Study of the regulatory programs of spermatogenesis through the integration of single-cell RNA and ATAC

Projects in Bioinformatics - Fall 2025

Johan Olesen
202104408
Msc. Student Bioinformatics

Samuele Soraggi
Supervisor
Special consultant, Bioinformatics Research Center, Aarhus University

12.01.2026

Contents

1 Introduction

Spermatogenesis is a complex process that permits the differentiation of stem cells into mature spermatozoa, and is of high relevance in studying infertility conditions and cross-species differences in the biological processes.

1.1 Goals for the project

Initial:

- learn basics of git
- learn sc workflow with scanpy, muon and scvi-tools
- work with real messy data
- Answer:
  - Cell states & trajectories: Can we recover a clean spermatogenic trajectory (spermatogonia → spermatocytes → spermatids) and supporting somatic lineages?
  - Peak→gene linkage: Which distal elements likely regulate stage-specific genes?
  - TF programs: Which TFs show coordinated motif accessibility + target expression? (e.g., STRA8, A-MYB, TAF7L)

Actually done:

- learn basics of git
- learn how to do Conda environment reproducibility
- small shell scripts to run on SLURM.
- learn single cell workflow with scanpy, muon and scvi-tools
- work with real messy data
- Answer: Cell states & trajectories: Can we recover a clean spermatogenic trajectory (spermatogonia → spermatocytes → spermatids) and supporting somatic lineages?
- Succefull celltype annotation of both scRNA-seq and scATAC-seq.
- Cell topics and DARs for scATAC-seq for further analysis

1.1.1 Data availability

All environment files, notebooks and scripts as well as the repport files for this project are available on a public github repository: SamueleSoraggi/PIB-johan-olesen.

1.2 Workflow Overview

The report is divided into three stages each with individual goals showing the different stages of the project. Stage 1 being the preliminary stage of aquiring data and getting it into a format where analysis can be begun. Stage 1 is described in **Figure 1**.

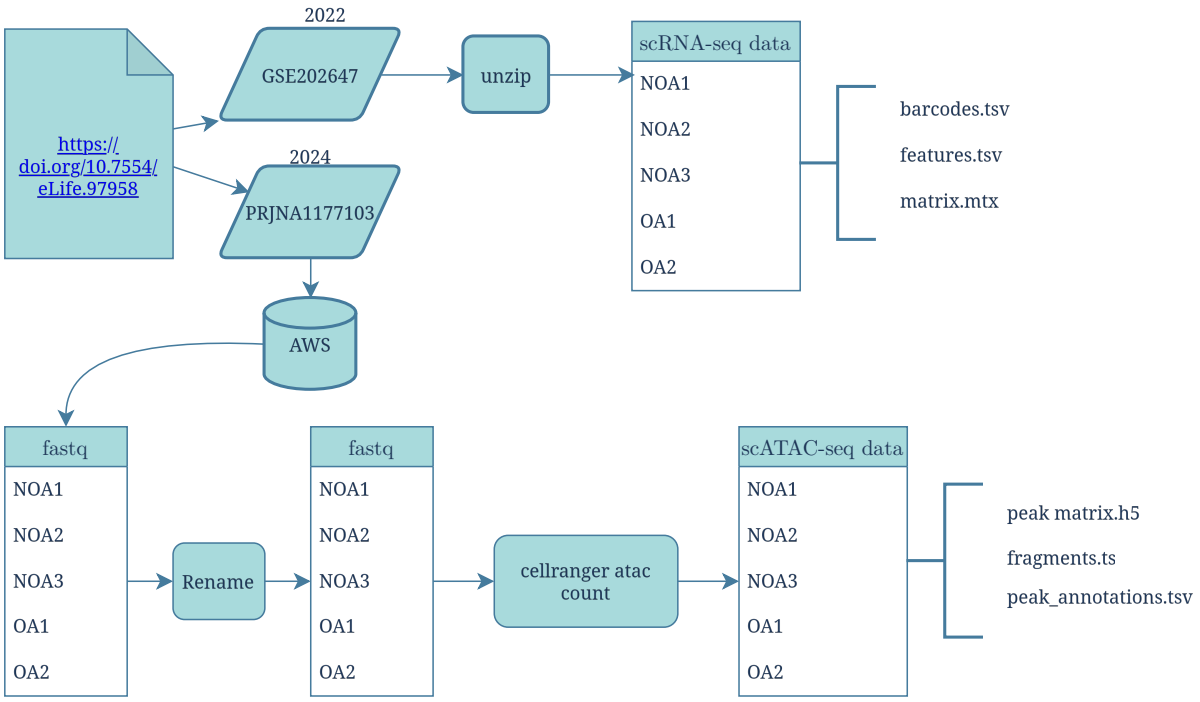


Figure 1: Stage 1 schematic of data aquisition and preparation.

The goal of stage 2 is to annotate the scRNA-seq data with celltypes. It is described in **Figure 2**.

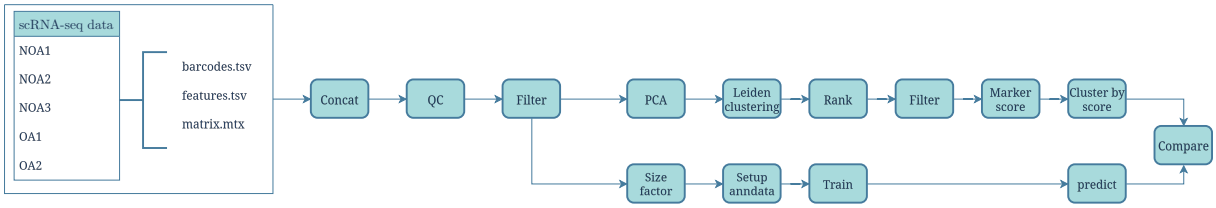


Figure 2: Stage 2 schematic of scRNA-seq celltype annotation with a *semi-manual* way and CellAssign model.

Stage 3 is about celltype annotation of the scATAC-seq data.

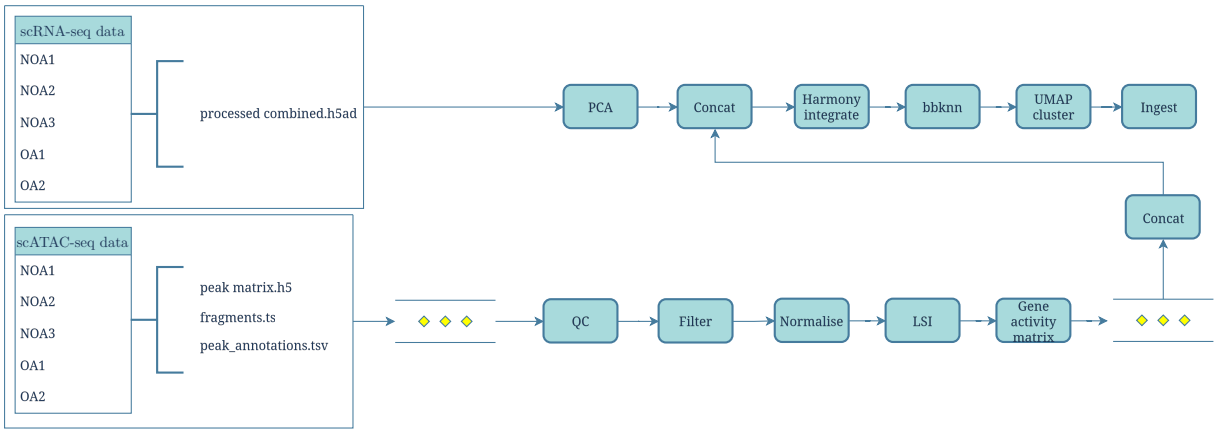


Figure 3: Stage 3a schematic of attempt at label transfer by integrating scRNA-seq and scATAC-seq.

Flowcharts drawn with `draw.io`.

Stage 3 was supposed to continue from after the celltype annotation to answer the other two research question, but did not due to ime constraints.

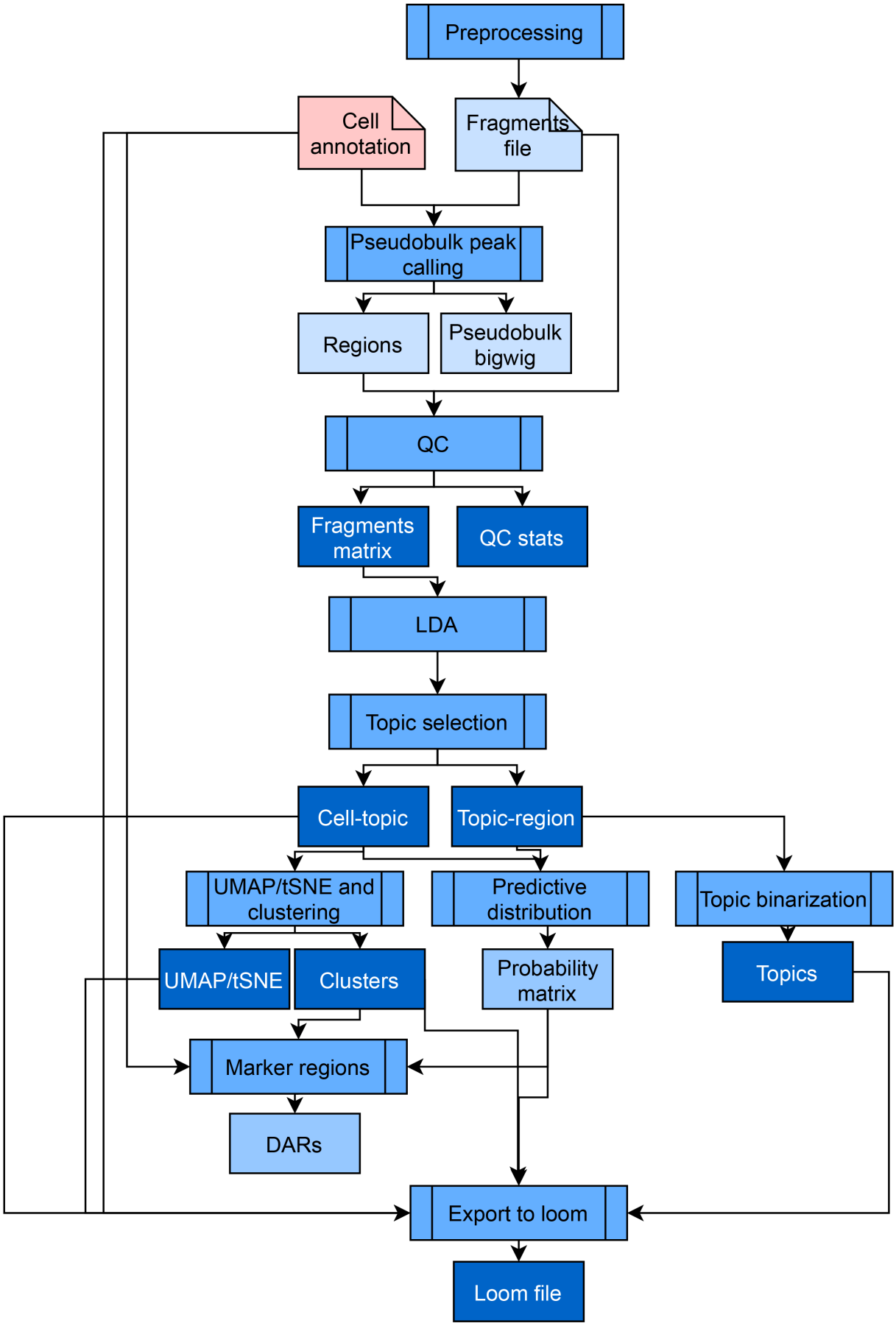


Figure 4: Stage 3b schematic of pycistopic workflow from PyCistopic documentation. [1]

1.2.1 Environment setup with Conda

First step was to get a working environment setup for the analyses. For this Conda was used to create a environment with the required packages, relying on `pip` for the most up-to-date packages.

For the tutorial run and scRNA-seq labelling the environmnet `torch_env.yml` [2] was used. This environment includes the scverse's anndata [3], mudata [4], scanpy [5], muon [6] and scvi-tools [7] packages, as well as full PyTorch [8] CUDA capabilities for scvi-tools.

For the second scATAC-seq workflow another environment was used because of versioning requirements; `cistopic_env.yml`, consisting of the SCENIC+ [9] suite.

1.2.2 Tutorial run

To start of before real testis data had been found. A quick run through of the tutorial run of multiome 10X PBMC [10] by Bredikin to get a quick overview of how to work with single cell data and anndata objects, and to check Conda environment worked.

Was succesful in creating the same analysis as the tutorial.

2 Stage 1: Data aquisition and preparation

For the first stage of the project, we will be focusing on acquiring and preparing data for analysis. This includes downloading and organizing the necessary datasets, as well as preprocessing the data to ensure it is ready for downstream analysis. The workflow is illustrated on **Figure 1**.

For the real dataset set out in the goals, Wang et al [11] have made their scRNA-seq and scATAC-seq data available.

The dataset consists of scRNA-seq from 2022 of 5 donors; NOA1, NOA2, NOA3, OA1 and OA2. NOA stands for non-obstructive azoospermia, OA for obstructive azoospermia.

The scRNA-seq data was available under NCBI Gene Expression Omnibus ID **GSE202647** from 2022, and was already ready for analysis.

The scATAC-seq data was available under NCBI BioProject ID **PRJNA1177103** from 2024. This was only the raw read data, so the Cellranger ATAC [12] pipeline was run for each of the five samples. For Cellranger to be able to run the data had to be structured and named in a specific way. Using the guidelines available from the offical 10X Cellranger documentation and matching the read length of each of the four files per sample was renamed accordingly.

Identifying each file to do correct renaming was done by looking at the read length. This was needed as the original files were just named numerically. The naming scheme for Cellranger ATAC count [13]: `[Sample Name]S1_L00[Lane Number][Read Type]_001.fastq.gz`, where `Read type`:

- `I1`: Dual index i7 read (optional)
- `R1`: Read 1
- `I2`: Dual index i5 read
- `R3`: Read 2

`Lane Number` does not matter. `Sample Name` can be anything.

In the case of the NOA3 donor the read length was R1; 50, R2; 49, I1; 8, I2; 16. Thereby producing these four files:

- `SRR31097965_S1_L001_I1_001.fastq`
- `SRR31097965_S1_L001_I2_001.fastq`
- `SRR31097965_S1_L001_R1_001.fastq`
- `SRR31097965_S1_L001_R2_001.fastq`

The `count` function was run with Cellranger ATAC version 2.2.0 on each sample with reference data *refdata-cellranger-arc-GRCh38-2024-A* using SLURM. This resulted

in the `filtered_peak_bc_matrix.h5`, `fragments.tsv` and `peak_annotation.tsv` for each sample ready for analysis.

Ideally, the data from scRNA-seq and scATAC-seq should be cell matched from their barcodes, but finding public datasets for testis with that criteria was difficult. Instead we will be matching cells by celltype instead. The data in Wang et al's study comes from the same five donors, but sequenced at seperate time points, so the cell barcodes do not match.

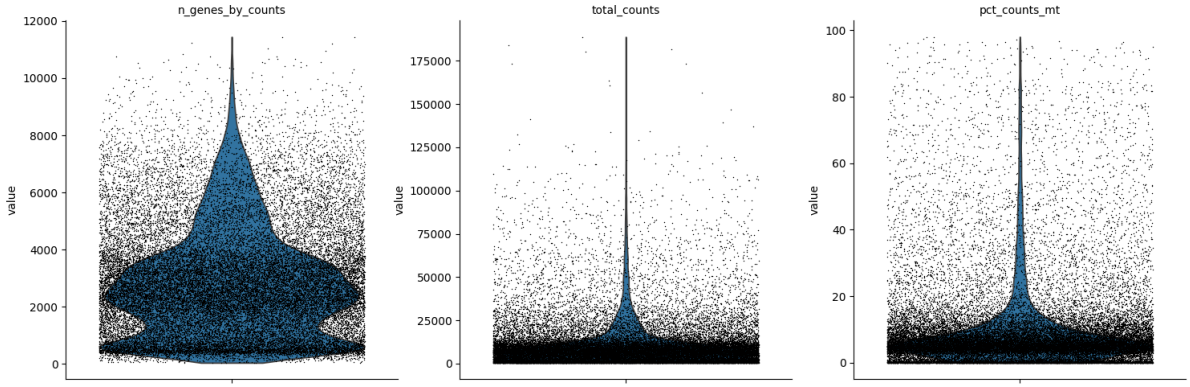
3 Stage 2: Celltype annotation of scRNA-seq data

For stage 2 the main goal is to get the scRNA-seq data processed and annotated with cell types.

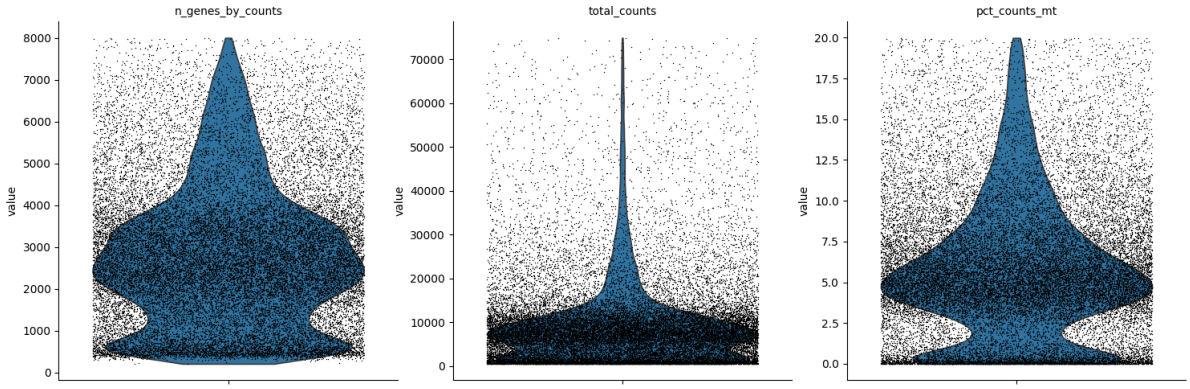
The workflow was done in Jupyter Notebooks in python using the `torch_env.yml` Conda environment. The analysis was mainly done using scanpy plethora of functions.

Firstly, all five dataset was loaded and concatenated into one AnnData object containing all 33683 cells and 27984 features.

Calculating quality control metrics to be able to filter followed. Using gene threshold of presens in minimum of 3 cells, cell threshold of uniquely expressed genes minimum of 300 genes and maximum of 8000, percentage of mitochondrial genes to be less than 20% and total counts of less than 75000. This yielded 28750 cells and 25464 features.



(a) Before filtering



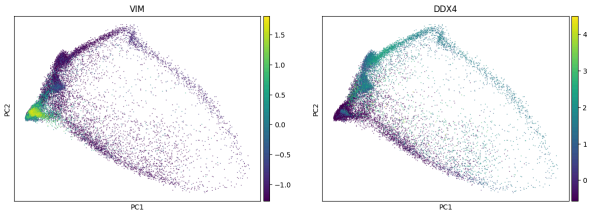
(b) After filtering

Figure 5: Quality control metrics of cells

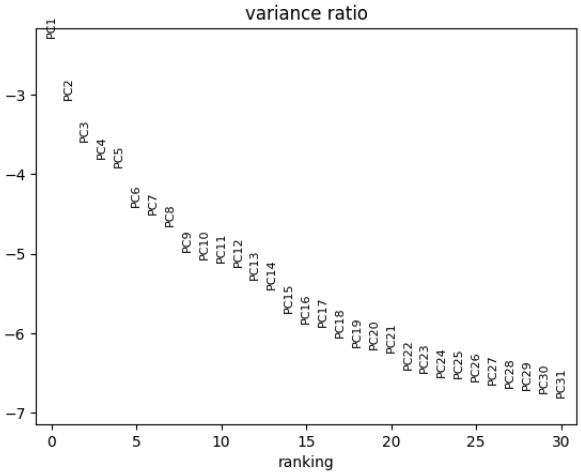
3.1 Clustering

Data was prepared for PCA and PCA was run using the arpack solver. As shown in **Figure 6a** the PCA transformation is able to distinguish between general cell lineages well. Now using the transformed data it was clustered using the leiden algorhitm using the top 25 principle components and 10 neighbours with a resolution of 0.5. This

yielded **Figure 7a**. Looking at each cluster and seeing which genes are differentially expressed compared to the other clusters, **Figure 7b**, show clusters 19 and 20 being potentially troublesome. Cluster 19 shows many ribosomal markers: *RPL10*, *RPL41*, *RPL39*, *RPS29*, *RPS15A*, *RPS27*, *RPS14*, *RPS3*, *RPL28*. This indicates potentially damaged cells so were filtered out. Cluster 20 shows very low differientation score and the genes highlighted are various tissue origins, so was also filtered out.

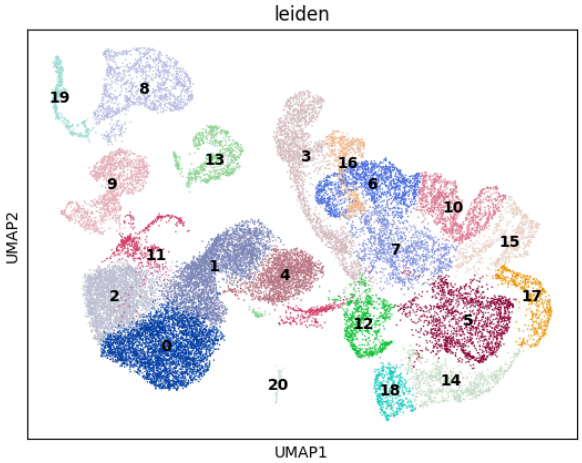


(a) PCA plot of *VIM* and *DDX4* expression in cells. *VIM* indicates somatic lineages. *DDX4* indicates germ cell lineages.

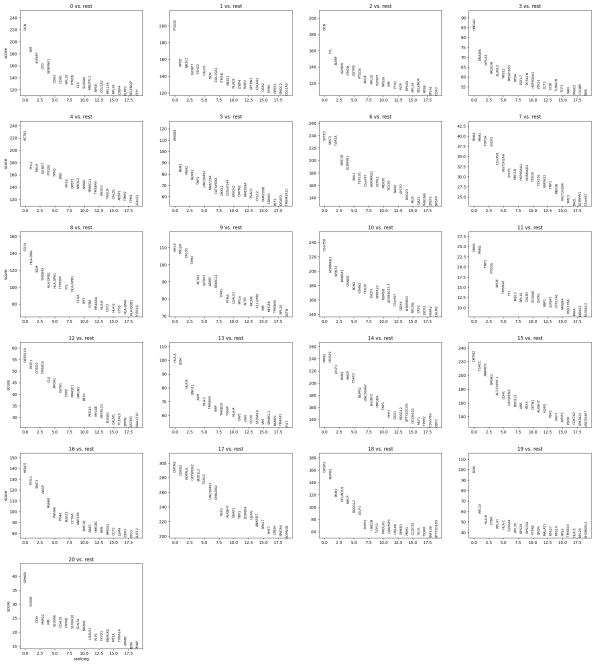


(b) PCA variance ratio ranking.

Figure 6: PCA associated plots



(a) Initial leiden clustering of cells.



(b) Top 10 ranked differentially expressed genes in each cluster compared to other clusters.

Figure 7:

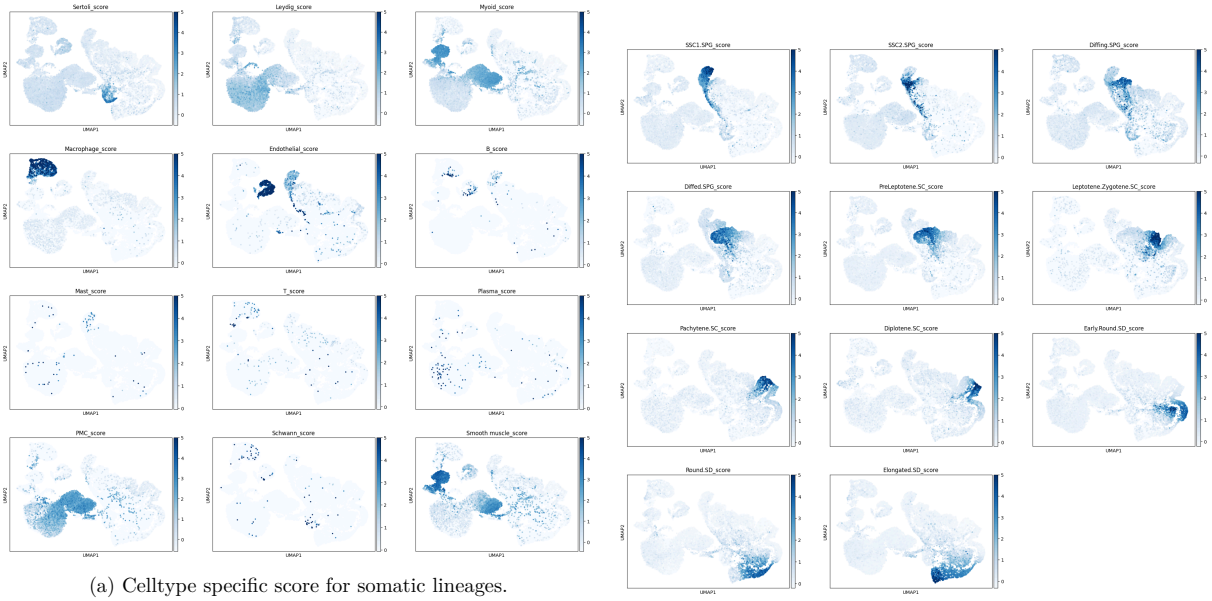
3.2 Celltype annotation

To find the celltype for each cluster a list of marker genes for the different expected cell types was produced in `celltype_markers.csv` and **Table 1**. The marker genes were primarily sourced from Wang et al [11]. Of note is that many more marker genes are present for the germ cell lineages as they were the main focus. To link each cell to a celltype, they were each scored using a custom scoring function. Each cluster was then assigned the highest mean score across the cells in the cluster. It works by taking the list of marker genes associated with each cell type and comparing the expression of the marker genes with 100 random genes. A higher score therefore means higher expression of associated genes indicating how well each cell matches the celltype. `marker_score` and `clustersByScores` were taken from NGS Summer 2025 course [14].

The scores for all the different celltypes can be seen in **Figure 8**. As expected we find the germline cells clusters together away from the somatic cells. More interestingly, the germline clusters form a spermatid continous cluster through the different developmental stages of germ cells, with overlaps of the celltype score in the transition part of the clusters. Starting in the top left all the way to the bottom left; SSC1 spermatogonia → SSC2 spermatogonia → Diffing spermatogonia → Diffed spermatogonia → Pre-leptogene spermatocytes → Leptogene-zygotene spermatocytes → Pachytene spermatocytes → Diplotene spermatocytes → Early round spermatid → Round spermatid → Elongated spermatid.

The somatic cells are mostly clustered in very seperated clusters, with sertoli, macrophages, and endothelial being clearly defined. Myoid, PMC, leydig and smooth muscle blends more together having big overlaps.

Figure 8 completes the first biological goal for the project.



(a) Celltype specific score for somatic lineages.

(b) for germline lineages. *SPG*: Spermatogonia, *SC*: Spermatocytes, *SD*: Spermatid.

Figure 8: Celltype specific score.

As shown in **Figure 9**, each germline celltype corresponds strongly to one cluster and slightly less to another one or two clusters. This is precisely what is shown on **Figure 8b** were there is slight overlap from one stage to another.

3.3 SCVI-tools CellAssign

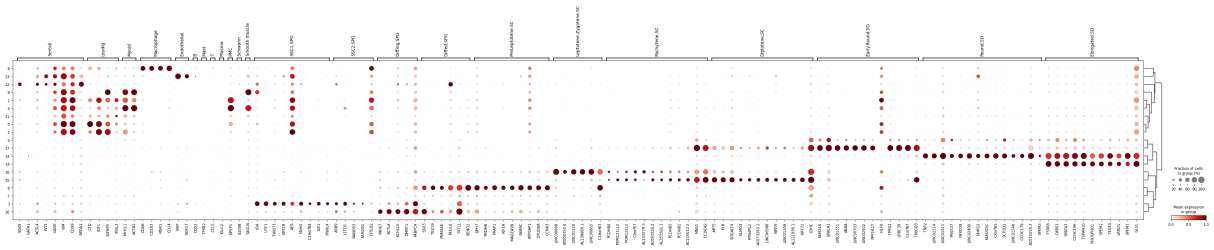


Figure 9: Dotplot showing the expression of marker genes across the leiden clusters.

So far the analysis has been semi manual in deciding celltypes by using the simple but effective scoring function. If instead use scvi-tools to make a model to assign celltype.

First, the AnnData object is prepared by using the raw data along with a library size factor and celltype markers. A SCVI CellAssign model is then setup and trained. Using the model to predict probability of each each being a specific celltype. Each cell was then assigned the celltype with highest probability. The predicted values for each cell is visualised on **Figure 10**. A majority of cells have probability of 1, with a small amount being more uncertain. Interestingly, compared to the scoring function, the predictions do not include any smooth muscle cells.

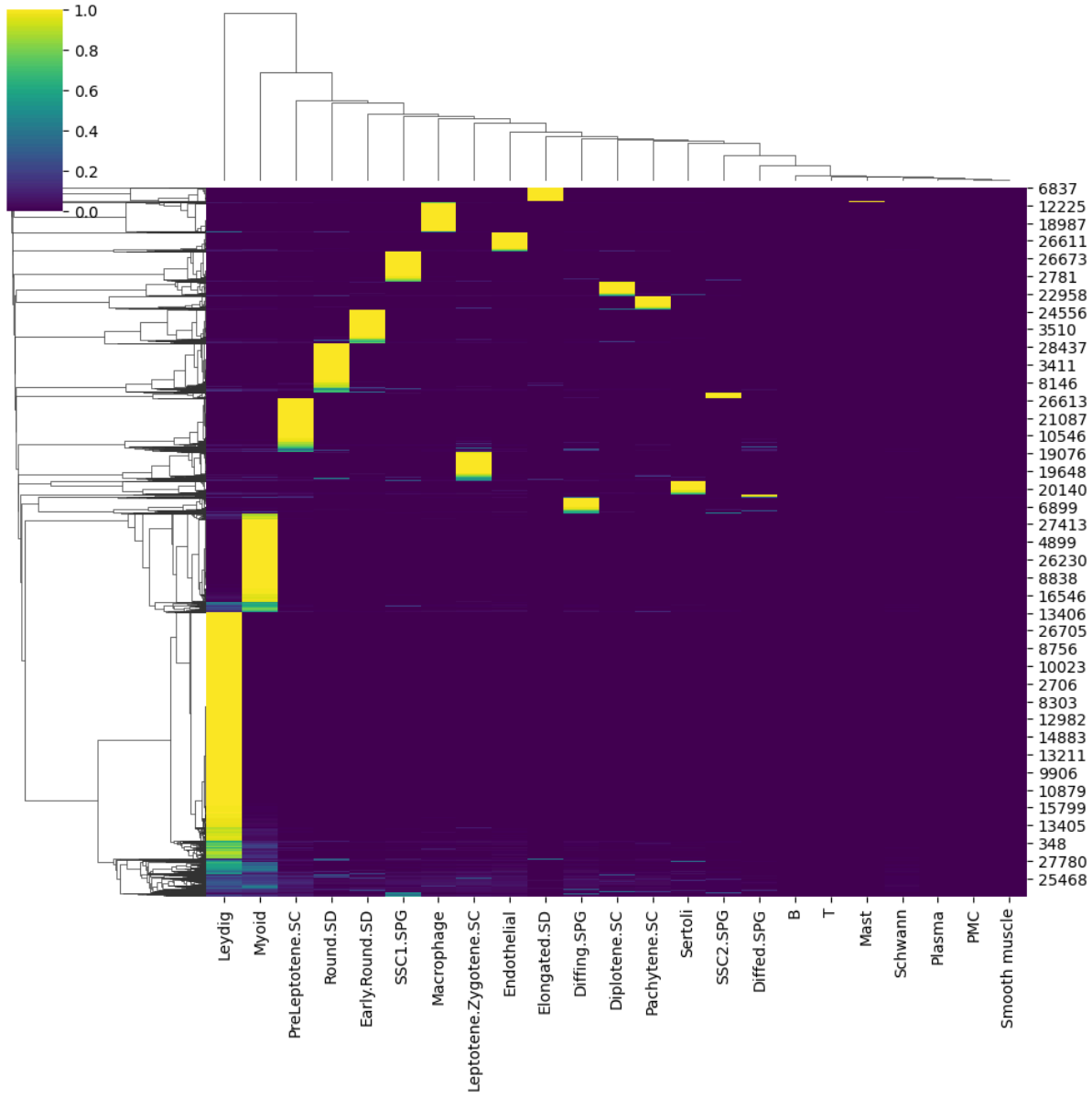


Figure 10: Clustermap of CellAssign model predictions. Colour indicating probability, lighter more likely.

Comparing the two methods reveals that beside the absence of smooth muscle cells in using CellAssign, the two methods yields similar results as seen on **Figure 11**. Instead likely having been assigned to leydig or myoid cells. Finally, as the two methods yielded similar results, the leiden cluster assigned would be the final celltypes.

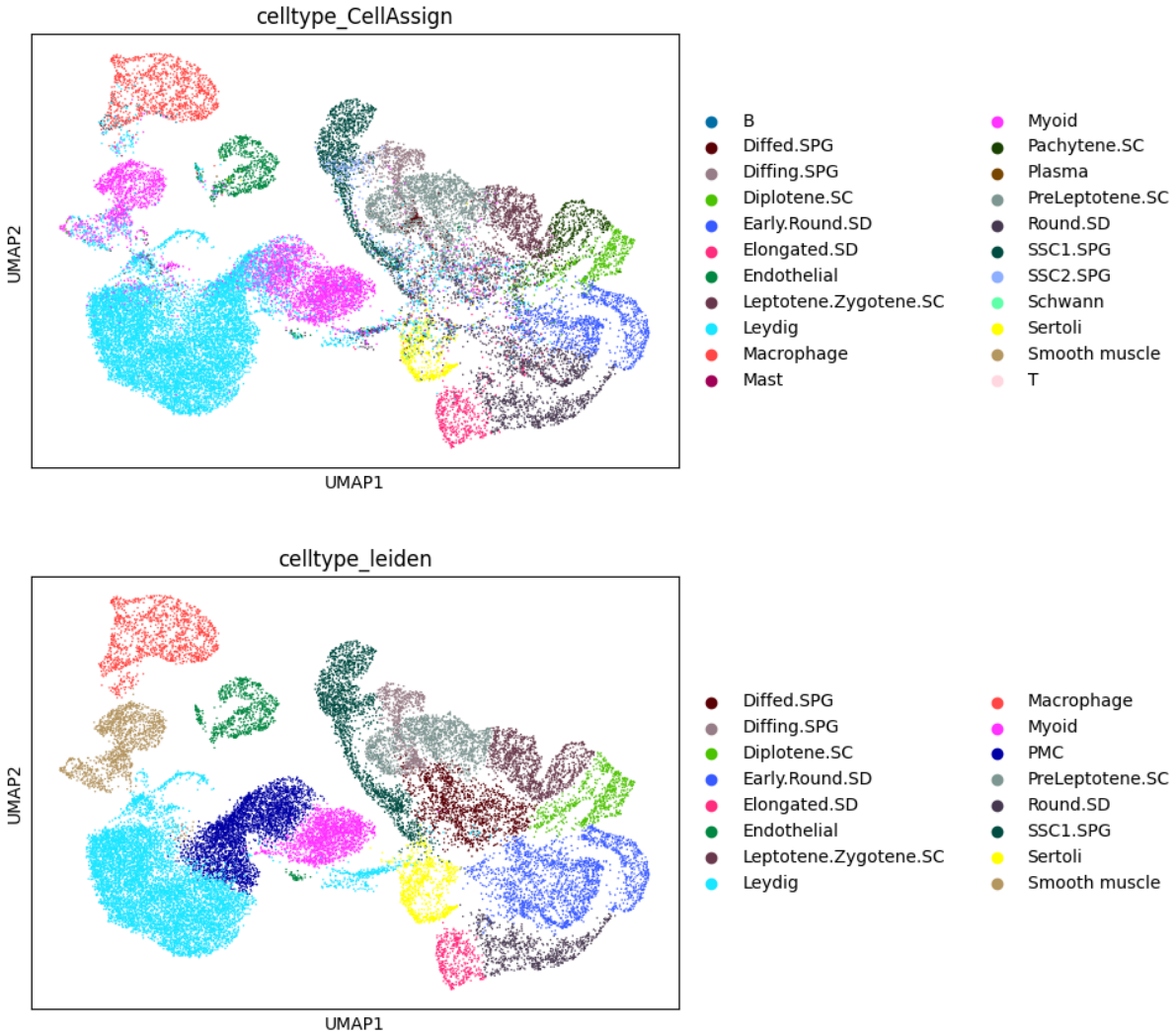


Figure 11: Celltype annotation result using either CellAssign model or leiden clustering using the same set of marker genes. *SPG*: Spermatogonia, *SC*: Spermatocytes, *SD*: Spermatid.

4 Stage 3a: Celltype annotation of scATAC-seq data with label transfer

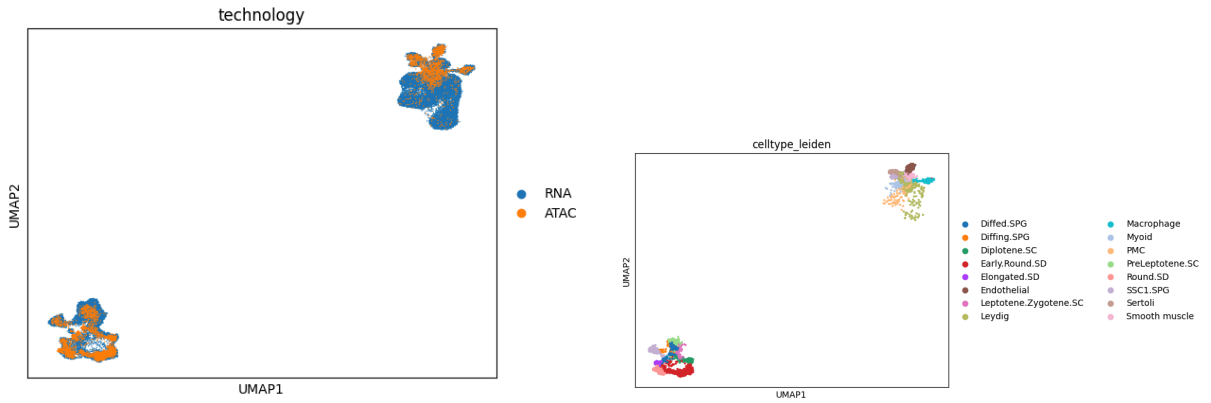
After annotating the scRNA-seq data we can use that to annotate the scATAC-seq data. Firstly, each sample was processed similar to the scRNA-seq data of quality control, filtering, normalised and LSI transformation instead of PCA. A gene activity matrix was created from the filtered cells and features from *refdata-cellranger-arc-GRCh38-2024-A*.

The gene activity matrix is a proxy for gene expression. This works as the ATAC data is the open chromatin regions and active transcription and therefore expression of RNA happens in these open chromatin regions. It is not a perfect comparison as the open chromatin regions also contain many other elements and not only genes.

Next, the combined RNA matrix and gene activity matrix are normalised, scaled and PCA transformed, before being concatenated together. This ensures best possible label transfer between the two sets. `harmony` was used to integrate the two sets, followed by `bbknn` to remove batch effects between the RNA set and ATAC set.

Finally, doing the label transfer of celltype from scRNA-seq to scATAC-seq. The label transfer was done by finding the RNA neighbours of each ATAC cell and assigning the consensus cell. If the distance was too big from an ATAC cell to its neighbours they were not assigned during the first pass. On the second pass the already assigned ATAC cells was included as neighbours. Passes were repeated until all cells had been annotated.

Using the NOA1 sample as an example yields **Figure 12**.



(a) NOA1 batch

(b) NOA1 celltype assignment

Figure 12: UMAP plot of NOA1 integration.

The complete set of all samples can be found in Section C.1.

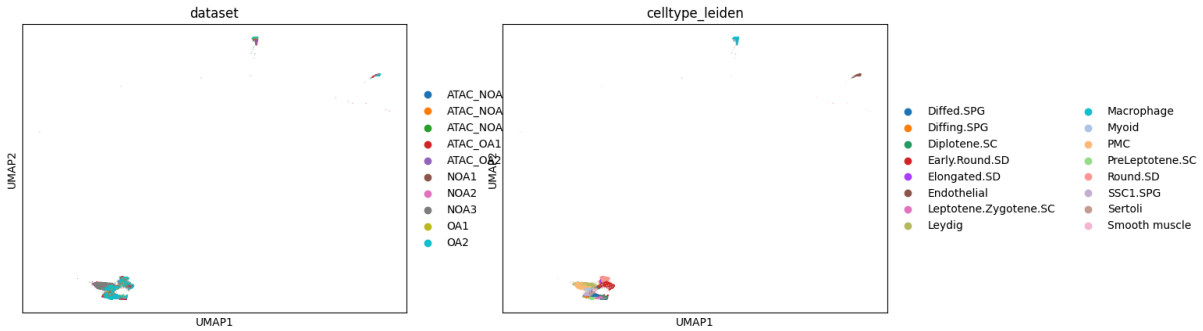


Figure 13: UMAP plot of integrated scATAC-seq on scRNA-seq

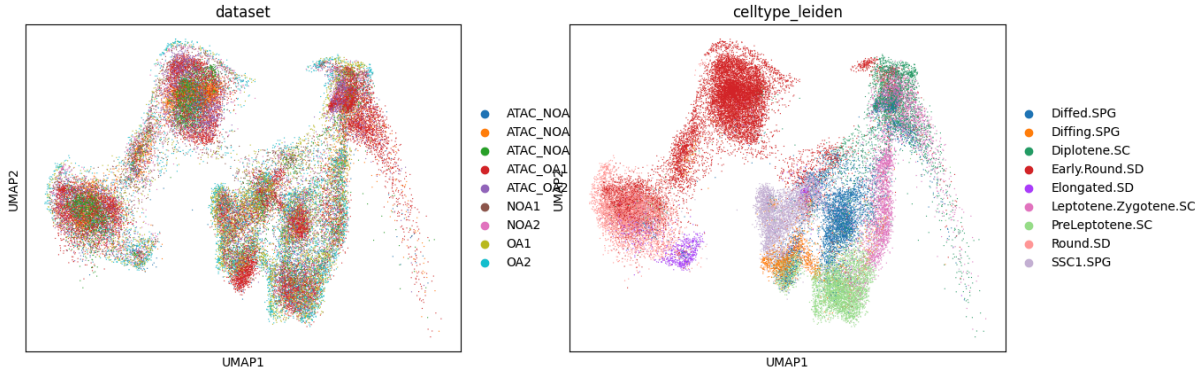


Figure 14: UMAP plot of integrated scATAC-seq on scRNA-seq subset on only germ cells for the integration.

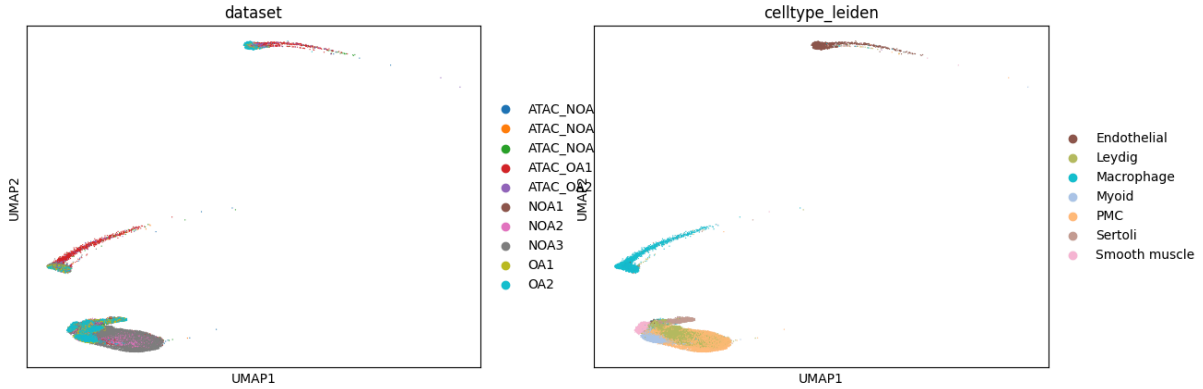


Figure 15: UMAP plot of integrated scATAC-seq on scRNA-seq subset on only somatic cells for the integration.

5 Stage 3b: Celltype annotation of scATAC-seq data with pycistopic

5.1 Preparing for cistopic objects

5.2 Cistopic objects

5.2.1 Model selection

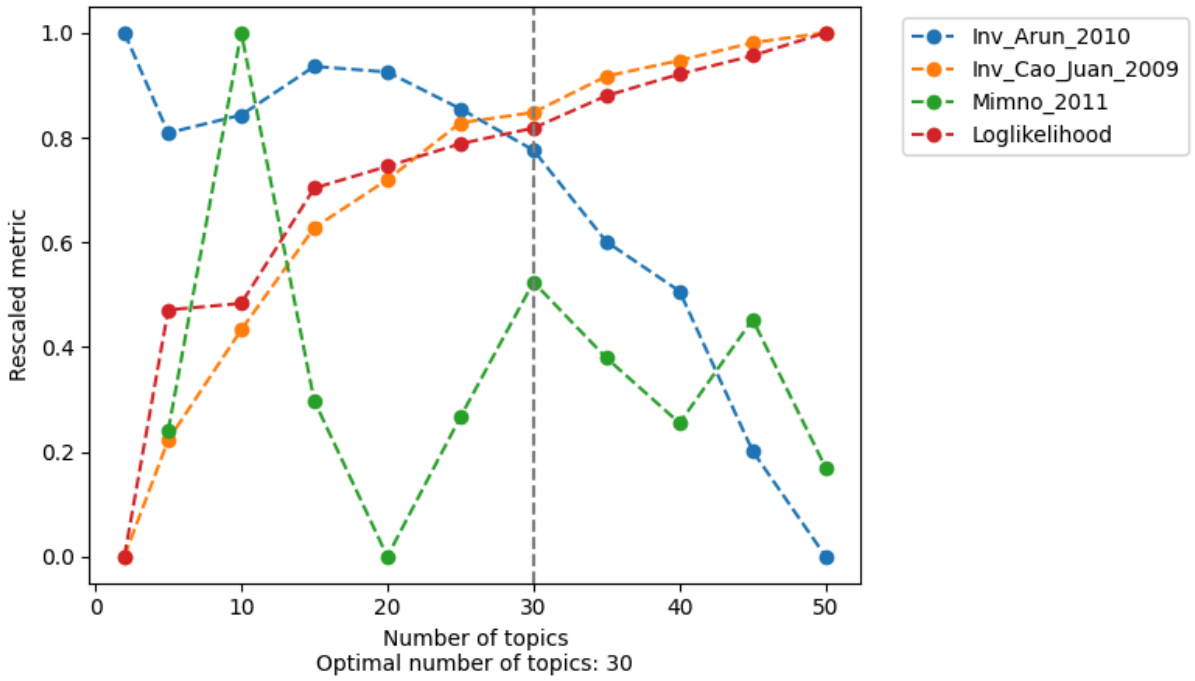


Figure 16: LDA model selection

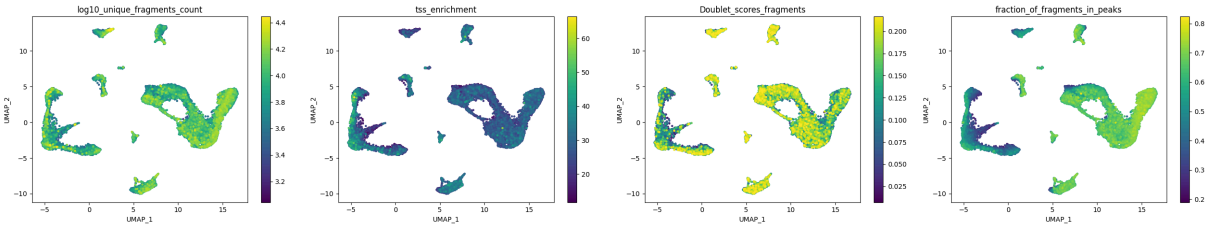


Figure 17: Statistics of scATAC-seq

5.2.2 Clustering

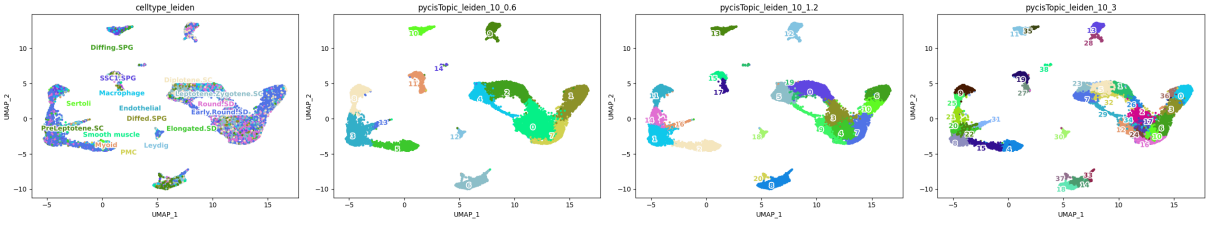


Figure 18: Clustering of scATAC-seq

5.2.3 Topics

Figure 19: Topic enrichment for all 30 topics.

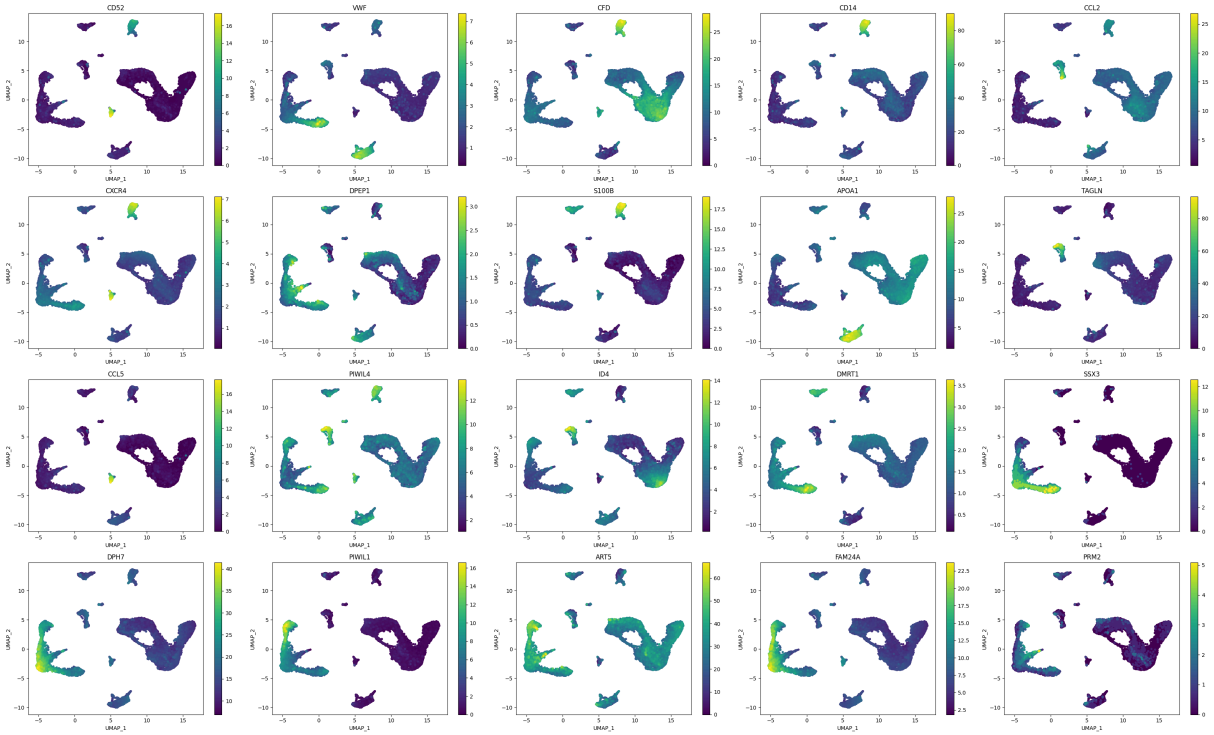


Figure 20: Enrichment of certain marker genes. *CD52*; B, *VWF*; endothelial, *CFD*; leydig, *CD14*; macrophage, *CCL2*; mast, *CXCR4*; plasma, *DPEP1*; PMC, *S100B*; schwann, *APOA1*; sertoli, *TAGLN*; smooth muscle, *CCL5*; T, *PIWIL4*; SSC0, *ID4*; SSC1.SPG, *DMRT1*; Diffing.SPG, *SSX3*; Diffed.SPG, *DPH7* PreLeptotene.SC, *PIWIL1*; Pachytene.SC, *ART5*; Diplotene.SC, *FAM24A*; Early.Round.SD, *PRM2*; Elongated.SD

5.2.4 Label transfer

harmony seems best.

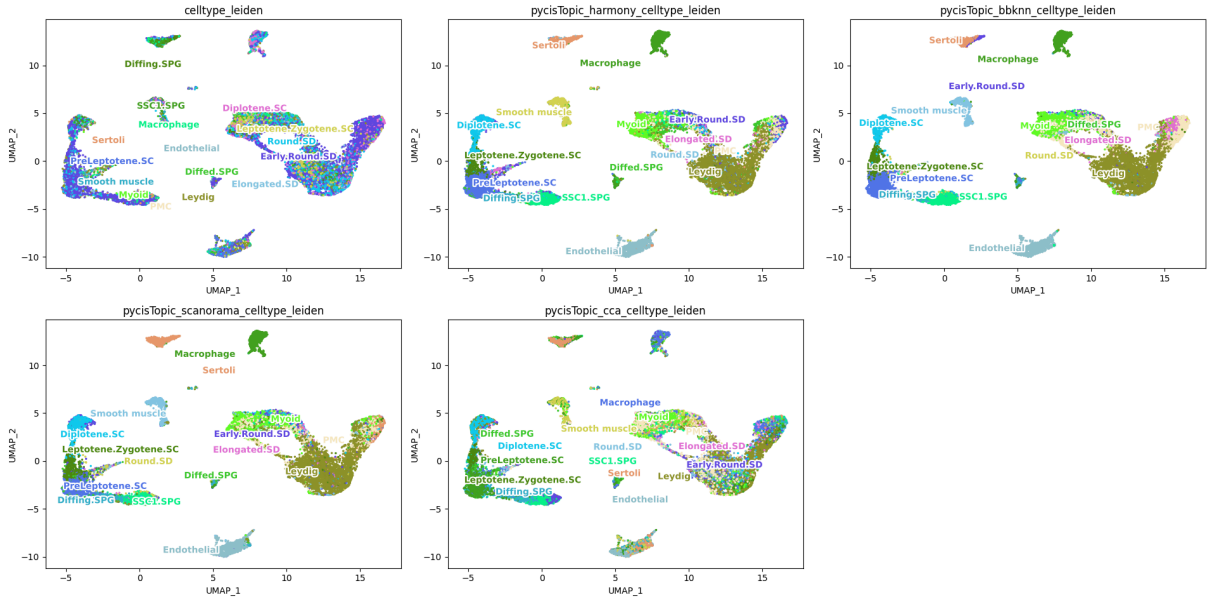


Figure 21: Celltype annotation using different integration methods using scRNA-seq to annotate.

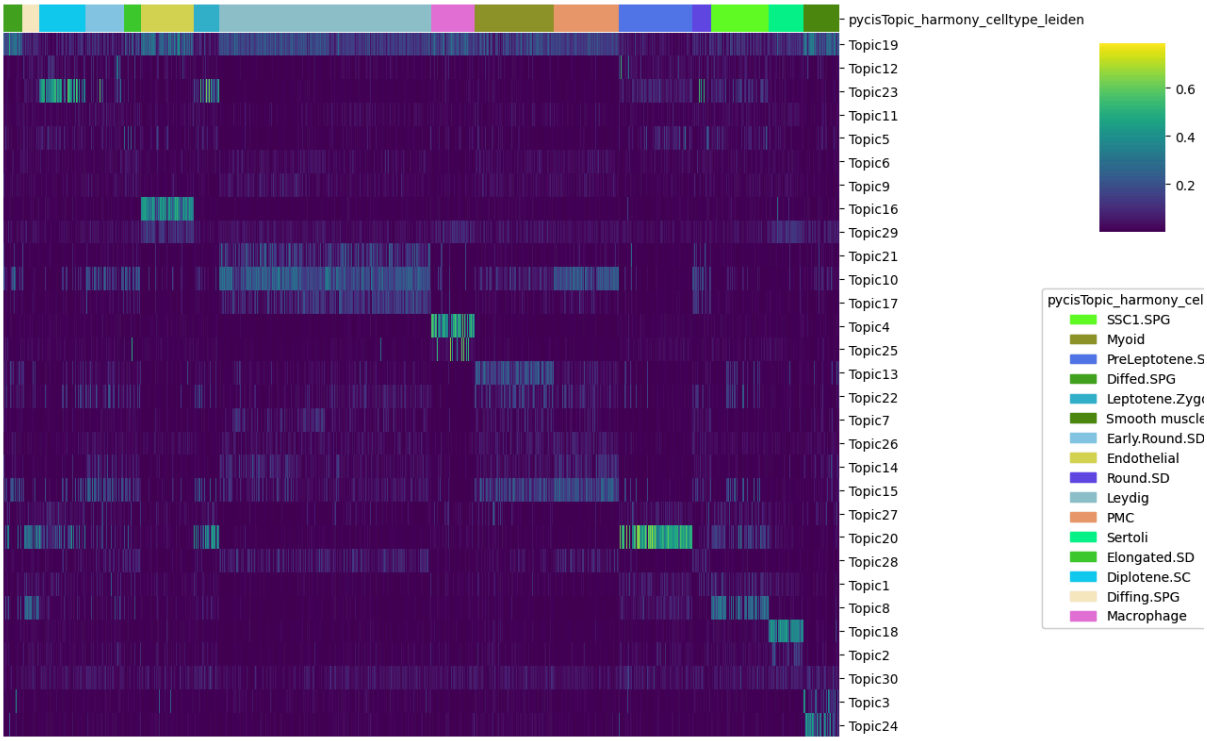


Figure 22: Topic enrichment heatmap for harmony annotated cell types.

6 Conclusion

In conclusion we completed one of the main research questions namely the clean spermatogenic trajectory (spermatogonia → spermatocytes → spermatids) and supporting somatic lineages for both scRNA-seq and scATAC-seq. Both were done with two different methodologies scRNA-seq using gene markers of leiden clustering and SCVI-tools CellAssign model, scATAC-seq using ingesting scRNA-seq data and pycistopic workflow with label transfer from scRNA-seq data annotations.

Learned a lot about working with single cell data using python for analysis. As the project ran out of time, not all of the initial goals were completed.

References

[1] González-Blas CB, Hulselmans G. Features — pycisTopic Documentation. [accessed 2026 Jan 9]. https://pycistopic.readthedocs.io/en/latest/features.html

[2] Soraggi S. SamueleSoraggi/PIB-johan-olesen. 2025 Dec 18 [accessed 2026 Jan 9]. https://github.com/SamueleSoraggi/PIB-johan-olesen

[3] Virshup I, Rybakov S, Theis FJ, Angerer P, Wolf FA. Anndata: Access and Store Annotated Data Matrices. Journal of Open Source Software. 2024 [accessed 2026 Jan 9];9(101):4371. https://joss.theoj.org/papers/10.21105/joss.04371. doi:10.21105/joss.04371

[4] Virshup I, Bredikhin D, Heumos L, Palla G, Sturm G, Gayoso A, Kats I, Koutrouli M, Berger B, Pe'er D, et al. The Scverse Project Provides a Computational Ecosystem for Single-Cell Omics Data Analysis. Nature Biotechnology. 2023 [accessed 2026 Jan 9];41(5):604–606. https://www.nature.com/articles/s41587-023-01733-8. doi:10.1038/s41587-023-01733-8

[5] Wolf FA, Angerer P, Theis FJ. SCANPY: Large-Scale Single-Cell Gene Expression Data Analysis. Genome Biology. 2018 [accessed 2026 Jan 9];19(1):15. https://doi.org/10.1186/s13059-017-1382-0. doi:10.1186/s13059-017-1382-0

[6] Bredikhin D, Kats I, Stegle O. MUON: Multimodal Omics Analysis Framework. Genome Biology. 2022 [accessed 2026 Jan 9];23(1):42. https://doi.org/10.1186/s13059-021-02577-8. doi:10.1186/s13059-021-02577-8

[7] Gayoso A, Lopez R, Xing G, Boyeau P, Valiollah Pour Amiri V, Hong J, Wu K, Jayasuriya M, Mehlman E, Langevin M, et al. A Python Library for Probabilistic Analysis of Single-Cell Omics Data. Nature Biotechnology. 2022 [accessed 2026 Jan 9];40(2):163–166. https://www.nature.com/articles/s41587-021-01206-w. doi:10.1038/s41587-021-01206-w

[8] Ansel J, https://orcid.org/0009-0007-5207-2179, View Profile, Yang E, https://orcid.org/0009-0008-0621-7872, View Profile, He H, https://orcid.org/0009-0004-1133-816X, View Profile, Gimelshein N, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. 2024 [accessed 2026 Jan 9]:929–947. (ACM Conferences). https://dlnext.acm.org/doi/10.1145/3620665.3640366. doi:10.1145/3620665.3640366

[9] Bravo González-Blas C, De Winter S, Hulselmans G, Hecker N, Matetovici I, Christiaens V, Poovathingal S, Wouters J, Aibar S, Aerts S. SCENIC+: Single-Cell Multiomic Inference of Enhancers and Gene Regulatory Networks. Nature Methods. 2023 [accessed 2026 Jan 9];20(9):1355–1367. https://www.nature.com/articles/s41592-023-01938-4. doi:10.1038/s41592-023-01938-4

[10] Bredikhin D. Processing Gene Expression of 10k PBMCs — Muon-Tutorials Documentation. [accessed 2026 Jan 9]. https://muon-tutorials.readthedocs.io/en/latest/single-cell-rna-atac/pbmc10k/1-Gene-Expression-Processing.html

[11] Wang S, Wang H, Jin B, Yan H, Zheng Q, Zhao D. scRNA-seq and scATAC-seq Reveal That Sertoli Cell Mediates Spermatogenesis Disorders through Stage-Specific Communications in Non-Obstructive Azoospermia Park J, Choi M, editors. eLife. 2025 [accessed 2025 Sept 12];13:RP97958. https://doi.org/10.7554/eLife.97958. doi:10.7554/eLife.97958

[12] Satpathy AT, Granja JM, Yost KE, Qi Y, Meschi F, McDermott GP, Olsen BN, Mumbach MR, Pierce SE, Corces MR, et al. Massively Parallel Single-Cell Chromatin Landscapes of Human Immune Cell Development and Intratumoral T Cell Exhaustion. Nature Biotechnology. 2019 [accessed 2026 Jan 9];37(8):925–936. https://www.nature.com/articles/s41587-019-0206-z. doi:10.1038/s41587-019-0206-z

[13] Specifying Input FASTQ Files for Cell Ranger ATAC | Official 10x Genomics Support. [accessed 2026 Jan 9]. https://www.10xgenomics.com/support/software/cell-ranger-atac/latest/analysis/inputs/specifying-input-fastq-files

[14] Soraggi S, Molinaro E. Hds-Sandbox/NGS_summer_course_Aarhus: 2024 Summer Course. 2024 June 24 [accessed 2026 Jan 12]. https://zenodo.org/records/12514541. doi:10.5281/zenodo.12514541

Appendix A Cellranger

Naming scheme for Cellranger ATAC count [13]: `[Sample Name]S1_L00[Lane Number][Read Type]_001.fastq.gz`, where `Read type`:

- `I1`: Dual index i7 read (optional)
- `R1`: Read 1
- `I2`: Dual index i5 read
- `R3`: Read 2

`Lane Number` does not matter. `Sample Name` can be anything.

Example of NOA1 sample:

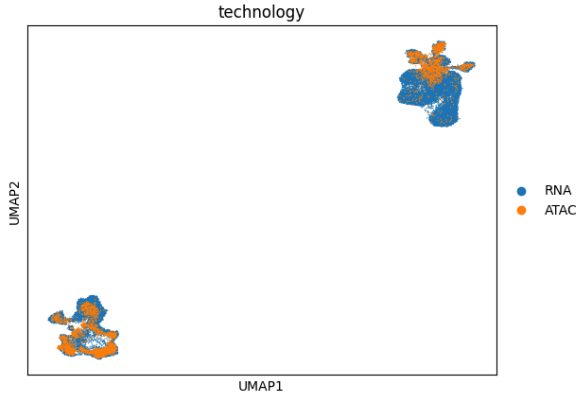
- SRR31097965_S1_L001_I1_001.fastq
- SRR31097965_S1_L001_I2_001.fastq
- SRR31097965_S1_L001_R1_001.fastq
- SRR31097965_S1_L001_R2_001.fastq

Appendix B scRNA-seq

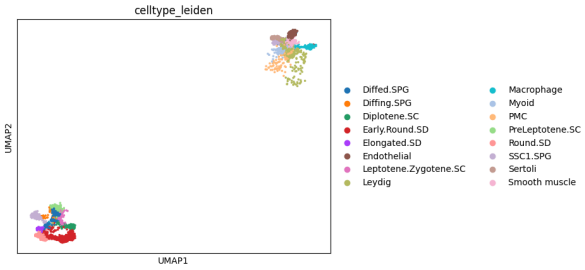
Cell Type	Marker Genes
Sertoli	SOX9, GATA1, ACSL4, WT1, GAS6, VIM, CD99, APOA1
Leydig	CFD, IGF1, IGFBP5, INSL3
Myoid	MYH11, ACTA2
Macrophage	CD68, CD163, MSR1, CD14
Endothelial	VWF, SOX17
B	CD52
Mast	TPSB2
T	CCL5
Plasma	IGLC2
PMC	DPEP1
Schwann	S100B
Smooth muscle	TAGLN
SSC1.SPG	ID4, UTF1, TWIST1, KRT18, AES, ENHO, C19orf84, SIX1, PIWIL4
SSC2.SPG	ASB9, L1TD1, NANOS3, FAM25G, CITED2
Diffing.SPG	MKI67, ACTL8, KLHL15, DMRT1, PABPC4
Diffed.SPG	SSX3, TEX19, PNMA6E, PEG10, TKTL1, BEND2
PreLeptotene.SC	DPH7, PRDM9, PRAP1, KIF1A, MAGEA9B, FAM9C, ATP6AP1, OTUD6A, CCNB3
Leptotene.Zygotene.SC	LINC00668, AP000350.6, LINC01120, AL138889.1, LINC00865, C18orf63
Pachytene.SC	PCDHB6, POM121L12, POM121L2, C9orf57, AL133279.3, AC093326.2, AL353581.1, PCDHB5, PCDHB2, AC135012.2, MNS1, CCDC42
Diplotene.SC	ART5, KLB, B3GALT4, ELMO3, RTN4RL2, AC073263.2, LINC00588, WNT6, LINC01309, AL121936.1, KRT72, LDHC
Early.Round.SD	FAM24A, SPACA1, LINC01351, ABRA, LINC00703, LINC02502, PPP1R27, H1F0, TPRG1, LRRC39, C1orf87, TMIGD3
Round.SD	TNP2, LINC02314, LINC01921, PRSS37, FBXO39, LINC02400, DHRS3, FAM205C, CXorf65, SCP2D1, LINC01548, CCDC179, AC010255.3, SPEM3
Elongated.SD	TSSK6, CABS1, SPATA3, CCDC196, TSPAN16, PHOSPHO1, SPEM2, TEX44, LRRD1, SPEM1, GLUL

Appendix C scATAC-seq

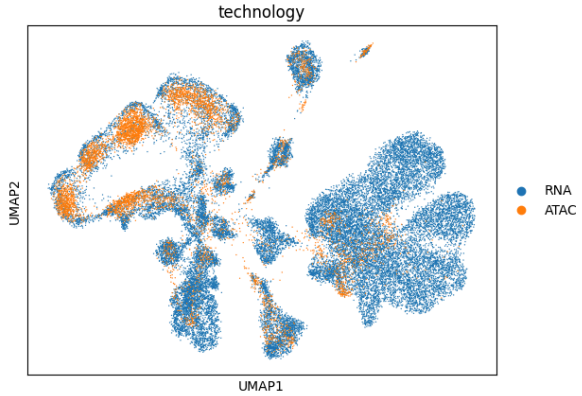
C.1 UMAP plots of integration of each scATAC-seq sample



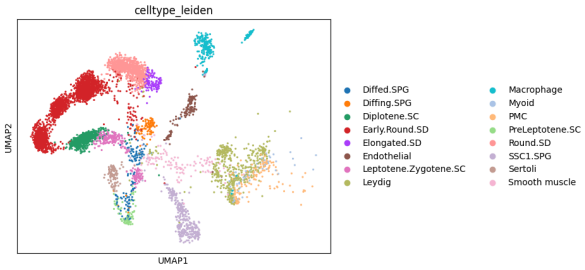
(a) NOA1 batch



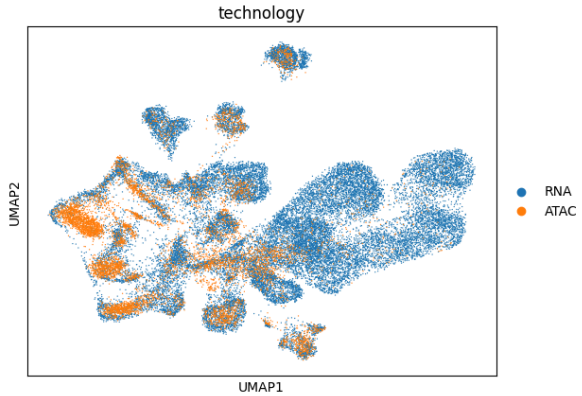
(b) NOA1 celltyp assignment



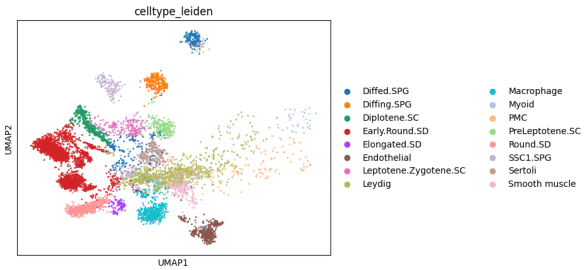
(c) NOA2 batch



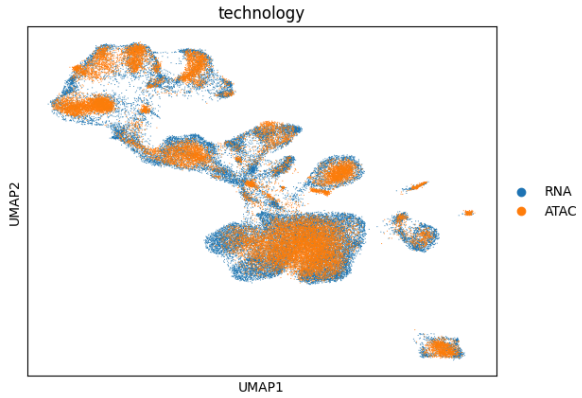
(d) NOA2 celltype assignment



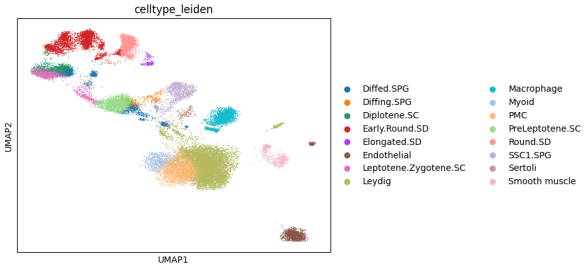
(e) NOA3 batch



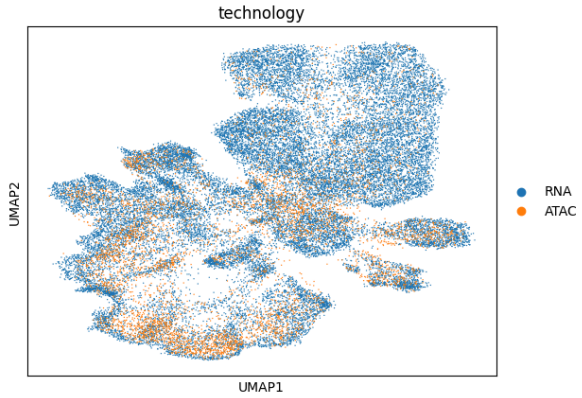
(f) NOA3 celltype assignment



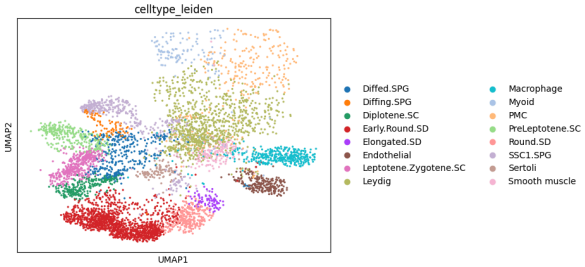
(g) OA1 batch



(h) OA1 celltype assignment



(i) OA2 batch



(j) OA2 celltype assignment

Figure 23: UMAP plots of integration of each scATAC-seq sample

Appendix D pycistopic workflow