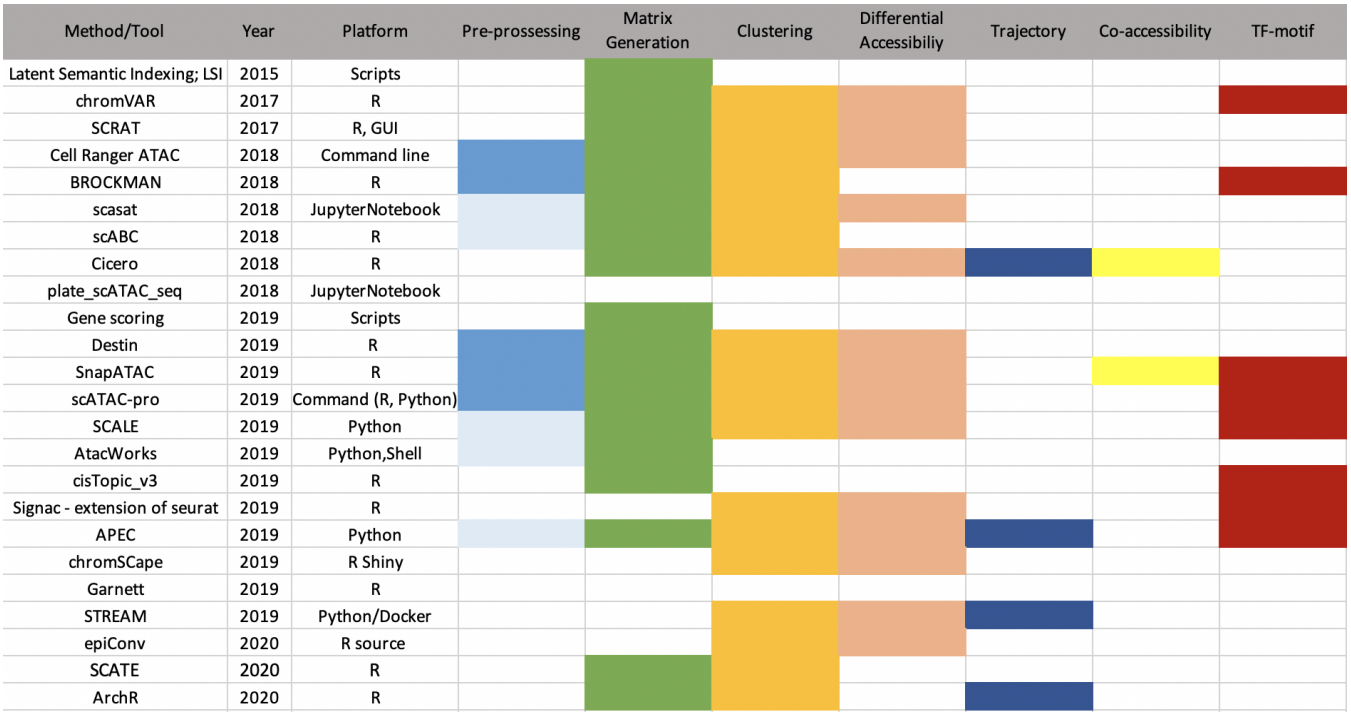


Figure 3: **Computational tools for scATAC-seq data analysis.**
The 4th to 10th columns are major steps of data analysis workflow,i.e. pre-processing, matrix generation, clustering, differential accessibility analysis, trajectory analysis, co-accessibility analysis and TF-motif analysis.

2.2 Selected scATAC-Seq clustering tools for Benchnmarking

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 report includes benchmarking results for five R based scATAC-Seq specific clustering

tools: `Cicero` (version 1.4.4) [12], `Destin` (version 1.0.1) [15] , `scABC` (version 0.99.0) [16] , `Signac` (version 0.2.5) [6] and `epiConv` [11]. These tools implemented different types of clustering algorithms, normalisation methods and dimensionality reduction methods as summarised in Table 1. `Cicero`, `Destin` and `epiConv` uses 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 requiring the user to provide this information.

Table 1: **Summary of selected scATAC-seq clustering tools.**

Tools	Binary	Normalisation	Dimension Reduction	Clustering Algorithm	Parameters	Provide 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
Signac	N	TF-IDF	LSI	Louvain or Leiden or Multilevel refinement Louvain or SLM	resolution	N
scABC	N	-	-	k-medoids + correlation analysis	k	Y
epiConv	Y	TF-IDF	UMAP	densityClust	k	Y

2.3 Data Preparation

2.3.1 Dataset Overview

The dataset contains 1,300 single cells from five lung cancer cell lines (i.e. H2228, H1975, H838, A549 and HCC827), which were sequenced using the Chromium 10x single cell platform using 10x Genomics protocol [17]. The dataset was pre-processed using 10x Genomics software `Cell Ranger ATAC` (version 1.0.1) which included aligning reads to the human genome (GRCh38), marking duplicates, peak calling, cell calling and generation of a peak by cell barcode matrix.

`Cell Ranger ATAC` outputs that were used in the downstream analysis include the following;

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

The `bam` file however, includes duplicates (marked with the sam flag 1024), mitochondria alignments and chimera alignments etc. There are 369,240 unique barcodes in this `bam` file, with 597,146,352 reads in total. 96.23% of the reads are mapped, 74.57% of the reads are duplicates and 94.79% of the reads are properly mapped. Looking at the distribution of read number across all barcodes, we find 13,833 barcodes with more than 1,000 reads and 1,463 barcodes with more than 10,000 reads were shown in Figure 4. For the 1,463 high read barcodes, the average number of reads per barcode is 365,471, which represents high read depths for these cells. The `raw peak-barcode matrix` contains the counts of fragment ends within each peak region for each barcode and has 112,754 peaks and 197,479 barcodes. These barcodes are generated by retaining reads that are pair-aligned with both reads having mapping quality (MAPQ) > 30, non-mitochondrial and non-chimerically mapped. Any 'non-cell' barcodes determined by the tool are next filtered out and the `filtered peak-barcode matrix` contains information from 1,320 'true' barcodes (i.e. cells).

The cell type of the barcodes were determined using genetic variation amongst the cells through `demuxlet` (version 0.0.1) with the `bam` file as input. 1,283 barcodes were assigned a cell type label that was regarded as ground-truth for the benchmarking analysis.

2.3.2 Barcode Selection

Before clustering, barcodes need to be selected to achieve a clean peak by barcode matrix foer use in benchmarking. Because the `CellRanger ATAC` determined 1,320 cell barcodes and `demuxlet` assigned 1,283 barcodes with a cell type label, one question is what are the differences and relationships between the two group of barcodes. As shown in Figure 5a, 970 barcodes had both a `demuxlet` cell type label and a high quality cell barcode. 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 H838 cells were not regarded as genuine cell barcodes by `CellRanger ATAC` as they had low read counts. I therefore, applied a more stringent filter need to remove barcodes with relatively fewer reads to clean up the dataset to improve data quality.

Firstly, a filter based on the number of high quality unique fragments was applied to keep the high quality barcodes (Figure 6). High quality means that both reads of the fragment have MAPQ > 30 and they are non-mitochondrial and non-chimerically mapped. The barcodes with less than or equal to 5,000 unique fragments and without cell type label were filtered out. The resulting number of barcodes were 669 and the number of barcodes remained for each cell type was summarised in Table 2.

Table 2: **Number of cells for each cell type after applying Filter 1**

Filter 1	**A549**	**H1975**	**H2228**	**H838**	**HCC827**	**Total**
No. Unique Fragments >5,000	130	212	91	94	142	669

The second filter considered the binary library sizes, which is the binarised column summation of the peak-barcode

matrix and also the binarised counts of fragments ends in the called peaks. Same as Filter 1, the barcodes without cell type label were removed. Multiple filtering threshold were considered because we were interested in the ability of the clustering tools to cope with different data quality. The threshold for filtration were set to be 1,000, 100 and binary library size for each barcode, and the selection of 1,000 was based on the distribution of the binary library size in each cell type, i.e. only select the group with relatively high binary library size barcodes. After applying this filter (Filter 2) with multiple thresholds, the number of remaining barcodes for each cell type was summarised in Table 3. 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 3: **Number of cells for each cell type after applying different level of Filter 2**

Filter 2	A549	H1975	H2228	H838	HCC827	Total
Binary Lib Size >1000	132	222	101	92	151	698
Binary Lib Size >100	164	256	128	155	190	893
Binary Lib Size >10	204	288	162	384	221	1259
Cell Type Label only	207	290	164	397	225	1283

2.4 Benchmarking evaluation metrics

To evaluate the clustering results, the adjusted Rand Index (ARI) [18] and homogeneity 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). The Rand Index (RI) is defined as:

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

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:

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)} \tag{2}$$

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

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{3}$$

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

The homogeneity score is defined as:

$$h = 1 - \frac{H(Y_{true}|Y_{pred})}{H(Y_{true})} \tag{4}$$

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

Both *ARI* and homogeneity are bounded between 0 and 1, with 1 indicating the best results and 0 indicating the worst.

2.5 Benchmarking Results

2.5.1 Comparing clustering methods when the input is a clean dataset

To benchmark the clustering step for the five selected tools (section 2.2), the *Filter 1* cleaned dataset was first used as input, which has five known cell type labels as ground-truth and high counts of unique fragments. `CellBench` [4] was used to allow running combinations of multiple clustering methods and evaluation methods on the dataset 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.

Figure 8 presents the performance of each of the clustering methods. All methods achieved high *ARI* with the lowest at 0.7 for `destin`. 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 8b, all methods achieved high homogeneity scores which were close to 1.

It was not surprising to see six clusters determined by many of the tools, as the tSNE plot of the input dataset indicated that the H1975 cells separate into two sub-clusters (Figure 9). The two sub-types may be generated due to experimental variation or different mutations (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 methods like `Cicero` and `Signac` are still reasonably good.

`Destin` also performed well on this clean dataset, but generated more clusters than expected.

2.5.2 Comparing clustering methods when the input has variable quality / increased noise

Next I compared the performance of clustering methods when the input data quality was varied. When the threshold of Filter 2 decreased from 1,000 to 0, more barcodes or cells with smaller library size were included in the dataset and the quality decreased. As shown in Figure 10, the *ARI* decreased as the data quality decreased for all methods tested. Even methods that were supplied with the expected number of clusters (5) such as `scABC`

and `epiConv` were unable to recapitulate the cell type labels as well as they did for the clean dataset. Performance generally suffered when cells with lower library sizes were included with the clean dataset, i.e. the tools were not able to cluster the cells correctly. Interestingly, `destin` correctly identified five clusters in the less stringently filtered dataset, while in previous results it over-clustered. However, `Cicero` with size only normalisation and PCA identified fewer clusters than expected in this dataset (data not shown) which lead to a reduction in ARI. Other methods generated similar clustering results between these two datasets. The two datasets contain a similar number of barcodes (669 vs 698) with 664 barcodes in common. This suggested that `destin` and `Cicero_size_PCA` were unstable. When lots of low quality barcodes were included the two `Cicero` methods using LSI tend to under-cluster (data not shown) with only three clusters identified.

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 a dimension reduced plot of the data (e.g. tSNE, PCA or UMAP) can help with this process.

2.5.3 Evaluating of the timing to run different clustering methods

Finally, I evaluated the time taken to run selected clustering methods. For the tested dataset with a feature matrix of 12,754 peaks by 1,283 cells, the longest time for clustering is less than 3 minutes. `Signac` achived the shortest median time and the two LSI methods of `Cicero` achieved the longest median time (Figure 11). Our dataset is relatively small, and for scalability of the tools to be assessed more thoroughly, larger datasets will need to be used in the future.

3 Future Work

The future work planned for this project include continuing these benchmarking efforts as follows:

- Include tools that are developed through other programming platforms than R (e.g. many are Python based);
- Add other evaluation metrics: for example the V-measure that can consider both homogeneity and completeness of the clustering performance;
- Include methods developed for scRNA-seq clustering analysis
- Consider other aspects of scATAC-seq analysis (e.g. trajectory analysis)

3.1 Timetable

The timeline for semester 2, 2020 is summarised in Figure 12.

July:

- Complete the benchmarking of clustering tools using different evaluation metrics, include tools not in R and assess the performance of methods on data with varying quality thresholds.

- Complete the benchmarking of trajectory analysis tools. Both scATAC-specific tools and those designed for scRNA-seq can be considered, using a publicly available dataset where trajectories are well understood [20].

August-September:

- More explorations focusing on parameter tuning of these tools for feature matrix generation.
- To improve the quality and depth of this project, the simulation of the datasets could be used to assess tool performance with variable noise levels on data with more similar cell states. By mixing of reads from single cells from different cell lines *in silico*, 'pseudo cells' can be created that form intermediate clusters that are controlled by the mixing proportion (e.g. 90:10, 75:25, 50:50 reads from cells randomly selected from line 1 versus cell line 2). The subtle variation between cells will further challenge the performance of the tools in the presence of known ground-truth.

October:

- Write up the thesis (due: Mid November)

November:

- Refine the thesis (due: Mid November)
- Prepare final presentation (due: Mid-late November)

References

1. Fang, R. *et al.* Fast and Accurate Clustering of Single Cell Epigenomes Reveals Cis-Regulatory Elements in Rare Cell Types. *bioRxiv,* 1–41 (2019).
2. Yu, W., Uzun, Y., Zhu, Q., Chen, C. & Tan, K. scATAC-pro: a comprehensive workbench for single-cell chromatin accessibility sequencing data. *bioRxiv,* 824326 (Oct. 2019).
3. Chen, H. *et al.* Assessment of computational methods for the analysis of single-cell ATAC-seq data. *bioRxiv,* 739011 (Aug. 2019).
4. Su, S. *et al.* CellBench: R/Bioconductor software for comparing single-cell RNA-seq analysis methods. doi:10.1093/bioinformatics/btz889. https://academic.oup.com/bioinformatics/advance-article-abstract/doi/10.1093/bioinformatics/btz889/5645177.
5. Kang, H. M. *et al.* Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36,** 89–94. ISSN: 15461696 (Jan. 2018).
6. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177,** 1888–1902. ISSN: 10974172 (2019).
7. Li, B. *et al.* APEC: An accesson-based method for single-cell chromatin accessibility analysis. *Genome Biology* **21,** 116. ISSN: 1474760X (May 2020).
8. Prompsy, P., Kirchmeier, P. & Vallot, C. ChromSCape : a Shiny/R application for interactive analysis of single-cell chromatin profiles. *bioRxiv,* 683037 (July 2019).

9. Pliner, H. A., Shendure, J. & Trapnell, C. Supervised classification enables rapid annotation of cell atlases. *Nature Methods* **16,** 983–986. ISSN: 15487105 (Oct. 2019).

10. Chen, H. *et al.* Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nature Communications* **10,** 1–14. ISSN: 20411723 (Dec. 2019).

11. Lin, L. & Zhang, L. Single-cell ATAC-seq clustering and differential analysis by convolution-based approach. *bioRxiv,* 2020.02.13.947242 (Feb. 2020).

12. Pliner, H. A. *et al.* Cicero Predicts cis-Regulatory DNA Interactions from Single-Cell Chromatin Accessibility Data. *Molecular Cell* **71,** 858–871. ISSN: 10974164 (2018).

13. González-Blas, C. B. *et al.* Cis-topic modelling of single-cell epigenomes. *bioRxiv,* 370346 (2018).

14. De Boer, C. G. & Regev, A. BROCKMAN: Deciphering variance in epigenomic regulators by k-mer factorization. *BMC Bioinformatics* **19,** 1–13. ISSN: 14712105 (2018).

15. Urrutia, E., Chen, L., Zhou, H. & Jiang, Y. Destin: toolkit for single-cell analysis of chromatin accessibility. *Bioinformatics.* ISSN: 1367-4803. doi:10.1093/bioinformatics/btz141 (Mar. 2019).

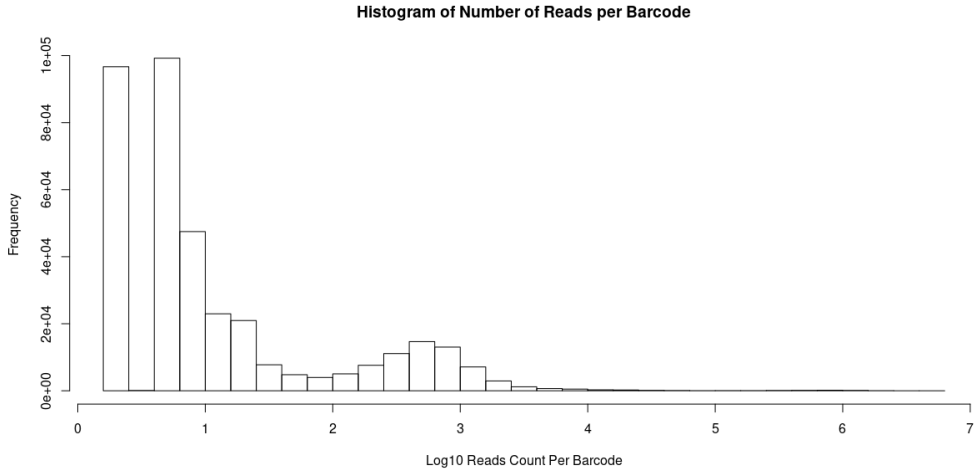
16. Zamanighomi, M. *et al.* Unsupervised clustering and epigenetic classification of single cells. *Nature Communications* **9,** 1–8. ISSN: 20411723 (2018).

17. 10x Genomics. *Chromium Single Cell ATAC Reagent Kits* https://support.10xgenomics.com/single-cell-atac/library-prep/doc/user-guide-chromium-single-cell-atac-reagent-kits-user-guide-v1-chemistry.

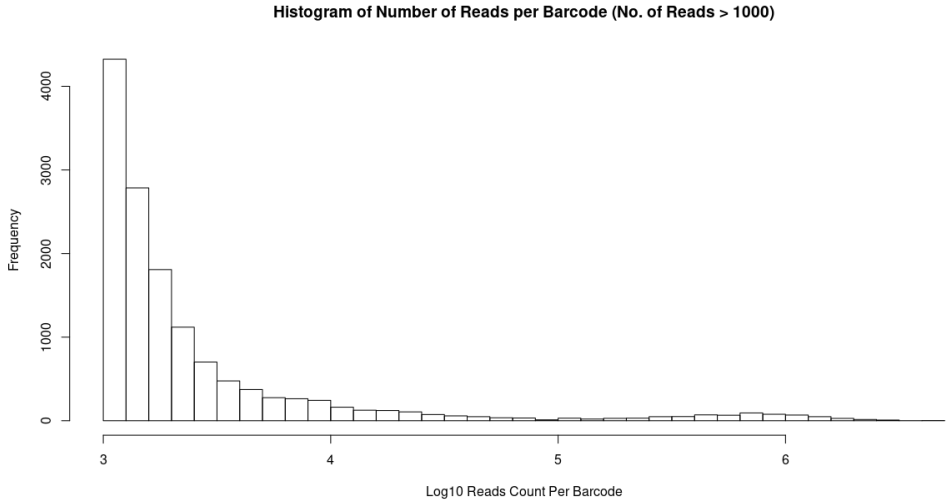
18. Hubert, L. & Arabie, P. Comparing partitions. *Journal of Classification* **2,** 193–218. ISSN: 01764268 (Dec. 1985).

19. Schep, A. N., Wu, B., Buenrostro, J. D. & Greenleaf, W. J. ChromVAR: Inferring transcription-factor-associated accessibility from single-cell epigenomic data. *Nature Methods* **14,** 975–978. ISSN: 15487105 (2017).

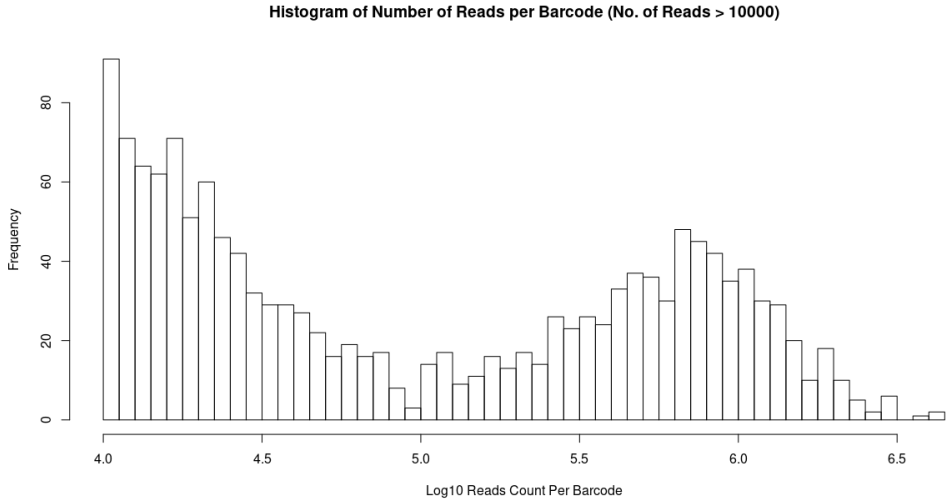
20. Satpathy, A. T. *et al.* Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral T cell exhaustion. *Nature Biotechnology* **37,** 925–936. ISSN: 1087-0156 (2019).



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

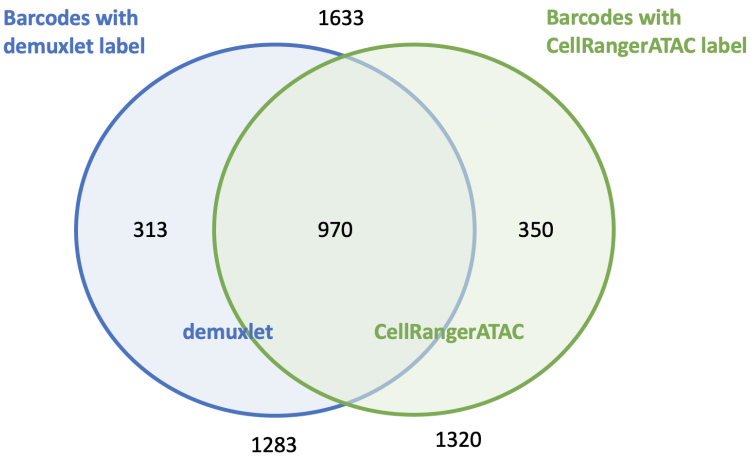


(b) Distribution of read-counts in barcodes with more than 1,000 reads.

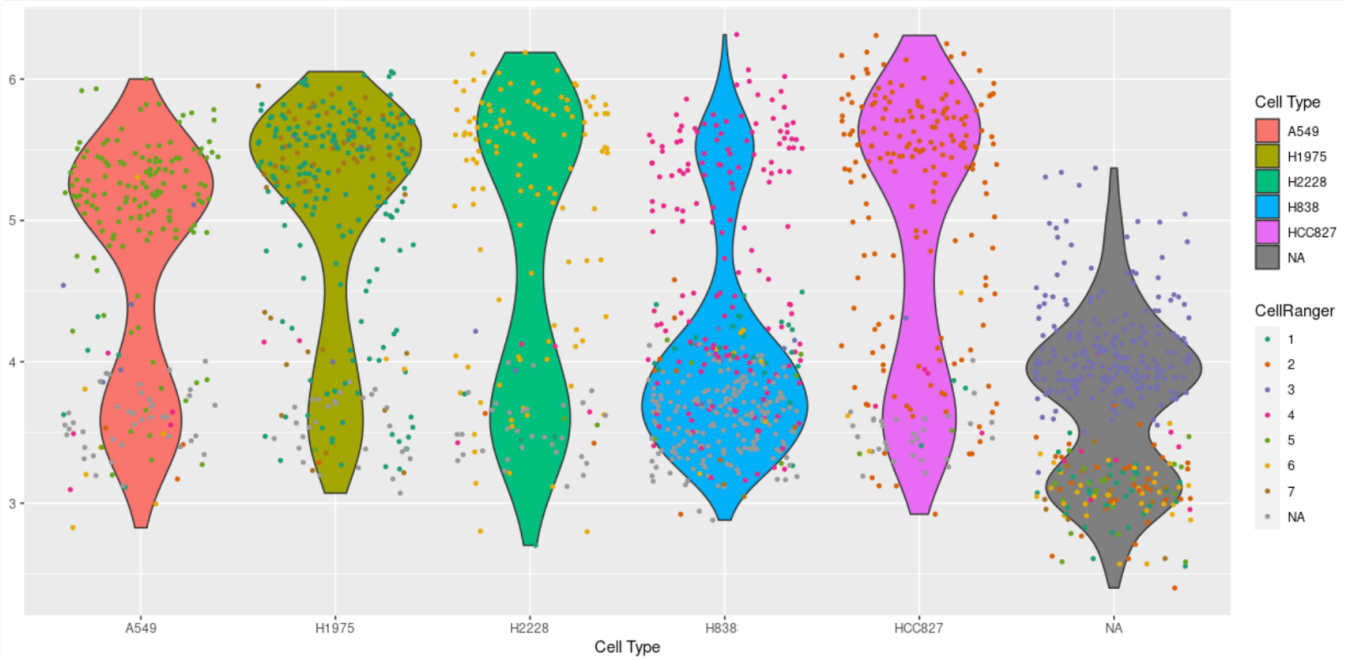


(c) Distribution of read-counts in barcodes with more than 10,000 reads.

Figure 4: **Distribution of read-counts of barcodes in `bam` file.**



(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 5: **Relationship between `CellRanger ATAC` determined cell barcodes and `demuxlet` labelled barcodes.**

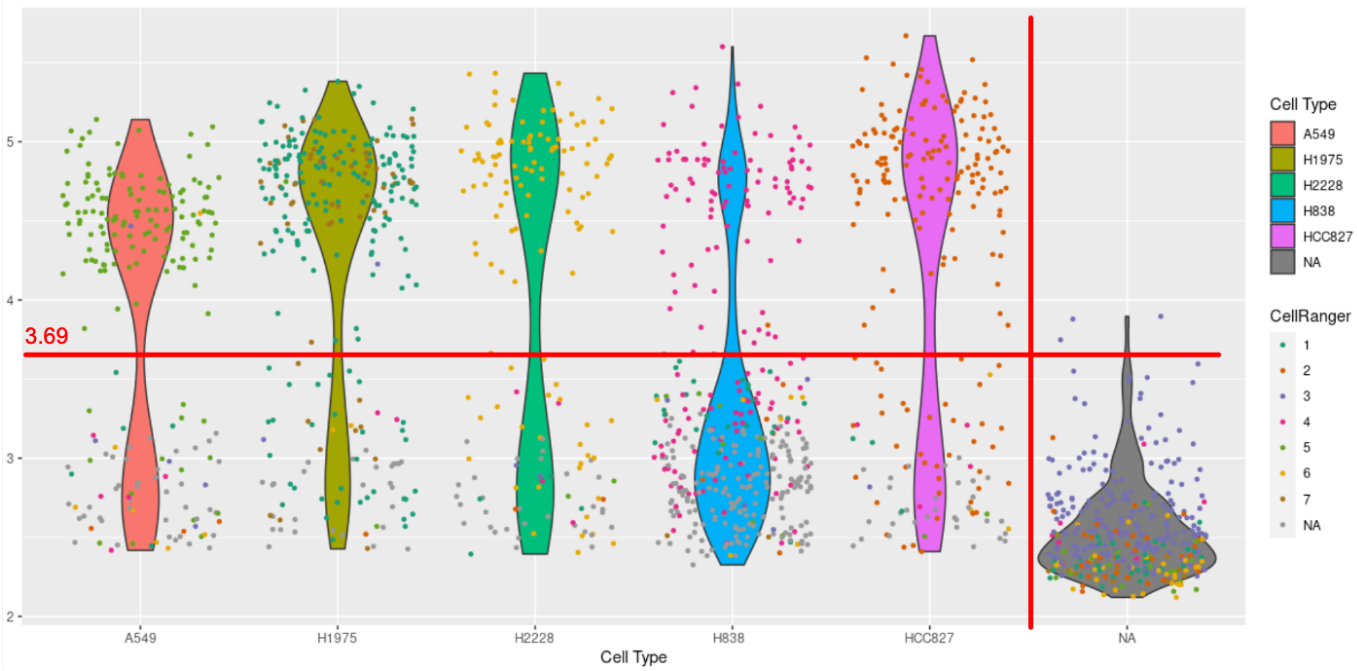


Figure 6: **Filter 1: filter of barcodes based on $\log_{10}$ of the number of high quality unique fragments per barcode for each cell type.**

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.

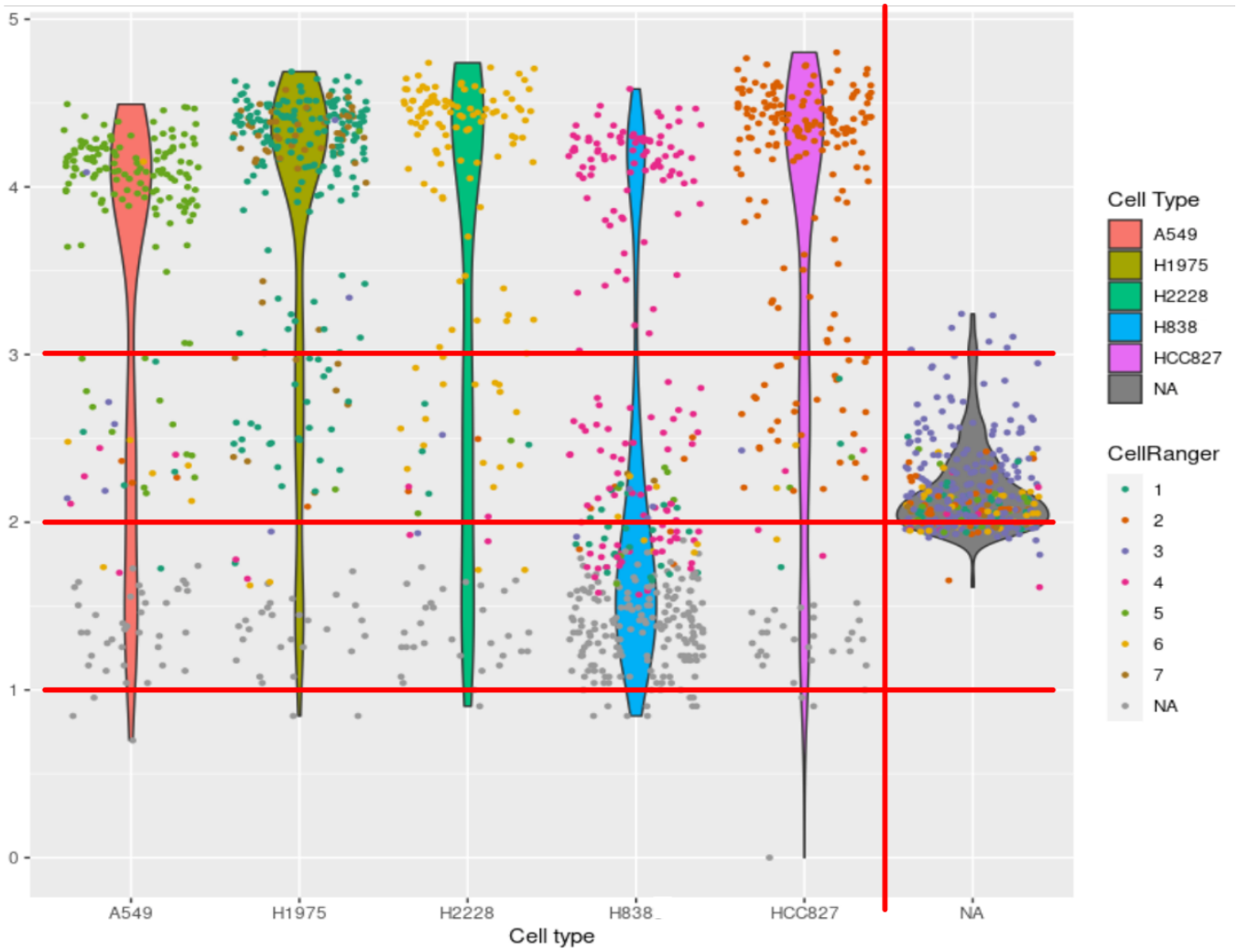


Figure 7: **Filter 2: filtering barcodes based on $\log_{10}$ of the binary library size.**
The plot shows the distribution of $\log_{10}$(read-pair numbers) in y-axis for each cell type (x-axis). Each violin plot is coloured according to cell types with grey indicating barcodes not labelled by `demuxlet`. Each dot represents a cell (i.e. barcode). The dots are coloured by `CellRanger 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.

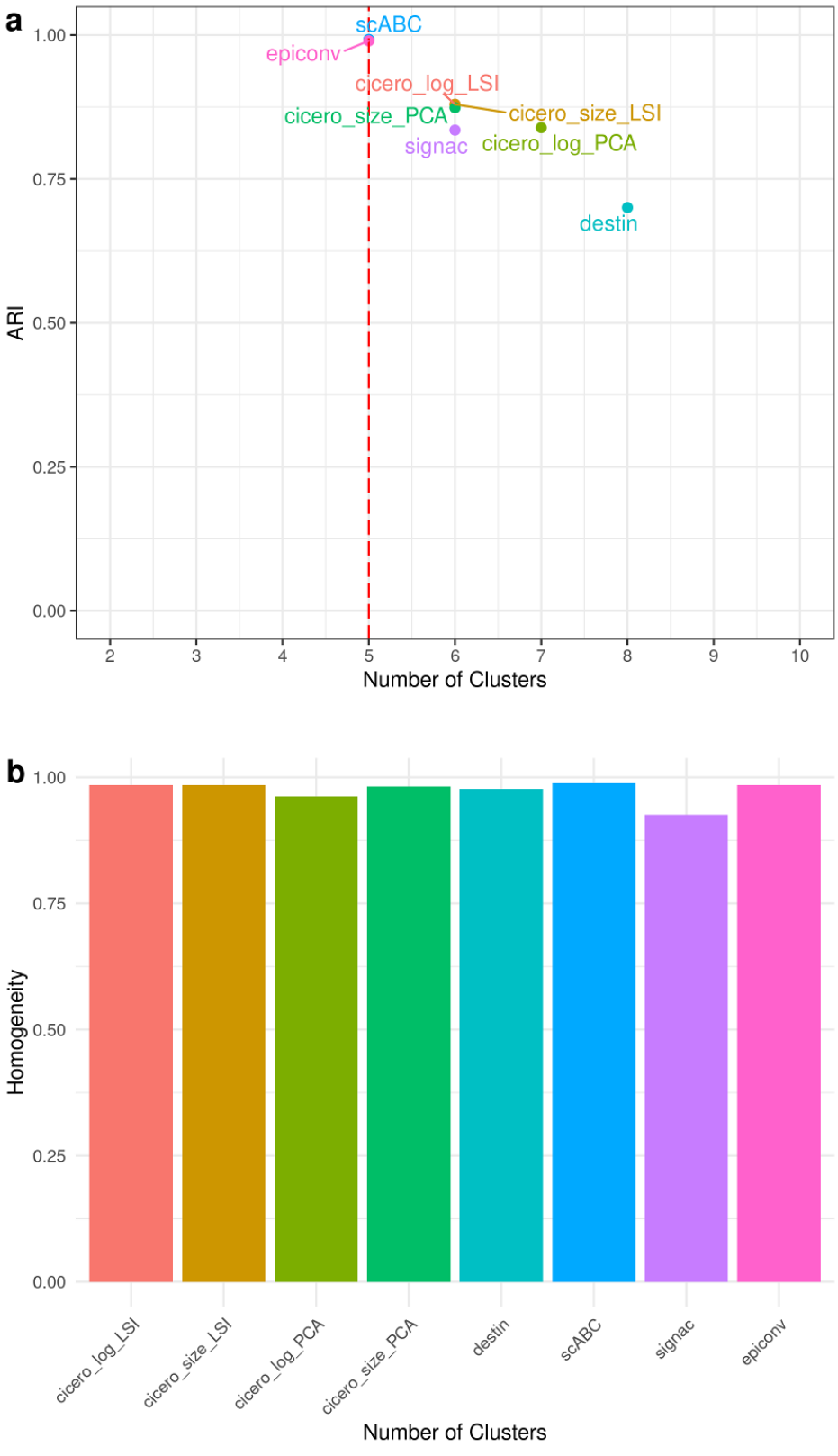


Figure 8: **Evaluation of the clustering step of five selected scTAC-seq data analysis tools.** Two normalisation methods (log or size only) and two dimension reduction methods (LSI or PCA) were applied for `Cicero`. (a) the Adjusted Rand Index (*ARI*) and number of clusters generated for each clustering method. (b) the homogeneity of the clusters generated by each tool.

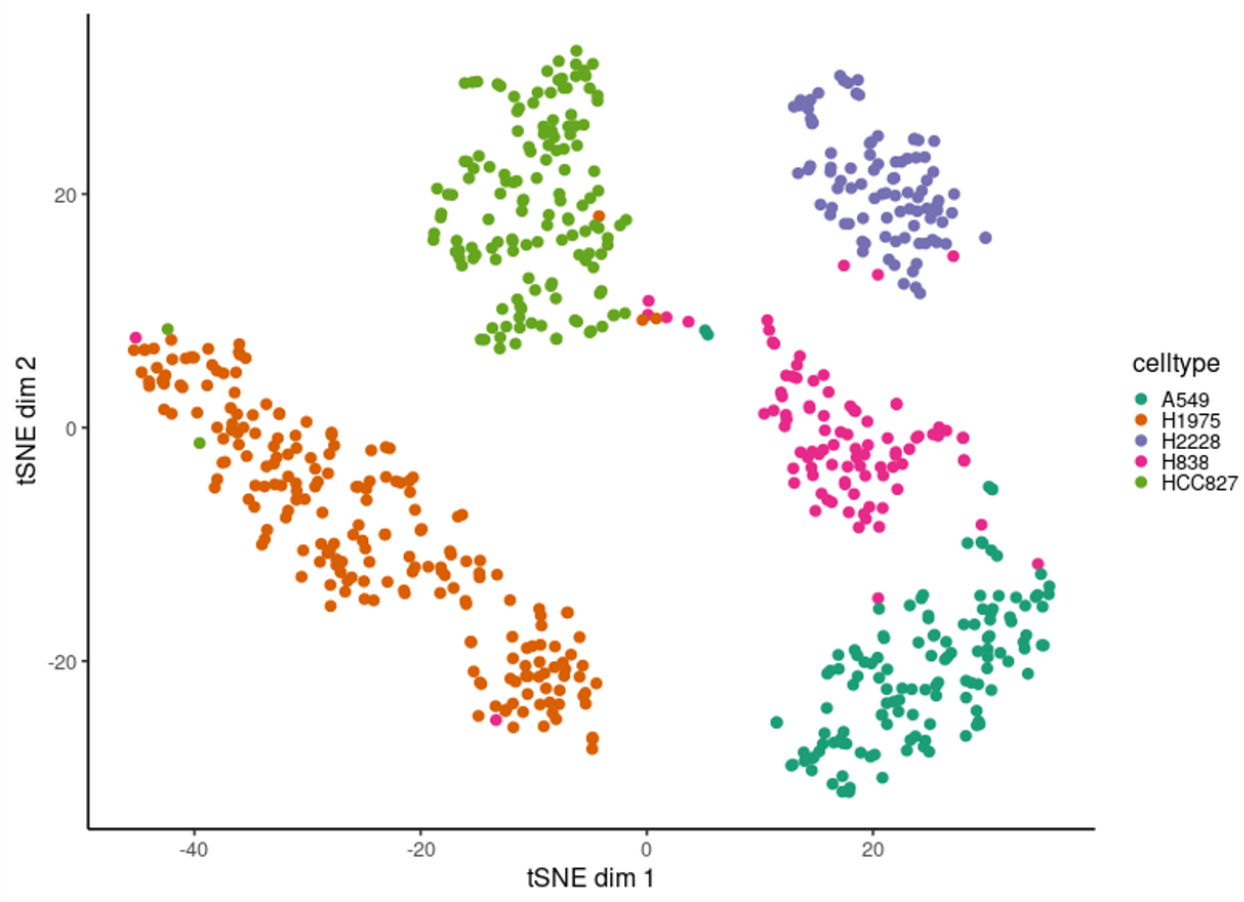


Figure 9: **The tSNE plot of the** *Filter 1* **cleaned dataset, generated by the** `chromVar` **[19].**

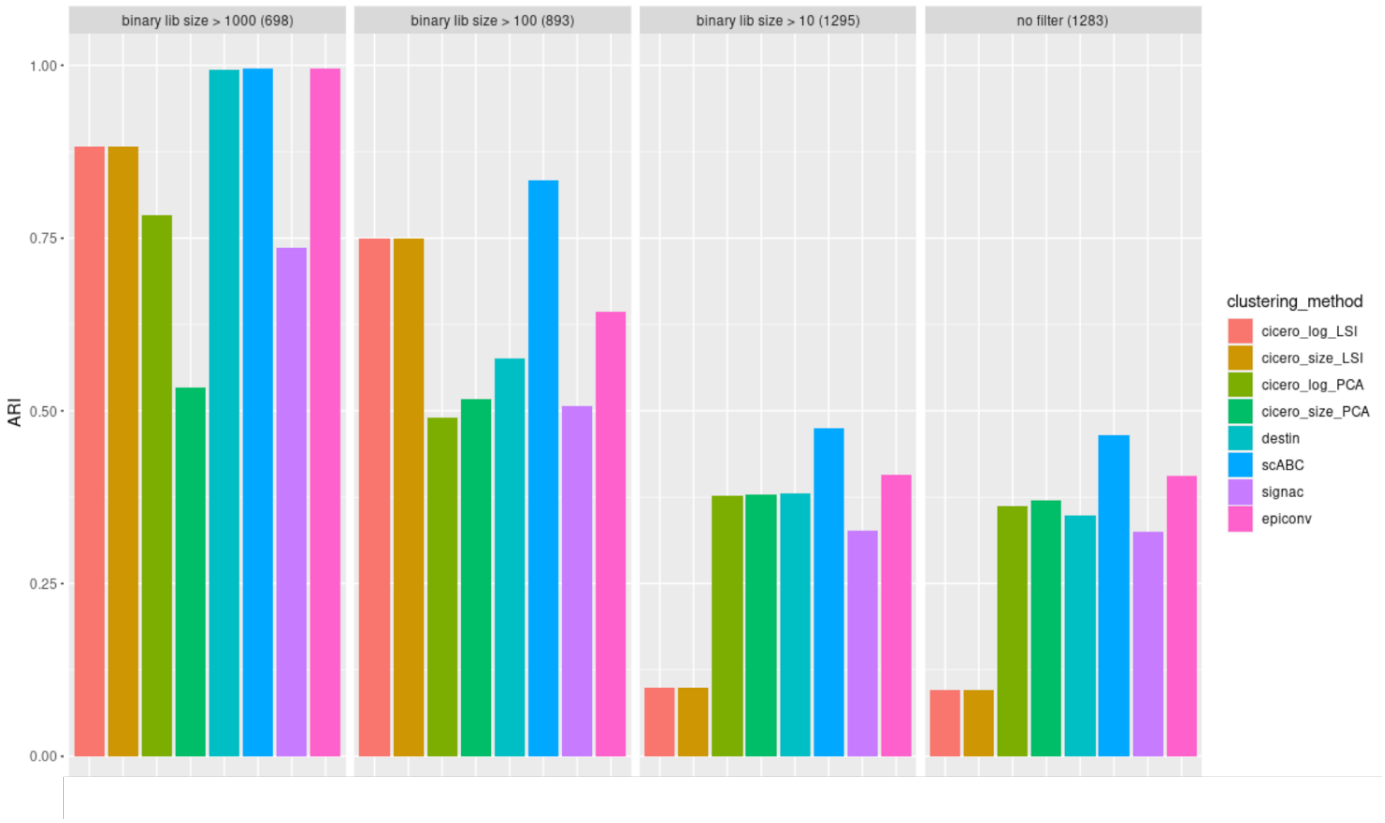


Figure 10: **Performance of clustering methods (*ARI*) on variable data qualities.**

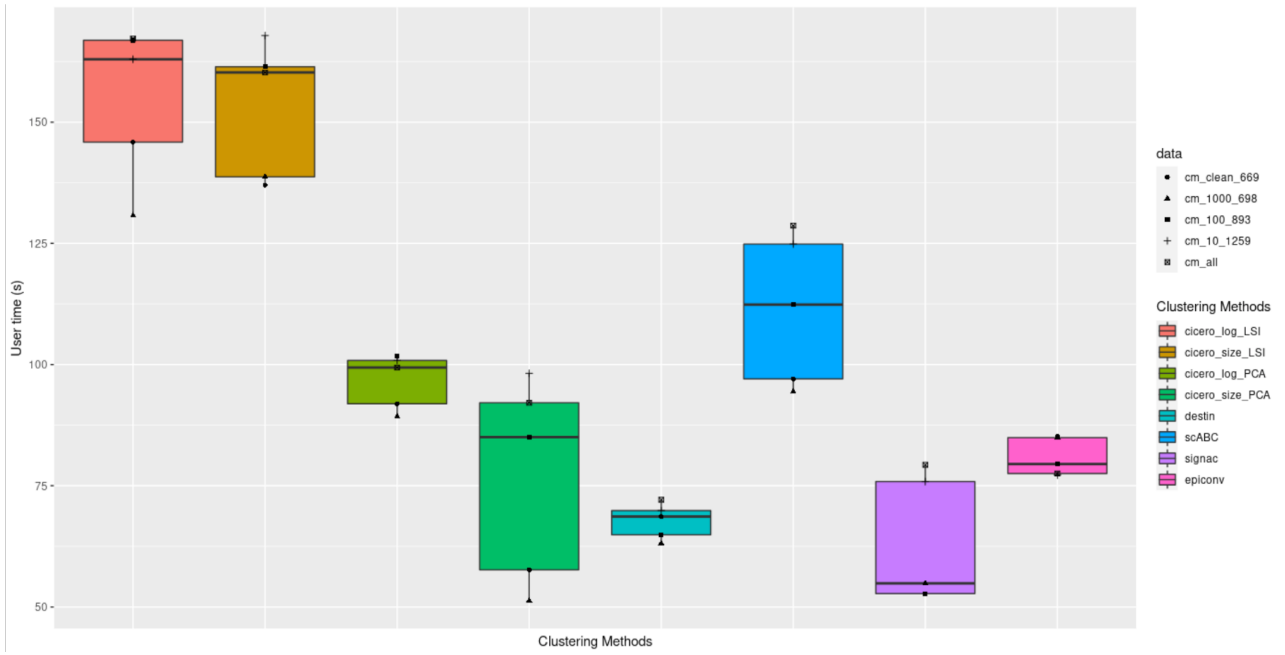


Figure 11: **The time consumption of the clustering methods.**

The shape of the point indicates the different datasets (more or less stringent barcode filtering). The colour indicates different clustering methods.

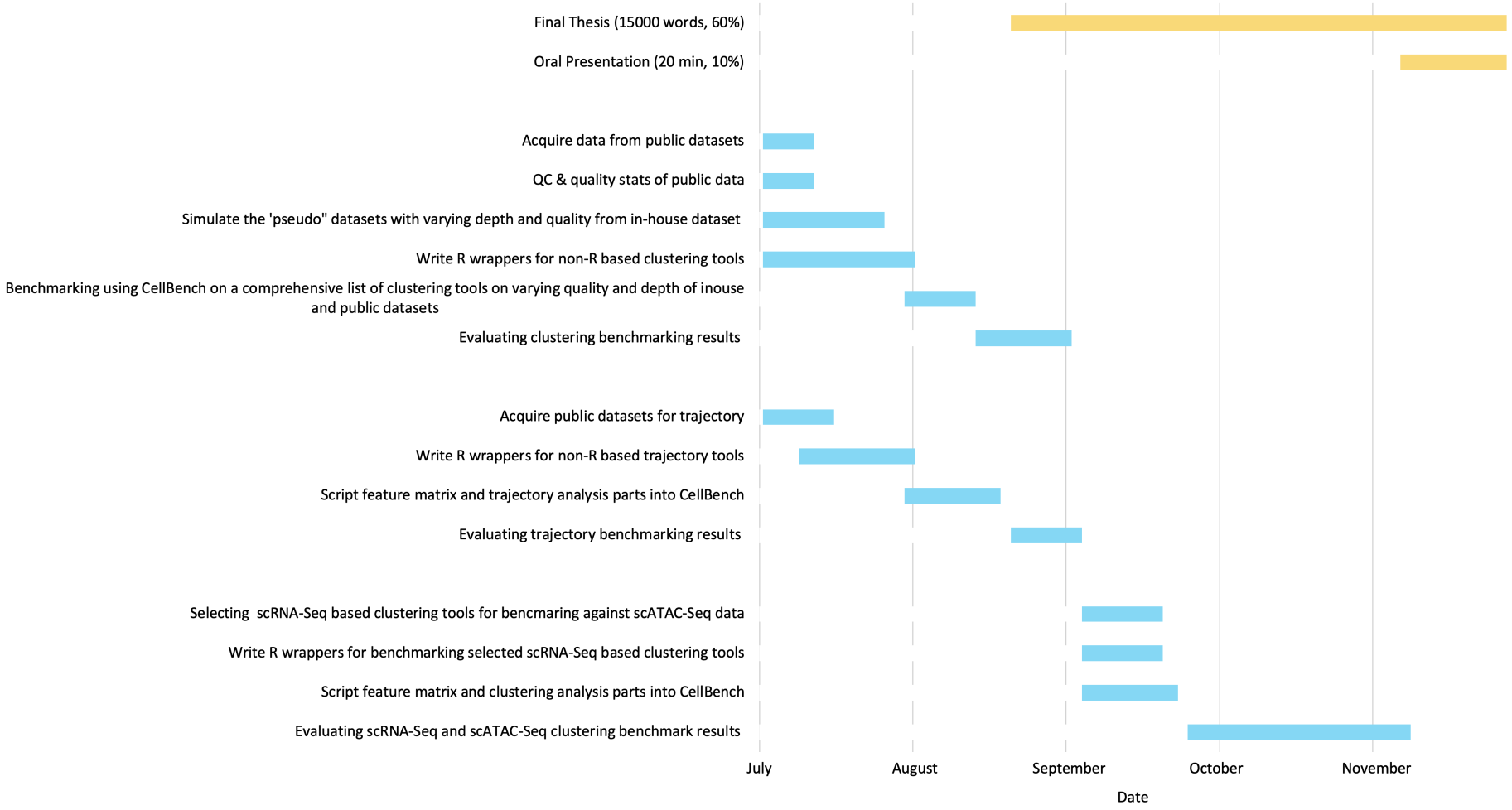


Figure 12: **The timeline for research in semester 2, 2020.**