SCAN10: A REPRODUCIBLE AND STANDARDIZED PIPELINE FOR PROCESSING 10X SINGLE CELL RNASEQ DATA

Maxime Lepetit[1], Mirela Diana Ilie[2,3], Marie Chanal[2], Gerald Raverot[2,4], Philippe Bertolino[2], Franck Picard[1] and Olivier Gandrillon[1,5,*].

1 - ENS de Lyon, CNRS UMR 5239, Laboratory of Biology and Modelling of the Cell, Lyon, France.
2 - Centre de Recherche en Cancérologie (CRCL) - INSERM 1052 - CNRS 5286 - Centre Léon Bérard - Université Claude Bernard Lyon 1, Institut Convergence Plascan, Lyon, France.
3 - Endocrinology Department, "C.I.Parhon" National Institute of Endocrinology, Bucharest, Romania.
4 - Endocrinology Department, Reference Center for Rare Pituitary Diseases HYPO, Groupement Hospitalier Est" Hospices Civils de Lyon, Bron, France.
5 - Inria Center Grenoble Rhone-Alpes, Equipe Dracula, Villeurbanne, France.

Keywords : pipeline, single cell transcriptomics, data analysis, mapping, annotation

1 Abstract

The recent explosion of single cell transcriptomics has led to the challenge of developing data analysis pipelines that are both fully reproducible and modular while allowing interoperability across multiple systems and institutions.

We present scAN10, a processing pipeline of 10X single cell RNAseq data, that inherits the ability to be executed on most computational infrastructures, thanks to Nextflow DSL2. The modular nature of Nextflow pipelines allows to easily integrate and assess different blocks for a given analysis step.

We illustrate the benefit of using scAN10 by showing its ability to assess the impact of the mapping step on the resulting output using a clinical 10X scRNAseq analysis of a human pituitary gonadotroph tumour.

2 Introduction

The recent explosion of single cell transcriptomics, mostly through scRNAseq, has led to the challenge of developing data analysis pipelines that are both fully reproducible and modular while allowing interoperability across multiple systems and institutions.

The initial step in scRNAseq data analysis consist in generating a count matrix from `fastq` sequence files. This step is often overlooked (see e.g. [1]), although it can represent a very critical step (see e.g. [2]). Therefore one needs freely available analysis pipeline that could allow to verify the impact of some early analysis step easily, such as the nature of the `GTF` file used [3] or downstream steps such as the normalization method on the generation of the count matrix.

The most widely used existing solutions like the Seurat suite [4] have been designed to be as much user-friendly as possible and therefore does not offer easy solution for incorporating alternative low-level analysis steps.

This is why in the present paper we approached this challenge by designing `scAN10`, a processing pipeline of 10X single cell RNAseq data, that inherits the ability to be executed on most computational infrastructures, thanks to Nextflow DSL2. The modular nature of Nextflow pipelines allows to easily integrate and assess different bricks blocks for a given analysis step. `scAN10` is available as an open source Gitlab repository. It takes raw paired-end

`fastq` files and genomic files (FASTA and GTF annotation files) as input, and outputs a clustered dimensionally-reduced version of the dataset.

We illustrate the benefit of using `scAN10` by showing its ability to assess the impact of the mapping step on the resulting UMAP projection as well as on some specific gene identification using a clinical 10X scRNA-seq analysis of one human pituitary gonadotroph tumour.

3 Results

3.1 Pipeline description

Figure 1 describes the overall processing of the sequencing files with the ordering of all steps described in section 5.

3.2 Raw dataset

To showcase the applicability of `scAN10` we processed a clinical dataset from a human pituitary gonadotroph tumour acquired from one male patient and sequenced by 10Xgenomics. The dataset was processed using `scAN10` with the following parameters :

- max_feature_RNA = 7000
- max_percent_mito= 25

The clustering (resolution =0.7) and UMAP embedding were done based on

the ten first principal components of the PCA, as determined by the rule of thumb heuristic and the broken stick method [5].

3.3 Version annotation effect

We first assess the impact of the Ensembl version of the `GTF` file on the final output. `GTF` files and their corresponding `FASTA` files were downloaded via `ftp` protocol using the `--`version parameter (see section 5). We set Cellranger as the default mapper and then assessed the impact of 4 different annotation releases (93, 98, 103 and 106) on the number of detected genes (figure 2A) and on the count per genes as assessed with the CHGA gene. (Figure 2B).

The overall impact of the `GTF` version seems relatively modest, especially in regard to the number of counts. The 106 version allowed to identify a larger number of genes and was kept for the next step.

3.4 Comparing filtered with unfiltered annotations

We then assessed the impact of filtrating the `GTF` file with the `mkgtf` Cellranger function. This filtration step is intended to remove unwanted genes classified by biotype. We used the default values of that function that removes biotypes such as gene_biotype:pseudogene from the `GTF` annotation file.

This step removes some ambiguity between reads location by allowing reads that would be flagged as multi mapped reads to be included in the quantifi-

cation.

We observed that this filtration step indeed had a major impact on both the number of genes detected (Figure 3A) but also on gene counts (Figure 3B).

3.5 Cellranger versus Kallisto-bustools

We then compared the impact of two popular alignment tools for single-cell RNA sequencing (Kallisto-bustools and Cellranger) using the 10x Genomics pre-built Cellranger reference packages version 2020-A for human.

As seen in Figure 4A, 81 % of the genes were identified by both algorithms whereas Kallisto-bustools identified more genes than Cellranger.

The impact of the mapper on counts for specific genes seemed to be negligible (Figure 4B). Therefore this tends to favor Kallisto-bustools for downstream analyses.

We finally assessed the impact of the mapper choice on the final clustering step. As seen in Figure 5, the impact was modest but apparent (e.g. cluster number 1 in the CR dataset was split in two in the KB dataset).

4 Discussion

We described `scAN10`, a Nextflow based processing pipeline of 10X single cell RNAseq data.

By applying `scAN10` to a clinical dataset we showed that the impact of the annotation version was relatively modest although using the latest Ensembl release (106) of the `GTF` and `FASTA` allows to identify a larger number of genes.

As expected, filtrating the `GTF` files by removing unwanted genes based on 10X reference packages generation had a major impact both on the number of genes but also on gene counts.

However, recent study observed differences in the mitochondrial content of the resulting cells when comparing a filtered annotation to the full annotation. Therefore removing processed pseudogene might lead to an enrichment of the mitochondrial content [6].

Futhermore when using Kallisto-bustools instead of Cellranger the impact of the count numbers for specific genes seemed to be small but meaningful. Kallisto-bustools produced higher total number of genes detected which 5169 unique genes as compared to Cellranger.

The final combination that was found to be the most effective for our dataset therefore was using Kallisto-bustools together with the filtered version of the 106 `GTF`. There is no reason to believe that such parameters might be universally applicable, and we therefore recommend the use of `scAN10` so assess such an impact on any other dataset before proceeding with higher level analysis.

With Nextflow each step is encapsulated in independent blocks called pro-

cesses. Each process communicates via channels. The orchestration of the workflow with DSL2 syntax allows to easily modify the pipeline, by adding new processes modules. In the future we expect to include some normalization procedures to the pipeline:

- The basic global-scaling normalization method from Seurat that divides the feature expression for each cell by the total expression and multiplies this by a scale factor and log-transforms the result.[7]
- `Sctransform` which uses regularized negative binomial regression and computes Pearson residuals that correspond to the normalized expression levels for each transcript.[8]
- `Scran` which uses pooling-based size factor estimation.[9]

The use of `scAN10` should be made straightforward to assess a combination of low-level steps together with the normalization step on the resulting output.

One major limitation using Nextflow is a lack of interactivity during pipeline running. Indeed, when Nextflow pipelines run, although it can output files, there is no interactive dialogue that could allow the user to modify parameters during the run. All the pipelines parameter need to be defined and set in the launching command.

We finally believe that `scAN10` will be a useful tool for the growing community of 10X scRNAseq *aficionados*.

5 Material and methods

5.1 Study ethic approval

This work is part of the SPACE-PIT study (MR004 n21-5439). It was approved by the Hospices Civils de Lyon ethical committee and registered at the "Centre National Information et Liberté" (`CNIL.fr`) under the reference 20_098. Informed consent was obtained from the patient.

5.2 Single cells preparation and sequencing

A tumor fragment from a gonadotroph surgically-resected adenoma was collected in Dulbecco's Modified Eagle Medium (DMEM, cat 41965062; Life Technologies). Single-cell suspension of the resected fragment was obtained through mechanical enzymatic dissociation (Collagenase P, cat 11213865001) then passed through a 100 μm mesh-strainer (#732-2759, VWR international).

Red blood cells were eliminated using a 10-minute incubation with a commercial red blood cell lysis buffer (eBioscience, cat #00-4300-54). The whole process was achieved within the 2 hours following the surgical resection, cell viability was evaluated to reach at least 70 percent prior encapsulation. Generation of the library was done using a Chromium controller from 10xGenomics. The entire procedure was achieved as recommended by the manufacturer's for the v3 reagent kit (10xGenomics). Single cell suspension

was loaded onto a Chromium Single Cell A Chip, aiming for 5,000 cells. The cDNA was amplified after a reverse transcription step prior a SPRIselect (Beckman Coulter), a cleaning, a quantification and an enzymatic fragmentation prior to the library sequencing on a NextSeq500 system (Illumina).

5.3 Implementation

The `scAN10` pipeline was powered and supported by the reactive workflow manager Nextflow [10]. In addition, the pipeline was coded with the DSL2 syntax extension. Nextflow simplifies the writing of computational pipelines by making them portable, scalable,parallelizable and ensuring a high level of reproducibility. Nextflow provides native support for container technologies such as Docker or Singularity. Each process in the pipeline will be run in a container. A reproducible container environment is built for each process from Docker images stored on the DockerHub. The analyses can be run on the user's preferred computing platform. Using the configuration file and corresponding profile, the pipeline can be run on a local computer via Docker or Singularity, as well as on a high-performance computing (HPC) cluster or in cloud-based environments. Nextflow includes a cache-based pipeline resume feature, no matter what the reason was for its stopping.

5.4 Input

`scAN10` takes three mandatory parameters as input: the paired-end `FASTQ` R1 and R2 from 10X Chromium sequencing and two genomics files (one `FASTA` an one `GTF`).

`FASTQ` files store the nucleotide sequence and the associated sequencing quality scores. Those files must be provided through the `--fastq` pipeline parameter and needs to be compressed in `gzip` format. Reads are sequenced in paired-ends thus 2 reads will be produced for each sequencing lane. In the case a sample has been sequenced on several lanes, all reads R1 can be concatenate together and all reads R2 together. The mapping step require the input of two additional files corresponding to the species of interest:

- A `FASTA` genomic file which stores the raw genome sequence.
- A Gene transfer format (`GTF`) file which stores genome annotation including gene positions.

One should note that for human datasets, `FASTA` and `GTF` files can be downloaded automatically by specifying a version number as an entry parameter (`--version`) available on the ENSEMBL database.

5.5 Preprocessing & Mapping

First, `FASTQ` files are processed and trimmed by using `Fastp` v0.20.1 , an ultra-fast `FASTQ` preprocessor with useful quality control and data-filtering features [11]. Reads with phred quality ≥30 (-q 30) is qualified to the quantification step. Length filtering is disabled (-L) while adapter sequence auto-detection is enabled (`--`detect_adapter_for_pe).

To reduce overlapping annotation, we recommend and provide an optional parameter `--`filtergtf allow the `GTF` filtration with Cellranger's `mkgtf` function based on the same biotype attribute used to generate the `GTF` file for the human Cell Ranger reference package. (see : `https://support.10xgenomics.com/single-cell-gene-expression/software/release-notes/build`)

The processed files are then mapped of a reference genome to quantify gene expression. In `scAN10`, the user can specify two different mappers (see below): Kallisto-bustools v0.26.0 [12] or Cellranger v5.0.1 [13].

- Kallisto-bustools is used thought the Python wrapper : kb_python. Starting with `FASTA` and `GTF` files an index of the reference can be built as a colored De Bruij graph with Kallisto via kb ref. with default parameter. Once an index has been generated or downloaded, `kb count` uses Kallisto to pseudoalign reads and `bustools` to quantify the data.
- Cellranger created and prepared reference package with Cellranger's mkref function. The alignment was run via Cellranger's count with

default parameters as described on `10xgenomics.com`.

5.6 Quality control

Empty Droplets With Droplet-based data most of the barcodes in the matrix correspond to empty droplets (eg barcode with sum expression over gene of 0). While Cellranger takes care of the empty droplet filtering, empty droplets from Kallisto-bustools gene expression matrix need to be removed. The Kallisto-bustools outputs were imported into `R` with customed `R` function. The UMI total counts were ranked using `DropletUtils::barcodeRanks` function from `DropletUtils` v1.14.2 . Empty droplets were removed by selecting the inflection point value on the resulting knee plot (lower cutoff = 10). The Cellranger matrix is imported from the standard filtered barcode output. After importation, either gene expression matrices were converted as Seurat object (Seurat v4.0.4) with `Seurat::CreateSeuratObject` function, including features detected in at least 3 cells (min.cells =3).

Low quality cells After removing empty droplets the pipeline computes 3 QC metrics per sample: the number of unique features per cells, the number of UMIs by cells and the percent of mitochondrial counts by cells. These QC metrics are then used to discard three main types of low quality cells [14] :

1. Cells in apoptosis may exhibit high % mitochondrial and low number of UMIs per cells

2. Cells that failed during library preparation exhibit low number of unique gene counts and low number of UMIs per cells

3. The pipeline uses the `R` package `DoubletFinder` v2.0.3 to detect and remove the potential doublets from the dataset. See [15] and [16].

Thresolding Default thresholding parameters used by the `scAN10` pipeline are:

1. min_feature_RNA=500
2. min_ncount_RNA=0
3. max_percent_mito="adaptive"
4. max_feature_RNA="adaptive"
5. max_ncount_RNA="adaptive"

To define maximum values of threshold `scAN10` allows to use an adaptive filter defined by a certain number of median absolute deviations (MADs) away from the median [17]:

$$median + 3 * mad$$

Non-expressed genes Genes with sum count along cells equal to 0 (eg not-expressed genes) are removed.

5.7 Normalization

To normalize the filtered gene expression matrix, the user can use `Sanity`, a Bayesian algorithm to infer gene-expression state [18]. We are fully aware that the normalization of single cell transcriptomic data is a research field on its own (see e.g. [2] and citations therein), and we expect this block in the pipeline to be susceptible to be modified in future versions of the pipeline. The modularity of the Nextflow syntax makes it ideal for such additions.

5.8 Clustering and two-dimensional visualization

A final step consists in variable features selection with `Seurat::FindVariableFeatures` using the `vst` method (selection.method = "vst") and selecting the 2000 first highly variable genes (nfeatures = 2000), followed by a first linear dimensionality reduction using PCA (Seurat::RunPCA with default parameter). The M first axis of the PCA are then used for the nearest-neighbor graph construction with `Seurat::FindNeighbors` function (dims=1:M).

Cluster determination was performed using the Louvain algorithm [19] run with `Seurat::FindClusters` function (algorithm = 1). The resolution parameter set by default to 0.7 is use for increasing (values >1) or decreasing (values <1) the number of clusters obtained. The quality of the clustering was assessed using the Silhouette score [20]. Finally, non-linear dimensional representation (using either t-SNE [21] or UMAP [22]) is then performed using `Seurat::RunTSNE` or `Seurat::RunUMAP` with default parameters using

the same number of dimensions than the nearest-neighbor graph building.

The resolution parameters used for clustering and the number of principal components kept for both clustering and dimension reduction embeddings can be respectively modified by the user at the start of the pipeline by the `--resolution_clustering` and the `--principal_component` parameters.

Alternatively this last step can be skipped allowing the user to use their own clustering method. Similarly, to avoid the introduction of layers of complexity and simplify the pipeline usage, the automatic annotation of clusters was not introduced. Users can annotate their dataset manually.

5.9 Output

Exhaustive list of processes outputs is available on the Readme of the gitlab repository (section 6).

6 Availability

scAN10 is freely available at : `https://gitbio.ens-lyon.fr/LBMC/sbdm/scan10`

7 Acknowledgements

We thank the Institut Convergence Plascan (Grant Number ANR-17-CONV-0002) for their support.

Mirela Diana Ilie has been supported by the Fondation ARC pour la recherche sur le cancer (ARCMD-DOC12020020001361).

We thank (i) Emmanuel Jouanneau and Alexandre Vasiljevic from the Neurosurgery and Pathology Departments, Reference Center for Rare Pituitary Diseases HYPO, "Groupement Hospitalier Est "Hospices Civils de Lyon, Bron, France for providing diagnosed Gonadotroph tumor material; (ii) Hector Hernandez-Vargas for his advices and support with setting the single cell dissociation and subsequent bioinformatic analysis; and (iii) Laurent Modolo from the LBMC for his help using the Nextflow syntax.

We thank the Pôle Scientifique de Modélisation Numérique (PSMN, Ecole Normale Supérieure de Lyon) where computations were performed.

We finally thank the BioSyL Federation (http://www.biosyl.org), the LabEx Ecofect (ANR-11-LABX-0048) and the LabEx Milyon of the University of Lyon for inspiring scientific events.

8 Author contributions

Maxime Lepetit wrote the scAN10 script, performed the analysis, generated the figures and wrote the paper. Mirela Diana Ilie participated in generating and analyzing the 10X data. Marie Chanal participated in generating the 10X data. Gerald Raverot participated in generating and analyzing the 10X data. Philippe Bertolino helped designing the study, analyzing the results,

reviewed and edited the manuscript, and secured the funding. Franck Picard helped designing the study and analyzing the results, reviewed and edited the manuscript, and secured the funding. Olivier Gandrillon helped designing the study and analyzing the results, wrote the paper, and secured the funding.

9 References

[1] M. D. Luecken and F. J. Theis. "Current best practices in single-cell RNA-seq analysis: a tutorial". In: Mol Syst Biol 15.6 (2019), e8746.

[2] R. Zhang, G. S. Atwal, and W. K. Lim. "Noise regularization removes correlation artifacts in single-cell RNA-seq data preprocessing". In: Patterns (N Y) 2.3 (2021), p. 100211.

[3] A.-H. Pool, H. Poldsam, S. Chen, M. Thomson, and Y. Oka. "Enhanced recovery of single-cell RNA-sequencing reads for missing gene expression data". In: bioRxiv (2022), p. 2022.04.26.489449.

[4] Y. Hao, S. Hao, E. Andersen-Nissen, 3rd Mauck W. M., S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zager, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. M. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. "Integrated analysis of multimodal single-cell data". In: Cell 184.13 (2021), 3573–3587 e29.

[5] D. A. Jackson. "Stopping Rules in Principal Components Analysis: A Comparison of Heuristical and Statistical Approaches". In: Ecology 74 (1993), pp. 2204–2214.

[6] Ralf Schulze Brüning, Lukas Tombor, Marcel H Schulz, Stefanie Dimmeler, and David John. "Comparative analysis of common alignment tools for single-cell RNA sequencing". In: GigaScience 11 (Jan. 2022). giac001. eprint: https://academic.oup.com/gigascience/article-pdf/doi/10.1093/gigascience/giac001/42297447/giac001.pdf.

[7] R. Satija, J. Farrell, and D. Gennert. "Spatial reconstruction of single-cell gene expression data." In: Nat Biotechnol 33.5 (2015), pp. 495–502.

[8] C. Hafemeister and R. Satija. "Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression." In: Genome Biol 20.1 (2019), p. 296.

[9] A.T. L. Lun, K. Bach, and J.C. Marioni. "Pooling across cells to normalize single-cell RNA sequencing data with many zero counts." In: Genome Biol 17.1 (2019), p. 75.

[10] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C Notredame. "Nextflow enables reproducible computational workflows". In: Nature biotechnology 35.4 (2017), pp. 316–319.

[11] S. Chen, Y. Zhou, Y. Chen, and J. Gu. "fastp: an ultra-fast all-in-one FASTQ preprocessor". In: Bioinformatics 34.17 (2018), pp. i884–i890.

[12] P. Melsted, A. S. Booeshaghi, F. Gao, E. Beltrame, L. Lu, K. E. Hjorleifsson, J. Gehring, and L. Pachter. "Modular and efficient preprocessing of single-cell RNA-seq". In: Nat. Biotechnol. 39 (2021), pp. 813–818.

[13] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, M. T. Gregory, J. Shuga, L. Montesclaros, J. G. Underwood, D. A. Masquelier, S. Y. Nishimura, M. Schnall-Levin, P. W. Wyatt, C. M. Hindson, R. Bharadwaj, A. Wong, K. D. Ness, L. W. Beppu, H. J. Deeg, C. McFarland, K. R. Loeb, W. J. Valente, N. G. Ericson, E. A. Stevens, J. P. Radich, T. S. Mikkelsen, B. J. Hindson, and J. H. Bielas. "Massively parallel digital transcriptional profiling of single cells". In: Nat Commun 8 (2017), p. 14049.

[14] T. Ilicic, J.K. Kim, A.A. Kolodziejczyk, F.O. Bagger, D.J. McCarthy, J.C. Marioni, and Teichmann S.A. "Classification of low quality cells from single-cell RNA-seq data." In: Genome Biology 17.1 (2016), p. 29.

[15] C.S. McGinnis, L.M. Murrow, and Z.J. Gartne. "DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors." In: Cell systems 8.4 (2019), 329–337. e4.

[16] N.M. Xi and J.J. Li. "Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data". In: Cell Systems 12.2 (2021), 176–194.e6.

[17] D. J. McCarthy, K. R. Campbell, A. T. L. Lun, and Q. F. Wills. "Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R". In: Bioinformatics 33.8 (Jan. 2017), pp. 1179–1186. eprint: https://academic.oup.com/bioinformatics/article-pdf/33/8/1179/25150420/btw777.pdf.

[18] J. Breda, M. Zavolan, and E. van Nimwegen. "Bayesian inference of gene expression states from single-cell RNA-seq data". In: Nat Biotechnol (2021).

[19] V. D. Blondel, J.-L. Guillaume, R Lambiotte, and E. Lefebvre. "Fast unfolding of communities in large networks". In: J. Stat. Mech. P10008 (2008), p. 12.

[20] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: Journal of Computational and Applied Mathematics 20 (1987), pp. 53–65.

[21] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE". In: Journal of Machine Learning Research 9 (2008), pp. 2579–2605.

[22] L. McInnes, J. Healy, N. Saul, and L. Großberger. "UMAP: uniform manifold approximation and projection". In: J. Open Source Softw 3 (2018), p. 861.

10 Figures caption

Figure 1 : A metromap view of the scAN10 pipeline.

Figure 2 : A. Number of detected genes when using different versions of the `GTF` file. B. Violin plot representation of the impact of the `GTF` version on the UMI counts for the CHGA gene.

Figure 3 : A. Number of detected genes when using either an unfiltered (106) of filtered (filter106) version of the `GTF` file. B. Violin plot representation of the filtration impact on the UMI counts for the CD68 gene.

Figure 4 : A. Number of detected genes when using either Cellranger (CR) or Kallisto-bustools (KB) as an alignment tool. B. Violin plot representation on the UMI counts for two genes, CHGA and RBP4.

Figure 5 : UMAP representation (A and B) and Silhouette scores (C and D) of the clusters obtained on data processed with CellRanger (A and C) or Kallisto-bustools (B and D). In E is shown an alluvial plot highlighting the conservation and differences in cluster composition depending upon the initial mapping method.

11 Figures

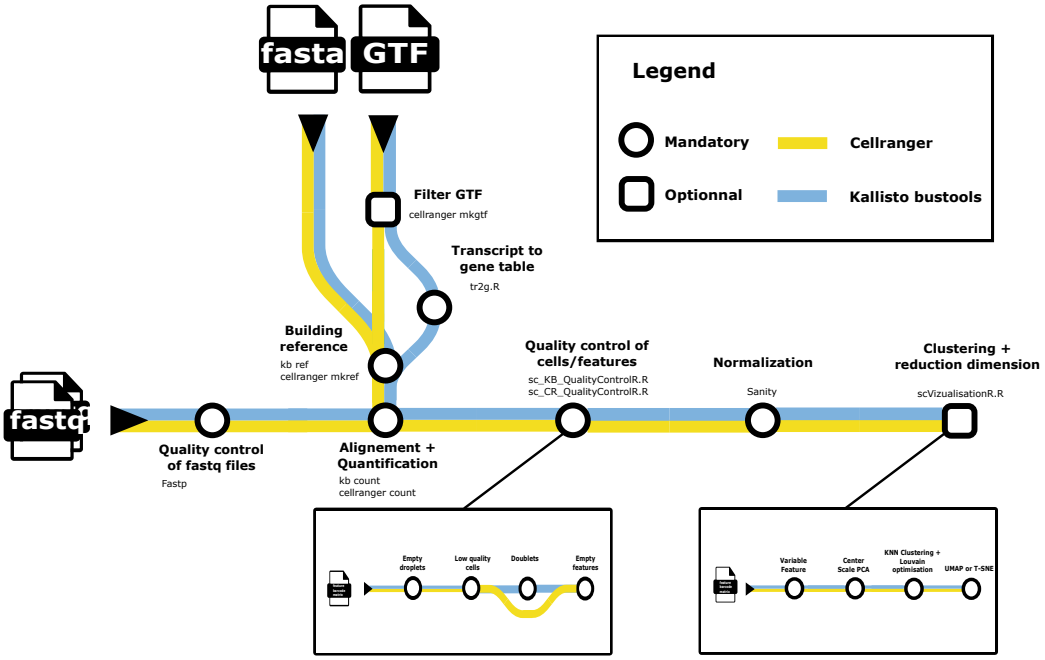

Figure 1

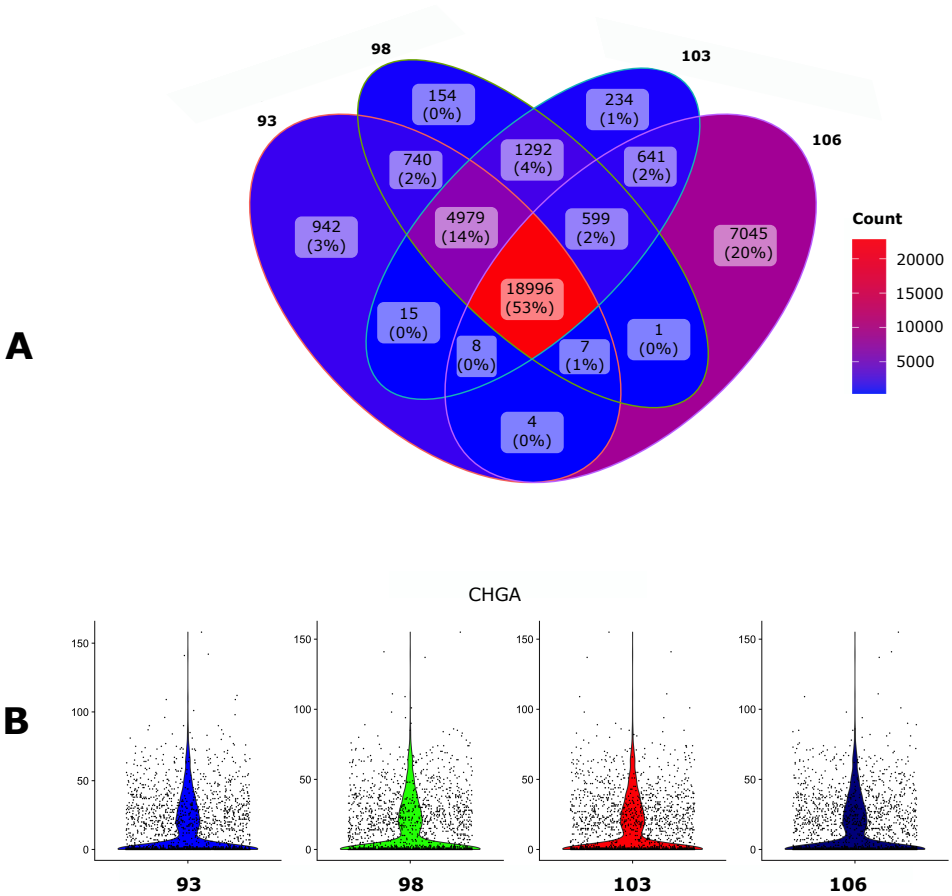
A
93
98
103
106
154 (0%)
234 (1%)
740 (2%)
1292 (4%)
641 (2%)
942 (3%)
4979 (14%)
599 (2%)
7045 (20%)
18996 (53%)
15 (0%)
8 (0%)
7 (1%)
1 (0%)
4 (0%)
Count
20000
15000
10000
5000
B
CHGA
150
100
50
0
93
98
103
106

Figure 2

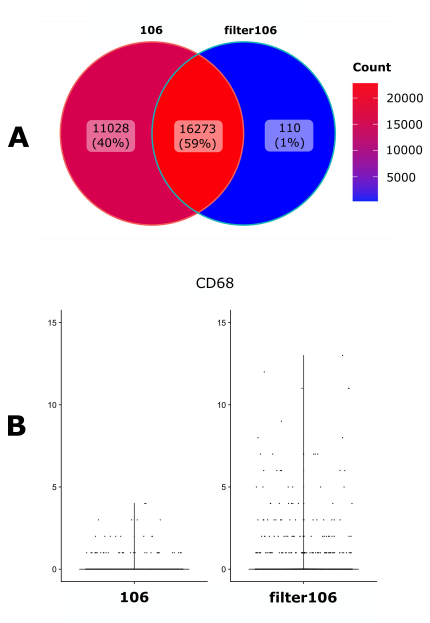
A
106
filter106
11028
(40%)
16273
(59%)
110
(1%)
Count
20000
15000
10000
5000
B
CD68
15
10
5
0
106
filter106

Figure 3

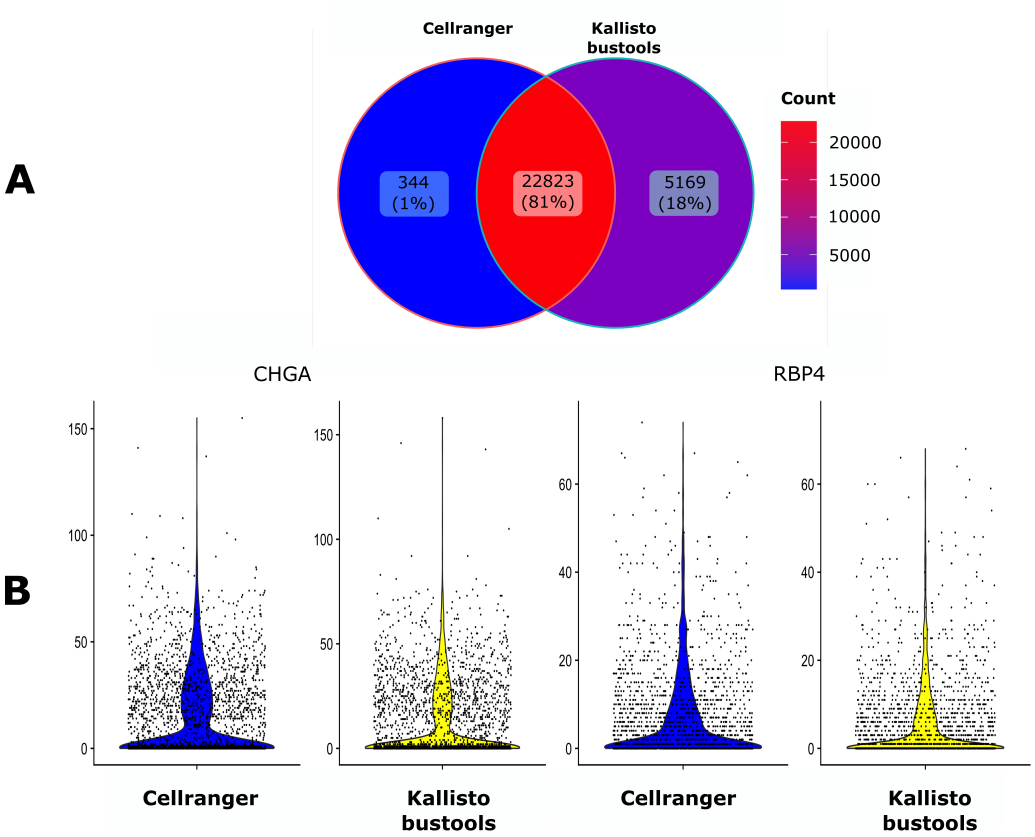
A
Cellranger
Kallisto bustools
344 (1%)
22823 (81%)
5169 (18%)
Count
20000
15000
10000
5000
B
CHGA
RBP4
Cellranger
Kallisto bustools
Cellranger
Kallisto bustools

Figure 4

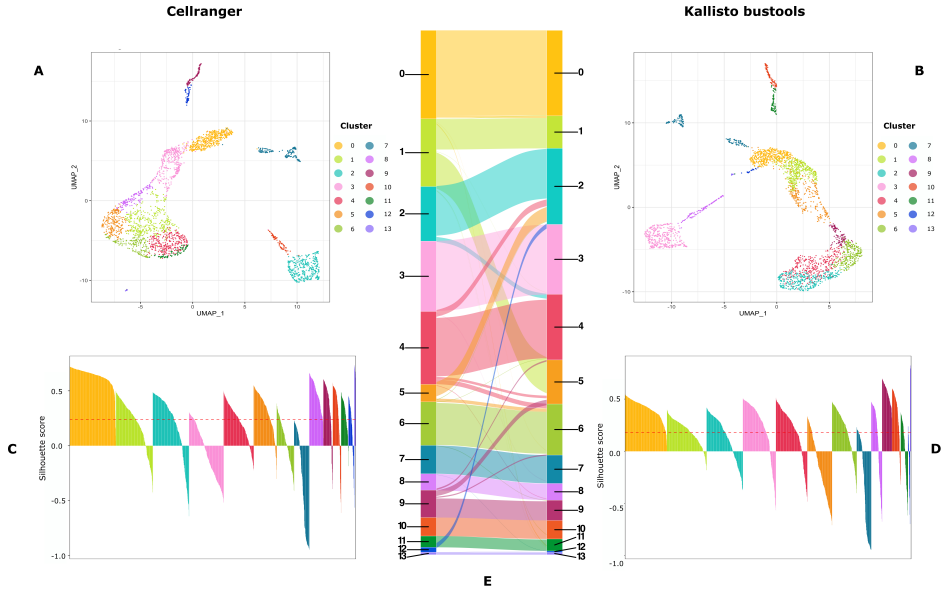
Cellranger
Kallisto bustools
A
B
C
D
E
Cluster
UMAP_1
UMAP_2
Silhouette score

Figure 5