Methods

Evgenii O. Tretiakov

2023-03-14

# Presentation  
library("glue")  
library("knitr")  
  
# JSON  
library("jsonlite")  
  
# Tidyverse  
library("tidyverse")

dir.create(here::here("output", DOCNAME), showWarnings = FALSE)  
  
write\_bib(c("base", "Seurat", "SeuratWrappers", "SeuratDisk", "sctransform",  
 "glmGamPoi", "patchwork", "scCustomize", "Nebulosa", "clustree",  
 "mrtree", "gprofiler2", "cowplot", "UpSetR", "ggstatsplot",  
 "gridExtra", "tidyverse", "dplyr", "tidyr", "magrittr", "stringr",  
 "skimr", "future", "purrr", "here", "workflowr", "zeallot", "knitr",  
 "kableExtra", "rmarkdown", "reticulate"),  
 file = here::here("output", DOCNAME, "packages.bib"))

versions <- list(  
 biomaRt = packageVersion("biomaRt"),  
 cellbender = "docker://etretiakov/cellbender:v0.0.1",  
 cellranger = "7.1.0",  
 cellranger\_ref = "mm10\_optimized\_v.1.0",  
 clustree = packageVersion("clustree"),  
 cowplot = packageVersion("cowplot"),  
 dplyr = packageVersion("dplyr"),  
 future = packageVersion("future"),  
 ggplot2 = packageVersion("ggplot2"),  
 ggstatsplot = packageVersion("ggstatsplot"),  
 glmGamPoi = packageVersion("glmGamPoi"),  
 gprofiler2 = packageVersion("gprofiler2"),  
 gridExtra = packageVersion("gridExtra"),  
 here = packageVersion("here"),  
 kableExtra = packageVersion("kableExtra"),  
 knitr = packageVersion("knitr"),  
 magrittr = packageVersion("magrittr"),  
 mrtree = packageVersion("mrtree"),  
 Nebulosa = packageVersion("Nebulosa"),  
 pandoc = rmarkdown::pandoc\_version(),  
 patchwork = packageVersion("patchwork"),  
 purrr = packageVersion("purrr"),  
 python = "3.8.8",  
 R = str\_extract(R.version.string, "[0-9\\.]+"),  
 reticulate = packageVersion("reticulate"),  
 rmarkdown = packageVersion("rmarkdown"),  
 scCustomize = packageVersion("scCustomize"),  
 sctransform = packageVersion("sctransform"),  
 Seurat = packageVersion("Seurat"),  
 skimr = packageVersion("skimr"),  
 stringr = packageVersion("stringr"),  
 Snakemake = "7.21.0",  
 souporcell = "shub://wheaton5/souporcell",  
 tidyr = packageVersion("tidyr"),  
 tidyverse = packageVersion("tidyverse"),  
 UpSetR = packageVersion("UpSetR"),  
 viridis = packageVersion("viridis"),  
 workflowr = packageVersion("workflowr"),  
 zeallot = packageVersion("zeallot")  
)

# Pre-processing

The Cell Ranger pipeline (v7.1.0) (Zheng et al. 2017) was used to perform sample demultiplexing, barcode processing and single-nuclei gene counting. Briefly, samples were demultiplexed to produce a pair of FASTQ files for each sample. Reads containing sequence information were aligned using the optimised mouse genome reference (vmm10\_optimized\_v.1.0) provided by Pool’s lab based on the default Cell Ranger mm10 genome version 2020-A that was cleared from gene overlaps, poorly annotated exons and 3’-UTRs and intergenic fragments (Pool et al. 2022). PCR duplicates were removed by selecting unique combinations of cell barcodes, unique molecular identifiers (UMIs) and gene ids with the final results being a gene expression matrix that was used for further analysis.

# Quality control

## Droplet selection

The droplet selection method of Cell Ranger identified 923 nuclei in ventrobasal thalamus and in principal sensory trigeminal nucleus - 731 nuclei based on EmptyDrops method (Lun et al. 2019) incorporated into cellranger count pipeline.

## Ambient RNA removal

Using those values as the expected number of cells, we applied a neural network-based approach called CellBender (docker://etretiakov/cellbender:v0.0.1) (Fleming et al. 2022). We set a false positive rate threshold at the level of 0.01 and set the neural network to learn over 150 epochs with a total of 5000 droplets included based on knee plots (please see online supplementary Cell Ranger reports).

## Doublets detection

We quantified log probability to be a doublet for every cell based on apriori knowledge of genotypes from each input sample and called variant occurrence frequencies that allowed to cluster nuclei to source organism or classify as a doublet (Heaton et al. 2020) (see shub://wheaton5/souporcell).

## Further filtering

Gene annotation information was added using the gprofiler2 package (v0.2.1) (Reimand et al. 2016); thus we filter cells based on high content of mitochondrial, ribosomal or hemoglobin proteins genes, specifically thresholds set as 1%, 1% and 0.5%; additionally pseudogenes and poorly annotated genes were also deleted from count matrix. Moreover, cells of low complexity were filtered out as (). Therefore, cells were assigned cell cycle scores using the CellCycleScoring function in the Seurat package (v4.3.0).

# Clustering

## Gene selection

We used the selection method in the Seurat package (v4.3.0) (Satija et al. 2015; Stuart and Satija 2019), which uses a modern variance stabilising transformation statistical technic that utilises scaling to person residuals (Hafemeister and Satija 2019). That way, we selected 3000 highly variable genes per dataset and regressed out complexity and cell-cycle variability prior to the final scaling of filtered matrixes.

## Graph-based and multi-level reconcile tree clustering

We performed Leiden algorithm graph-based clustering. PCA was performed using the selected genes and the jackknife tested (Chung and Storey 2015) principal components (we tested the significance of feature for randomly picked 1% of data over 1000 iterations; see PCScore function in functions.R script of code directory) were used to construct a shared nearest neighbour graph using the overlap between the 15 nearest neighbours of each cell. Leiden modularity optimisation (Traag, Waltman, and Eck 2019) was used to partition this graph with an array of resolution parameters where 30 modularity events were sampled between 0.2 and 2.5. Clustering tree visualisations (Zappia and Oshlack 2018) were produced using the clustree package (v0.5.0) showing the resolution of previously identified clusters. By inspecting these resolutions reconcile tree produced by mrtree package (v0.0.0.9000) (Peng et al. 2021) and calculating adjusted multi-resolution Rand index chosen as maximum value if there is no higher modularity within 0.05 AMRI difference (see SelectResolution in function.R file of code directory).

# Marker genes

Marker genes for each cluster were identified using logreg test (Ntranos et al. 2019) implemented in Seurat framework (v) (Stuart and Satija 2019). Genes were considered significant markers for a cluster if they had an FDR less than 0.001. Identities were assigned to each cluster by comparing the detected genes to previously published markers and our own validation experiments.

# Other packages

Visualisations and figures were primarily created using the ggplot2 (v3.4.1), cowplot (v1.1.1) (Wilke 2020) and patchwork (v1.1.2.9000) packages using the viridis colour palette (v0.6.2) for continuous data. UpSet plots (Conway, Lex, and Gehlenborg 2017) were produced using the UpSetR package (v1.4.0) (Gehlenborg 2019) with help from the gridExtra package (v2.3) (Auguie 2017). Data manipulation was performed using other packages in the tidyverse (v1.3.2.9000) (Wickham 2023) particularly dplyr (v1.1.0) (Wickham et al. 2023), tidyr (v1.1.0) (Wickham, Vaughan, and Girlich 2023) and purrr (v1.0.1) (Wickham and Henry 2023). The analysis project was managed using the Snakemake system (v ) (Mölder et al. 2021) and the workflowr (v1.7.0) (Blischak, Carbonetto, and Stephens 2021) package which was also used to produce the publicly available website displaying the analysis code, results and output. Reproducible reports were produced using knitr (v1.42) (Xie 2023) and R Markdown (v2.20) (Allaire et al. 2023) and converted to HTML using Pandoc (v2.19.2).

# Summary

## Output files

versions <- purrr::map(versions, as.character)  
versions <- jsonlite::toJSON(versions, pretty = TRUE)  
readr::write\_lines(versions,  
 here::here("output", DOCNAME, "package-versions.json"))

# References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2023. *Rmarkdown: Dynamic Documents for r*.

Auguie, Baptiste. 2017. *gridExtra: Miscellaneous Functions for "Grid" Graphics*.

Blischak, John, Peter Carbonetto, and Matthew Stephens. 2021. *Workflowr: A Framework for Reproducible and Collaborative Data Science*. <https://github.com/workflowr/workflowr>.

Chung, Neo Christopher, and John D. Storey. 2015. “Statistical significance of variables driving systematic variation in high-dimensional data.” *Bioinformatics (Oxford, England)* 31 (4): 545–54. <https://doi.org/10.1093/bioinformatics/btu674>.

Conway, Jake R, Alexander Lex, and Nils Gehlenborg. 2017. “UpSetR: An R Package for the Visualization of Intersecting Sets and Their Properties.” Edited by John Hancock. *Bioinformatics* 33 (18): 2938–40. <https://doi.org/10.1093/bioinformatics/btx364>.

Fleming, Stephen J., Mark D. Chaffin, Alessandro Arduini, Amer-Denis Akkad, Eric Banks, John C. Marioni, Anthony A. Philippakis, Patrick T. Ellinor, and Mehrtash Babadi. 2022. “Unsupervised Removal of Systematic Background Noise from Droplet-Based Single-Cell Experiments Using CellBender,” July. <https://doi.org/10.1101/791699>.

Gehlenborg, Nils. 2019. *UpSetR: A More Scalable Alternative to Venn and Euler Diagrams for Visualizing Intersecting Sets*. <http://github.com/hms-dbmi/UpSetR>.

Hafemeister, Christoph, and Rahul Satija. 2019. “Normalization and Variance Stabilization of Single-Cell RNA-Seq Data Using Regularized Negative Binomial Regression.” *Genome Biology* 20 (1): 296. <https://doi.org/10.1186/s13059-019-1874-1>.

Heaton, Haynes, Arthur M. Talman, Andrew Knights, Maria Imaz, Daniel J. Gaffney, Richard Durbin, Martin Hemberg, and Mara K. N. Lawniczak. 2020. “Souporcell: robust clustering of single-cell RNA-seq data by genotype without reference genotypes.” *Nature methods* 17 (6): 615–20. <https://doi.org/10.1038/s41592-020-0820-1>.

Lun, Aaron T. L., Samantha Riesenfeld, Tallulah Andrews, The Phuong Dao, Tomas Gomes, and John C. Marioni. 2019. “EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data.” *Genome biology* 20 (1): 63. <https://doi.org/10.1186/s13059-019-1662-y>.

Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, et al. 2021. “Sustainable Data Analysis with Snakemake.” *F1000Research* 10 (January): 33. <https://doi.org/gjjkwv>.

Ntranos, Vasilis, Lynn Yi, Páll Melsted, and Lior Pachter. 2019. “A Discriminative Learning Approach to Differential Expression Analysis for Single-Cell RNA-Seq.” *Nature Methods* 16 (2): 163–66. <https://doi.org/gftsnv>.

Peng, Minshi, Brie Wamsley, Andrew G. Elkins, Daniel H. Geschwind, Yuting Wei, and Kathryn Roeder. 2021. “Cell type hierarchy reconstruction via reconciliation of multi-resolution cluster tree.” *Nucleic acids research* 49 (16): e91. <https://doi.org/10.1093/nar/gkab481>.

Pool, Allan-Hermann, Helen Poldsam, Sisi Chen, Matt Thomson, and Yuki Oka. 2022. “Enhanced Recovery of Single-Cell RNA-Sequencing Reads for Missing Gene Expression Data,” April. <https://doi.org/10.1101/2022.04.26.489449>.

Reimand, Jüri, Tambet Arak, Priit Adler, Liis Kolberg, Sulev Reisberg, Hedi Peterson, and Jaak Vilo. 2016. “g:Profiler-a web server for functional interpretation of gene lists (2016 update).” *Nucleic acids research* 44 (W1): W83–89. <https://doi.org/10.1093/nar/gkw199>.

Satija, Rahul, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. 2015. “Spatial reconstruction of single-cell gene expression data.” *Nature biotechnology* 33 (5): 495–502. <https://doi.org/10.1038/nbt.3192>.

Stuart, Tim, and Rahul Satija. 2019. “Integrative single-cell analysis.” *Nature reviews. Genetics* 20 (5): 257–72. <https://doi.org/10.1038/s41576-019-0093-7>.

Traag, V. A., L. Waltman, and N. J. van Eck. 2019. “From Louvain to Leiden: Guaranteeing Well-Connected Communities.” *Scientific Reports* 9 (1): 5233. <https://doi.org/10.1038/s41598-019-41695-z>.

Wickham, Hadley. 2023. *Tidyverse: Easily Install and Load the Tidyverse*.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*.

Wickham, Hadley, and Lionel Henry. 2023. *Purrr: Functional Programming Tools*.

Wickham, Hadley, Davis Vaughan, and Maximilian Girlich. 2023. *Tidyr: Tidy Messy Data*.

Wilke, Claus O. 2020. *Cowplot: Streamlined Plot Theme and Plot Annotations for Ggplot2*. <https://wilkelab.org/cowplot/>.

Xie, Yihui. 2023. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. <https://yihui.org/knitr/>.

Zappia, Luke, and Alicia Oshlack. 2018. “Clustering Trees: A Visualization for Evaluating Clusterings at Multiple Resolutions.” *GigaScience* 7 (7): giy083. <https://doi.org/10.1093/gigascience/giy083>.

Zheng, Grace X. Y., Jessica M. Terry, Phillip Belgrader, Paul Ryvkin, Zachary W. Bent, Ryan Wilson, Solongo B. Ziraldo, et al. 2017. “Massively parallel digital transcriptional profiling of single cells.” *Nature communications* 8 (January): 14049. <https://doi.org/10.1038/ncomms14049>.