OXFORD (GIGA)n SCIENCE

GigaScience, 2023, **12**, 1–10
DOI: 10.1093/gigascience/giad091
Research

Cellsnake: a user-friendly tool for single-cell RNA sequencing analysis

Sinan U. Umu [1,*], Karoline Rapp Vander-Elst [2], Victoria T. Karlsen [2], Manto Chouliara [2], Espen Sønderaal Bækkevold [2,3], Frode Lars Jahnsen [1,2] and Diana Domanska [2,4]

[1]Department of Pathology, Institute of Clinical Medicine, University of Oslo, Oslo 0372, Norway
[2]Department of Pathology, Oslo University Hospital-Rikshospitalet, Oslo 0372, Norway
[3]Institute of Oral Biology, University of Oslo, Oslo 0372, Norway
[4]Department of Microbiology, University of Oslo, Rikshospitalet, Oslo 0372, Norway
*Correspondence address. Sinan U. Umu, Department of Pathology, University of Oslo, Sognsvannsveien 20, Rikshospitalet, 0372, Oslo E-mail: sinanuu@uio.no

Abstract

Background: Single-cell RNA sequencing (scRNA-seq) provides high-resolution transcriptome data to understand the heterogeneity of cell populations at the single-cell level. The analysis of scRNA-seq data requires the utilization of numerous computational tools. However, nonexpert users usually experience installation issues, a lack of critical functionality or batch analysis modes, and the steep learning curves of existing pipelines.

Results: We have developed cellsnake, a comprehensive, reproducible, and accessible single-cell data analysis workflow, to overcome these problems. Cellsnake offers advanced features for standard users and facilitates downstream analyses in both R and Python environments. It is also designed for easy integration into existing workflows, allowing for rapid analyses of multiple samples.

Conclusion: As an open-source tool, cellsnake is accessible through Bioconda, PyPi, Docker, and GitHub, making it a cost-effective and user-friendly option for researchers. By using cellsnake, researchers can streamline the analysis of scRNA-seq data and gain insights into the complex biology of single cells.

Keywords: scRNA, RNA-seq, workflow, microbiome, single-cell, snakemake, Seurat

Background

Single-cell RNA sequencing (scRNA-seq) is a method used to study gene expression at the single-cell level. This stands in contrast to bulk RNA sequencing, which provides information only on the average transcript expression within a population of cells. With recent technological advancements and decreasing sequencing costs, scRNA-seq has become increasingly accessible, enabling researchers to identify novel cell types, cell states, and cellular interactions [1–4].

A standard scRNA-seq bioinformatics workflow typically involves several steps, including data filtering, normalization, scaling, dimensionality reduction, clustering, visualization, differential expression analysis, functional analysis, and annotation [4, 5]. Various analysis workflows for different platforms (i.e., 10X Genomics, Drop-seq, inDrops, SMART-seq2, and Fluidigm C1) have been developed to process, analyze, and holistically visualize scRNA-seq data [2, 6–8]. Popular workflows like Seurat [9], SingleCellExperiment (of Bioconductor) [7], and Scanpy [6] have extensive features for scRNA analysis. The analysis of scRNA-seq data poses several challenges, including the high-dimensional data structure, technical issues (e.g., dead cells, doublets, and low unique molecular identifier [UMI] counts), batch effects, low expression levels, and the presence of complex cell subsets with multiple cell states [5]. To address these, a variety of supplementary bioinformatics tools have been developed. While some of these can be integrated into existing workflows, many require substantial expertise and bioinformatics knowledge.

Another challenge is working with multiple scRNA-seq datasets. Comprehensive documentation for the analysis of a single sample using recommended parameters is usually provided. However, it is hard for a regular user to keep track of all the decisions taken during analyses, especially if more than one sample is available. This also creates challenges if one wants to see the effect of basic parameter changes and document the results for further hypothesis testing. It is also challenging to harness the power of high-performance computing (HPC) systems when needed. There are some efforts to make batch analysis, such as the cloud-based system SingleCAnalyzer [10], the R package scTyper [11], the web application Cellenics (open-source software of Biomage), and Single-Cell Omics workbench on Galaxy [12]. Cellranger from 10X Genomics also provides dataset clustering and basic differential expression analysis [13] for initial quality control (QC). However, all these workflows have limited functionality or were designed for a specific need. Online (or cluster-based) solutions might also not be suitable due to data privacy rules for sensitive data or do not provide compatible files (e.g., R data files) for downstream analysis on another platform.

Here, we introduce cellsnake, a platform-independent command-line application and pipeline for scRNA-seq analysis. Cellsnake provides a reproducible, flexible, and accessible solution for most scRNA-seq data analysis applications. One of the key features of cellsnake is its ability to utilize different scRNA-seq algorithms to simplify tasks such as automatic mitochondrial (MT) gene trimming, selection of optimal clustering

Received: June 1, 2023. **Revised:** August 25, 2023. **Accepted:** October 5, 2023

Table 1: An overview of the tools and algorithms used in the cellsnake workflow, as well as an explanation of what they do and which versions are used

Tool	Version	Reference	Notes
Seurat	4.2.0	[8]	Main analysis platform
SeuratDisk	0.9020	[27]	Format converter
Clustree	0.5.0	[15]	Clustering interrogation
MultiK	1.0	[19]	Optimal cluster detection
miQC	1.6.0	[18]	Auto MT gene trimming
DoubletFinder	2.0.3	[20]	Doublet detection
SingleR	2.0.0	[28]	Cell type annotation
CellTypist	1.2.0	[16]	Cell type annotation
Kraken2	2.1.2	[29]	Metagenomics
CellChat	1.6.1	[30]	Ligand–receptor analysis and miscellaneous
clusterProfiler	4.4.4	[31]	KEGG, GO, and module enrichment
Monocle3	1.0.0	[32]	Cell trajectory and velocity

resolution, doublet filtering, visualization of marker genes, enrichment analysis, and pathway analysis. Cellsnake also allows parallelization and readily utilizes HPC platforms. In addition to that, cellsnake provides metagenome analysis if unmapped reads are available. Another advantage of cellsnake is its ability to generate intermediate files (such as R data files) that can be stored, extracted, shared, or used later for more advanced analyses or for reproducibility purposes. With cellsnake, researchers can perform scRNA-seq data analysis in a reproducible and efficient manner, without requiring extensive bioinformatics expertise.

Methods

Cellsnake workflow and tools

The cellsnake (RRID:SCR_023666 and biotoolsID: cellsnake) wrapper was written in Python, while the main workflow was implemented in Snakemake [14]. To find optimal cluster resolution, we utilized clustree [15]. Seurat analysis pipeline [8] provides all the main functions required for processing scRNA data in cellsnake. These functions are wrapped into different R scripts, which can also be used as standalone scripts by advanced users. Cellsnake facilitates automatic format conversion when required. For instance, CellTypist [16] requires AnnData format, and the workflow converts the files back to the required file format in R. By default, cellsnake stores files into 2 folders: analyses and results. The analyses folder contains metadata and R data files, which can be accessed by the user. Seurat is used for integration, and after integration, the workflow runs on the integrated dataset automatically, and the output files are stored in separate folders (i.e., analyses_integrated and results_integrated).

Parameter selection and autodetection

Cellsnake provides Seurat's default values for fundamental parameters like min.cells (i.e., features detected at least this many cells) or min.features (i.e., cells at least this many features). In addition, nondefault parameters can be provided using a YAML file, and a YAML file template can be printed and edited. Cellsnake determines which principal component exhibits cumulative percent greater than 90% and percent variation associated with the principal component as less than 5 [17]. To filter MT genes, cellsnake uses the miQC tool [18]. If that fails, it uses the median absolute deviation of the MT gene expression as an alternative. MultiK algorithm [19] is used to determine optimal resolution detection, and doublet filtering is done using the DoubletFinder tool [20]. Autodetection of parameters is not offered as a default option in cellsnake due to its computational expense and potential for failure with large sample sizes. Cellsnake utilizes a special directory structure for MT percentage and resolution, and the results are saved in different folders named after the selected parameters. These results are not overwritten and can be reviewed later, or the parameters can be modified for further investigation.

Cellsnake testing and benchmarks

To test cellsnake, we obtained 4 samples containing exclusively macrophages from gut mucosal tissue [21], along with 2 fetal brain datasets [22] and 6 fetal liver datasets [23]. The fetal brain datasets were provided in matrix file format, while the other datasets were in FASTQ format and processed by Cellranger (v.7.0.0) with the default settings and the default databases. For a comprehensive evaluation, we compared the features of cellsnake with 2 other holistic tools, Cellenics and Single Cell Omics workbench [12]. The Cellenics community instance [24] is hosted by Biomage [25].

Results

Cellsnake can be run either as a Snakemake workflow or as a standalone tool

Cellsnake utilizes a variety of tools and algorithms (Table 1) and consists of 2 primary components: the main workflow and the wrapper. The cellsnake wrapper assists with the main workflow and provides an easy-to-use option for users. The workflow (Fig. 1) is primarily designed using the Seurat pipeline (v4.2) and the Snakemake workflow manager. As needed, the workflow integrates various algorithms to enhance the basic functionality of Seurat. For instance, when 1 droplet encapsulates more than 1 cell, it appears as a single cell and can affect the downstream analysis. Addressing this issue in the workflow is crucial [26]. A distinctive feature of cellsnake is its default doublet filtering option, a functionality not included in the standard Seurat pipeline. Users can also adjust other parameters by modifying the configuration files, which are formatted in YAML. This flexibility empowers precise analysis of scRNA-seq data.

Cellsnake covers most of the methods offered by Seurat, including integration. The workflow is automatically repeated for an integrated dataset once the analyses have been concluded for all individual samples available in the study (i.e., QC, filtering etc.). Analysis outcomes such as dimension reduction, clustering, differential expression analysis, functional enrichment, and cell type annotations are reported for the integrated sample. Since the datasets individually passed the initial QC and are trimmed for

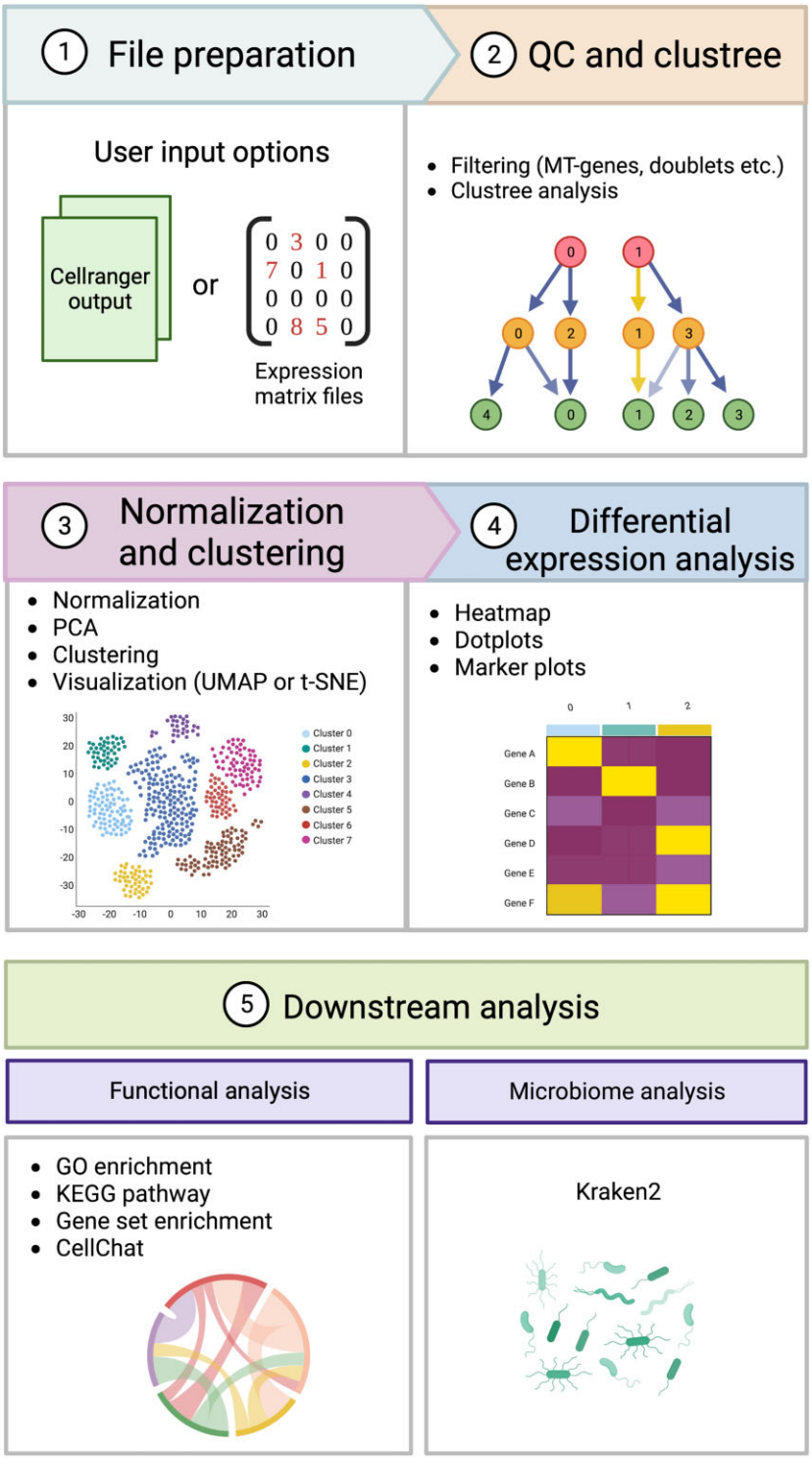

Figure 1: Overview of the scRNA-seq pipeline in cellsnake. (1) Cellsnake can accept the output files from Cellranger in addition to raw expression matrix files if provided in an appropriate format. (2) QC is performed by filtering out MT genes, doublets, and cells with a low gene number as examples. Clustree is then used to find the optimal resolution for the dimensionality reduction. (3) Afterward, the dataset is normalized and scaled before the principal component analysis and visualized by UMAP or t-distributed stochastic neighbor embedding. (4) To find the differences in gene expression levels within the dataset, differential gene expression analysis is performed with several outputs such as heatmaps, dot plots, and marker plots. (5) To get an even better insight into the dataset, the pipeline contains several functional analyses such as GO enrichment, KEGG pathway, gene set enrichment, and CellChat. Metagenome analysis is also available if the input file from step 1 is the direct output from Cellranger. This is done by using the metagenomics tool Kraken2.

Table 2: Cellsnake commands and a summary of their outputs

Mode*	Outputs	How to run?
cellsnake minimal	Dimension reduction plots, QC metrics, technical plots (MT, counts, gene, feature), clustree plot	`$ cellsnake minimal data` OR `$ snakemake -j 5 --config option=minimal`
cellsnake standard	All minimal outputs and CellTypist, singleR annotations, enrichment analyses tables, trajectory plots, summarized marker plots	`$ cellsnake standard data` OR `$ snakemake -j 5 --config option=standard`
cellsnake advanced	All standard outputs and CellChat results, detailed top markers per cluster plots	`$ cellsnake advanced data` OR `$ snakemake -j 5 --config option=advanced`
cellsnake integrate	A single integrated object for analysis	`$ cellsnake integrate data` OR `$ snakemake -j 5 --config option=integrate`

*Data folder may contain multiple samples and this will trigger a batch analysis.

artifacts, these steps are skipped by the workflow. Cellsnake can also generate publication-ready plots for both individual and integrated samples. It also automatically produces plots for markers (i.e., genes), which can be investigated to better understand the predicted clusters (i.e., cell subsets). Additionally, cellsnake provides the option to produce supplementary plots, featuring dimension reduction and expression images for selected genes or markers. This functionality adds a valuable level of customization to the analysis, enabling the user to explore targeted genes or markers of interest in greater detail.

The input of cellsnake can be either Cellranger output directories for batch analysis or single-expression matrix files (e.g., h5 files) for individual sample processing. Cellsnake automatically detects the input format and runs accordingly with minimal user intervention and with minimal lines of input commands (Table 2). The Cellsnake workflow offers 3 primary modes with distinct options: minimal, standard, and advanced. The minimal mode is suitable for fast analysis, parameter selection, and downstream integration. Fundamental parameters, such as filtering thresholds and clustering resolution, can be determined via a minimal run at an early stage, which will reduce computational cost. Standard and advanced workflow modes contain additional features and algorithms (Table 2).

Reanalyses of publicly available datasets using cellsnake

We showcase some features of the pipeline using publicly available datasets. The first dataset is from the fetal brain containing (only) count tables from 2 samples (Figs. 2 and 3). We processed 2 samples using the default settings (e.g., MT filtering threshold 10% and resolution parameter 0.8). Minimal mode only takes 4 minutes on a laptop for 2 samples of the fetal brain dataset. Another 5 minutes is enough for both integration and processing of the integrated sample with minimal mode. The user can decide on the parameters early on (Fig. 2), and the standard mode will finish in 50 minutes without parallel processing. Cellsnake utilizes different tools and provides outputs for all as figures (Supplementary Figs. 1–6) or as tables.

The second dataset is from the fetal liver containing 3 $CD45^+$ and 3 $CD45^-$ FACS-sorted samples from 3 different donors (Fig. 4A). This time, we selected automatic filtering of MT gene-abundant cells rather than a hard cutoff when preprocessing the samples. In total, 29,045 cells passed the filtering threshold. The standard workflow took 3 hours with only 2 CPU cores on a standard laptop, which is enough for most use cases. The samples were later integrated, and the optimal number of clusters was predicted automatically. The separation of 2 groups (Fig. 4A, B) in the integrated dataset is similar to what was reported in the original study [23], which indicates that cellsnake is capable of reproducing key findings from published studies. The differential expression analysis also reveals that the AHSP gene is highly expressed in $CD45^+$ samples, which is in line with the known function of this gene in erythroid cells.

Cellsnake can analyze metagenomics from single-cell data

Another unique feature of Cellsnake is its ability to perform metagenomics analysis using Kraken2. If a database is provided, Cellsnake will automatically run Kraken2. After collapsing read counts to a taxonomic level based on user input, such as genus or phylum, results are reported accordingly. Cellsnake provides metagenomic results in the form of dimension reduction plots and barplots, and users can load metadata into R for personalized downstream analysis.

This feature was tested on 4 samples from mucosal macrophages, with automatic trimming of MT genes and selection of resolution (Fig. 5). Cellsnake reported results based on the optimal number of clusters, and nonhuman material detected by Kraken2 is visualized on integrated Uniform Manifold Approximation and Projection (UMAP) plots (Fig. 5A, B). Users can also obtain a detailed list of results based on the selected taxonomic level in an Excel file.

Discussion

In recent years, there has been an increasing interest in scRNA-seq as it is a powerful technique for understanding the cellular heterogeneity of tissues and organs. However, the scRNA-seq data analysis can be complex and time-consuming. Cellsnake was designed to simplify this process, enabling researchers without extensive bioinformatics experience to easily analyze their data. It includes a range of automated preprocessing and downstream analysis tools and also provides advanced features for additional analysis. Its user-friendly interface and reproducibility features make it a valuable tool for researchers seeking to understand transcriptional heterogeneity in tissues at single-cell resolution.

Cellsnake has several critical functionalities for scRNA-seq data analysis. It includes preprocessing steps, such as QC, filtering, and parameter auto-selection, and also has downstream analysis tools for identifying differentially expressed genes, perform-

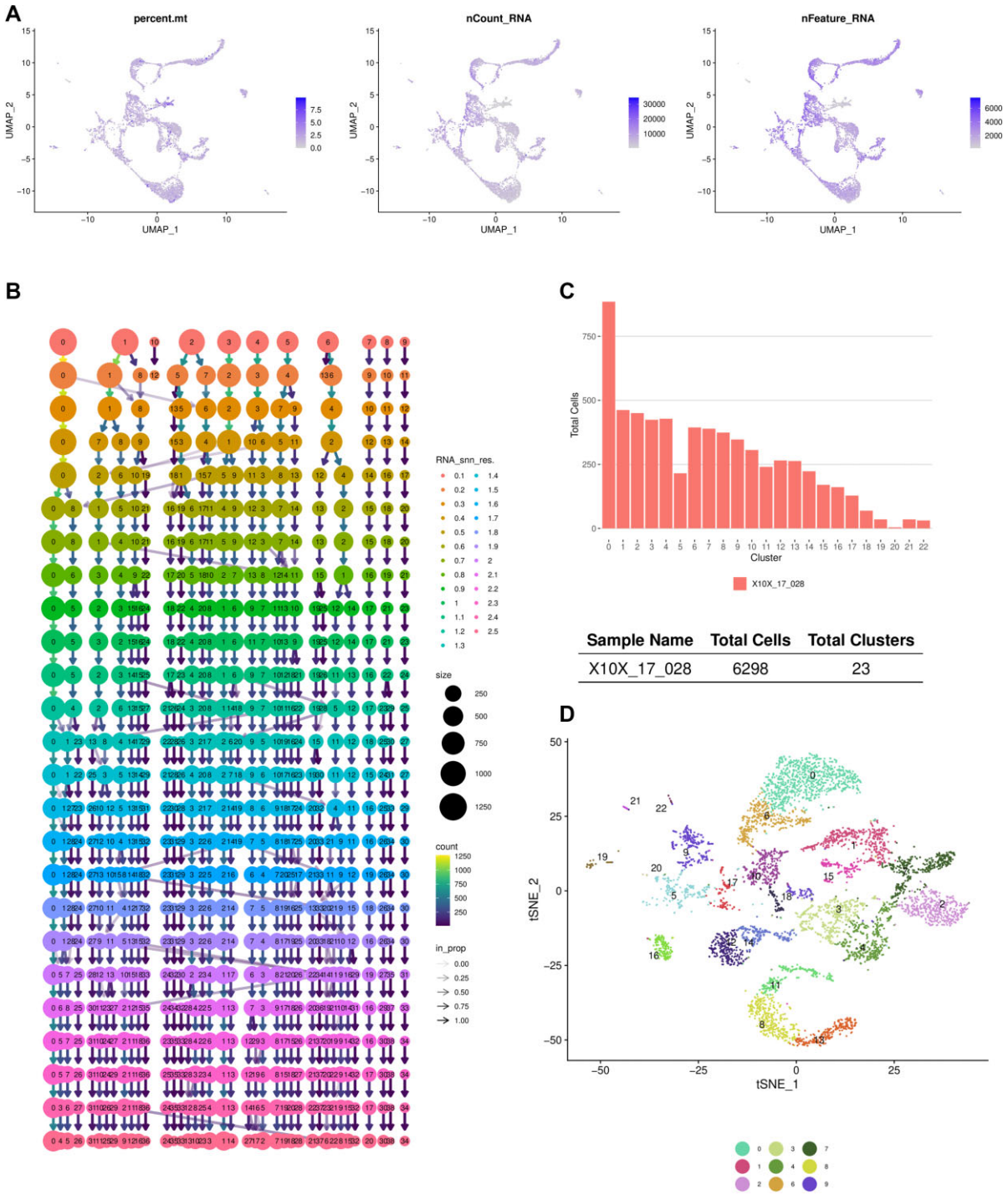

Figure 2: Cellsnake quickly generates standard output plots that include technical information. (A) The user can investigate the fundamental statistics like MT gene percentage, number of genes detected, and reads mapped per cell information. Here the results shown are based on one of the fetal brain samples. (B) Clustree analysis is not part of the Seurat pipeline, but cellsnake offers this by default. This plot can be used to find the optimal number of clusters. (C) The selected resolution resulted in 23 clusters and 6,298 cells passed the filtering thresholds (after filtering doublets and low-quality cells). (D) t-distributed stochastic neighbor embedding (tSNE) plot shows the clusters. Cellsnake prints only the top clusters in the legend to prevent overplotting. The user will get UMAP, principal component analysis, and tSNE plots by default.

ing clustering, visualization, and exploring cell type–specific gene expression patterns. These features are crucial for characterizing cell subpopulations and identifying specific genes and pathways associated with them. Cellsnake also includes advanced features such as supporting the integration of multiple scRNA-seq datasets to identify shared and unique cell types across different tissues or conditions. Cellsnake also ensures reproducibility by creating separate folders when required, restricting the versions of the tools in the environment, saving config files with the cellsnake version, explicitly sharing different images for each version in the Docker repository, and storing results for downstream analysis by default. In comparison to other tools (Table 3), cellsnake has several advantages, including a comprehensive range of tool utilization, unique features, the ability to run locally or on HPC

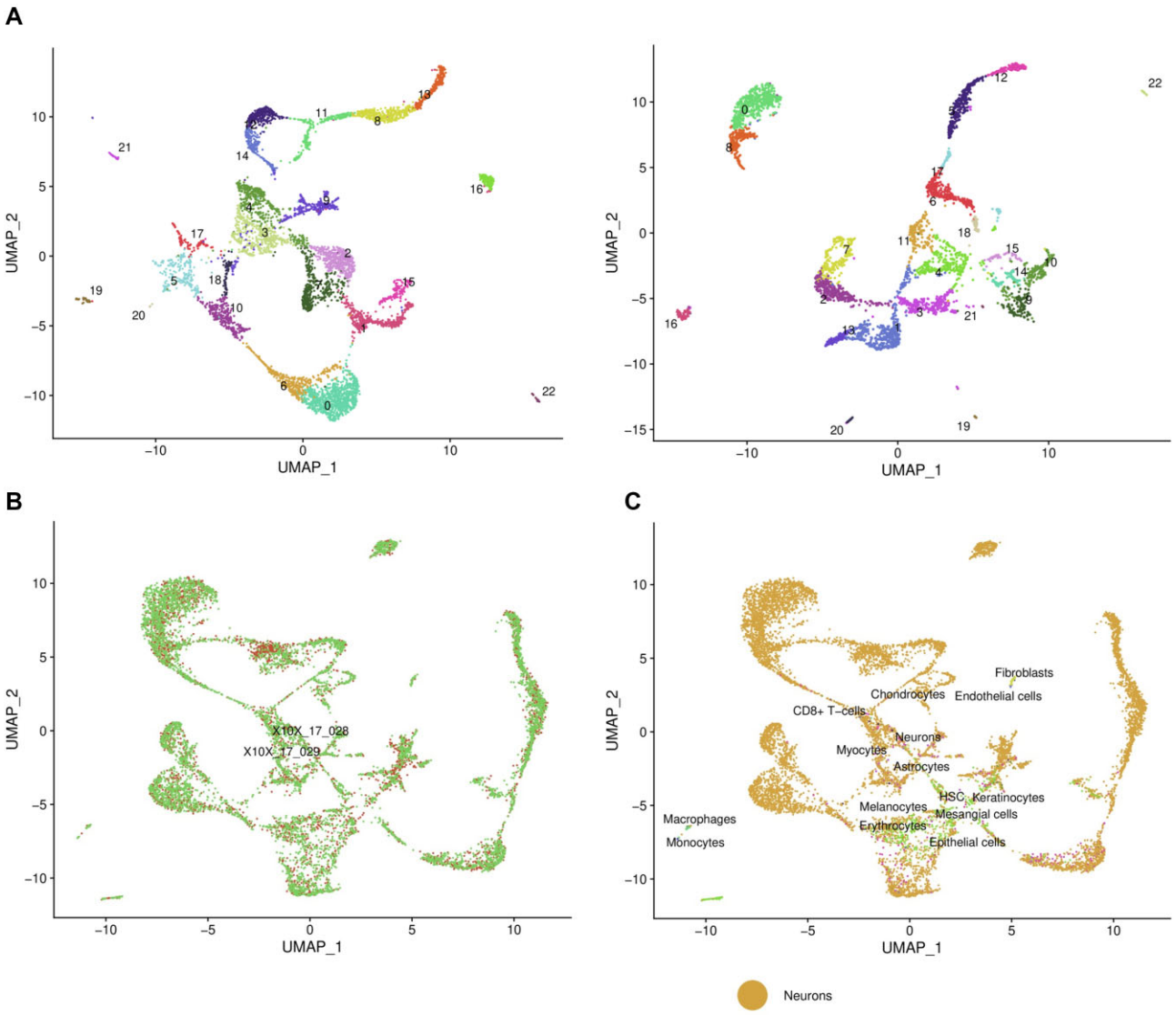

Figure 3: Cellsnake processes integrated samples similar to the individual samples and generates the same plots. (A) The UMAP plots were generated for 2 samples from the fetal brain dataset, seen in the first and second panels. (B) The UMAP plot shows clusters for the integrated samples. (C) The UMAP plot shows cluster annotation based on the singleR package "BlueprintEncodeData" model predictions. The results showed the cells were mostly predicted as neurons, which are consistent with the dataset, but there are also some mispredictions. The detailed annotations can be accessed as Excel tables and heatmaps.

platforms, and seamless integration with other workflows using Docker or Bioconda. Additionally, cellsnake also provides R data serialization (RDS) files to enhance data sharing and accessibility.

Recent studies have shown that the heterogeneity in microbiota and the present cell types along with their functions are codependent [33]. Cell-associated microbial reads can be identified in scRNA-seq data [34]. Cellsnake uses Kraken2 [29] to analyze these data, and cellsnake provides the ability to fine-tune parameters to increase sensitivity and/or specificity and to use personal databases. This can help researchers identify potential microbial associations with host cells and tissues. Some of these microbial hits can originate from environmental contamination or can be false positives. These outcomes might not necessarily reflect real biological associations; nevertheless, the results may provide valuable insights for QC such as recognizing potential contamination sources.

There are some limitations of the workflow that need to be addressed. First, cellsnake requires disk space to keep track of the entire pipeline, including metadata files that are required for advanced downstream analysis. Although the users can delete large files, they may want to keep metadata files for reproducing the results at a later time. Second, the fully featured workflow relies on Cellranger outputs from the 10X Genomics platform, which may not always be available. Even though cellsnake was designed and tested utilizing this platform, it can still use the count matrix files from other platforms, such as the fetal brain dataset. Third, while cellsnake has moderate performance in terms of memory and speed on standard workstations for an average number of cells, the auto-detection of parameters (e.g., resolution parameter) can be slow when processing samples with a large number of cells. To improve performance, a parallel version of the MultiK tool was used, which is not officially supported by the authors of MultiK (see Methods). Finally, the underlying tools utilized by cellsnake may involve various parameters. The fundamental parameters can be adjusted by the user and supplied through the configuration files, while the rest are set to default

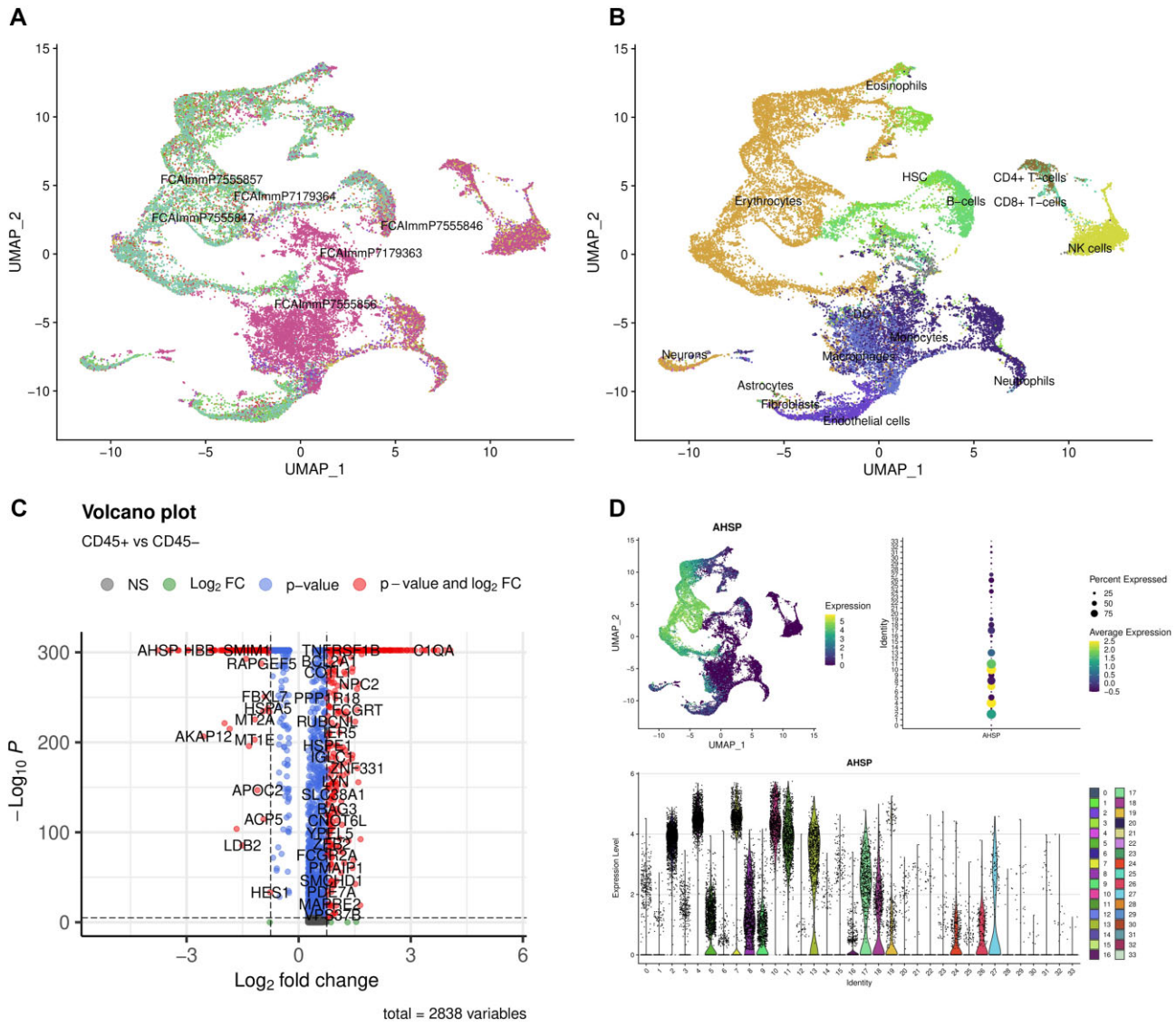

Figure 4: The fetal liver dataset consists of 6 FACS-sorted samples, integrated by cellsnake. (A) Cellsnake displays the integrated UMAP plot and labels and (B) annotates the clusters. (C) The user can provide the clinical information, which shows differentially expressed genes among 2 groups. (D) It is also possible to visualize selected marker genes. For example, the AHSP gene is upregulated in $CD45^{+}$ samples compared to $CD45^{-}$ samples.

values. This approach was preferred to make the workflow more user-friendly.

In conclusion, cellsnake is a convenient and adaptable tool, empowering researchers to analyze scRNA-seq data in a reproducible and customizable manner. With its advanced features and streamlined workflow, cellsnake stands as a valuable bioinformatics asset for investigating cellular heterogeneity and gene expression patterns at single-cell resolution within tissues.

Future Directions

Accurate bioinformatics software requires long-term development and commitment to the project [35]. It is also a major problem in the field that many projects are abandoned after publication, becoming unusable and outdated. For instance, cerebroApp [36], a component of cellsnake's development version, was dropped as it is no longer in active development. Cellsnake is an open-source tool that is actively developed, allowing anyone to open pull requests and report issues on its GitHub page. To keep the software bug-free and streamlined, future developments of cellsnake will involve incorporating new tools, such as the latest Seurat version, and removing obsolete tools from the main workflow. The users can access the previous releases for reproducibility. Although cellsnake is mainly designed for the 10X Genomics single-cell platform, we plan to expand its compatibility with other platforms and offer additional support for various input formats. Our aim is for cellsnake to become an essential toolkit for fast, accurate, tunable, and comprehensive scRNA data analysis.

Availability and Requirements

- Project name: cellsnake
- Project homepage: https://github.com/sinanugur/cellsnake [37]
- Documentation: https://cellsnake.readthedocs.io/en/latest/ [38]
- RRID:SCR_023666
- biotoolsID: cellsnake
- Operating system: Platform independent
- Programming language: Python, R

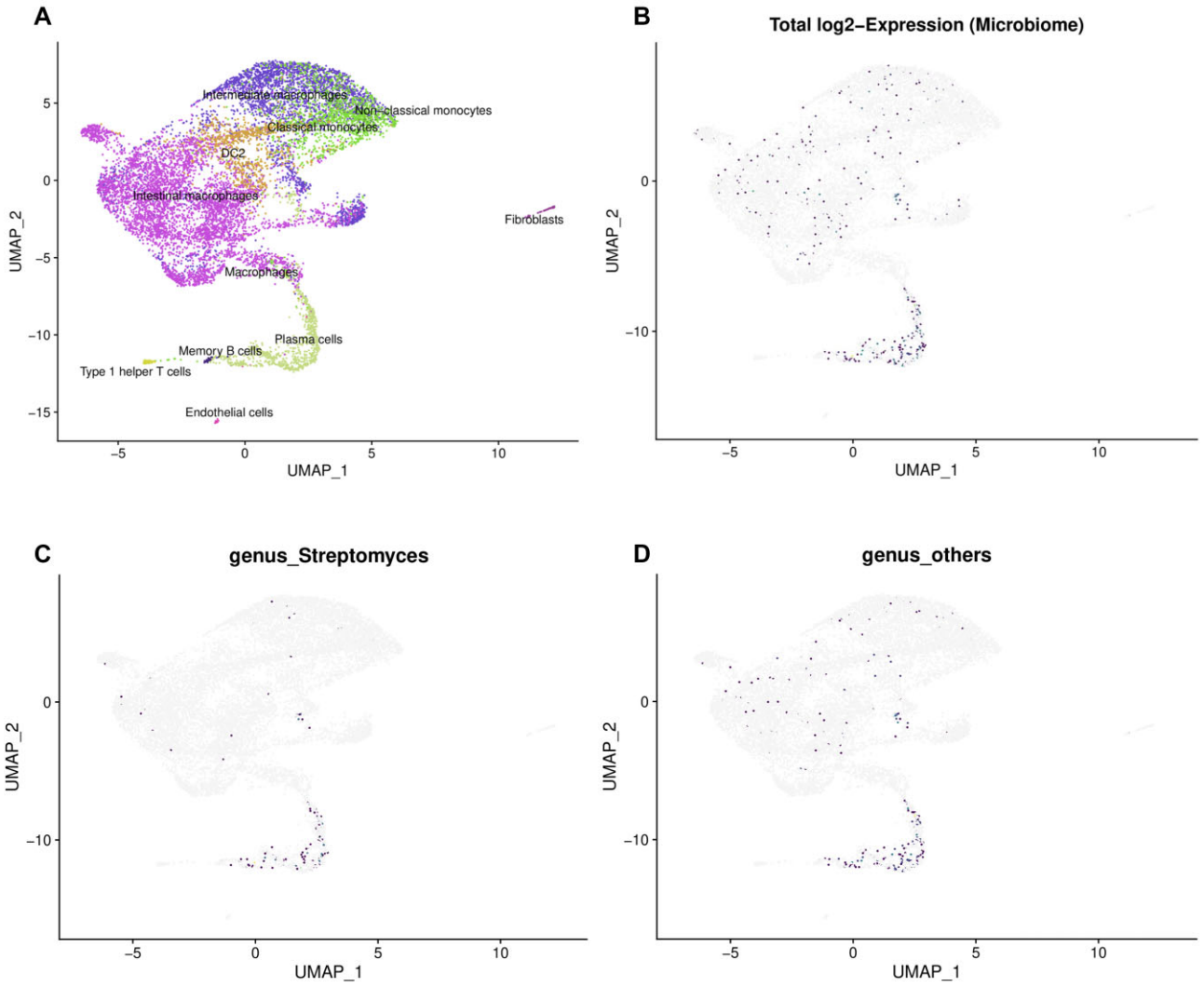

Figure 5: Cellsnake's metagenomics feature was tested on mucosal macrophages. (A) Four samples were integrated. The clusters were predicted and annotated using the CellTypist immune model. (B) The cells annotated as "plasma cells" contain the highest number of bacterial reads. (C) The foreign reads were mostly associated with *Streptomyces*. (D) Cellsnake reports the top 10 most prevalent taxonomic groups by default. The rest collapsed and were reported as "others." The user can select the desired taxonomic level (in this case, it was genus). All results are also saved as tables, which include reads detected per cluster and annotation.

Table 3: Standard features of cellsnake compared to available holistic tools/workflows

	Cellsnake	Cellenics	Single Cell Omics Workbench
Platform	Snakemake/Python wrapper/Docker	Web based	Web based (Galaxy)
Input file type	Count tables (10X or others), R Data File	Count tables (10X)	Count tables (10X), FASTQ and others
Doublet filtering	Yes	Yes	No
MT gene filtering	Yes (auto)	Yes (auto)	Yes
Find clusters	Yes (auto)	Yes	Yes
Clustree plot	Yes	No	No
Differential expression analysis	Yes	Yes	Yes
Enrichment analysis	KEGG and GO	No	No
Cell type annotation	Yes	Yes	No
Detailed gene expression plots	Yes	No	No
Metagenome analysis	Yes	No	No
Trajectory analysis	Yes	Yes	Yes
Integration	Yes (Seurat only)	Yes (various algorithms)	Yes
Output and downstream analysis	Plot files, expression tables, Seurat RDS files and Excel files, etc.	Plots and expression tables	Miscellaneous

- Other requirements: Python 3.8 or higher, R 4.2.2
- License: MIT
- PyPi: https://pypi.org/project/cellsnake [39]
- Bioconda: https://anaconda.org/bioconda/cellsnake [40]
- Docker: https://hub.docker.com/r/sinanugur/cellsnake [41]
- Snakemake workflow: https://github.com/sinanugur/scrna-workflow [42]

Additional Files

Supplementary Fig. S1. Summarized marker plots.
Supplementary Fig. S2. Heatmap plots showing clusters/markers.
Supplementary Fig. S3. SingleR annotation prediction plots.
Supplementary Fig. S4. Celltypist label transfer prediction plots.
Supplementary Fig. S5. CellChat prediction plots.
Supplementary Fig. S6. Monocle3 trajectory analysis.

Abbreviations

GO: Gene Ontology; HPC: high-performance computing; KEGG: Kyoto Encyclopedia of Genes and Genomes; MT: mitochondrial; QC: quality control; RDS: R data serialization; scRNA-seq: single-cell RNA sequencing; tSNE: t-distributed stochastic neighbor embedding; UMAP: Uniform Manifold Approximation and Projection; UMI: unique molecular identifier.

Authors' Contributions

S.U.U. devised the project; created the workflow, wrapper, and R scripts; and drafted the manuscript with input from all authors. K.R.V. contributed to the figures and the R scripts. V.T.K. contributed to the R scripts. M.C. contributed to the R scripts and revised the preliminary manuscript. E.S.B. revised the manuscript and acquired financial support. F.L.J. revised the manuscript and acquired financial support. D.D. supervised the project, contributed to the R scripts, revised the manuscript, and acquired financial support.

Funding

This work was supported by The Research Council of Norway (project number 315483).

Data Availability

The publicly available datasets for the fetal brain and liver are available under accessions PRJNA429950 and PRJEB34784, respectively. Macrophage-only samples from gut mucosal tissue are deposited in the European Genome-Phenome Archive (EGA) under the following accession number: EGAD00001007765. The EGA-deposited files are under controlled access, requiring the data access committee permission for retrieval. The cellsnake analysis results on test samples are available at Zenodo [43]. A copy of the fetal brain dataset can also be found in our frozen Zenodo repository [43]. Other data further supporting this work are openly available in the *GigaScience* repository, GigaDB [44]. The analysis results are completely anonymized and shared in a way that is compliant with the Data Access Agreement declared at EGA for dataset EGAD00001007765.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

Cellsnake was tested on the Educloud high-performance computing platform of the University of Oslo. We thank all Educloud staff for their support and services.

References

1. Saliba A-E, Westermann AJ, Gorski SA, et al. Single-cell RNA-seq: advances and future challenges. Nucleic Acids Res 2014;42:8845–60. https://doi.org/10.1093/nar/gku555.
2. Jovic D, Liang X, Zeng H, et al. Single-cell RNA sequencing technologies and applications: a brief overview. Clin Transl Med 2022;12:e694. https://doi.org/10.1002/ctm2.694.
3. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. Genome Biol 2016;17:63. https://doi.org/10.1186/s13059-016-0927-y.
4. Nayak R, Hasija Y. A hitchhiker's guide to single-cell transcriptomics and data analysis pipelines. Genomics 2021;113:606–19. https://doi.org/10.1016/j.ygeno.2021.01.007.
5. Lähnemann D, Köster J, Szczurek E, et al. Eleven grand challenges in single-cell data science. Genome Biol 2020;21:31. https://doi.org/10.1186/s13059-020-1926-6.
6. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol 2018;19:15. https://doi.org/10.1186/s13059-017-1382-0.
7. Amezquita RA, Lun ATL, Becht E, et al. Orchestrating single-cell analysis with Bioconductor. Nat Methods 2020;17:137–45. https://doi.org/10.1038/s41592-019-0654-x.
8. Hao Y, Hao S, Andersen-Nissen E, et al. Integrated analysis of multimodal single-cell data. Cell 2021;184:3573–87. e29. https://doi.org/10.1016/j.cell.2021.04.048.
9. Satija R, Farrell JA, Gennert D, et al. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 2015;33:495–502. https://doi.org/10.1038/nbt.3192.
10. Prieto C, Barrios D, Villaverde A. SingleCAnalyzer: interactive analysis of single cell RNA-seq data on the cloud. Front Bioinform 2022;2:793309. https://doi.org/10.3389/fbinf.2022.793309.
11. Choi J-H, In Kim H, Woo HG. scTyper: a comprehensive pipeline for the cell typing analysis of single-cell RNA-seq data. BMC Bioinf 2020;21:342. https://doi.org/10.1186/s12859-020-03700-5.
12. Tekman M., et al. A single-cell RNA-seq training and analysis suite using the galaxy framework. Biorxiv. 2020. https://doi.org/10.1101/2020.06.06.137570.
13. Zheng GXY, Terry JM, Belgrader P, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;8:14049. https://doi.org/10.1038/ncomms14049.
14. Mölder F, Jablonski KP, Letcher B, et al. Sustainable data analysis with Snakemake. F1000Res 2021;10:33. https://doi.org/10.12688/f1000research.29032.2.
15. Zappia L, Oshlack A. Clustering trees: a visualization for evaluating clusterings at multiple resolutions. Gigascience 2018;7:giy083. https://doi.org/10.1093/gigascience/giy083.
16. Domínguez Conde C, Xu C, Jarvis LB, et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science 2022;376:eabl5197. https://doi.org/10.1126/science.abl5197.
17. hbctraining-team. Introduction to single-cell RNA-seq. hbctraining.github.io/scRNA-seq/lessons/elbow_plot_metric.html. Accessed 4 October 2023.
18. Hippen AA, Falco MM, Weber LM, et al. miQC: an adaptive probabilistic framework for quality control of single-cell RNA-sequencing data. PLoS Comput Biol 2021;17:e1009290. https://doi.org/10.1371/journal.pcbi.1009290.

19. Liu S, Thennavan A, Garay JP, et al. MultiK: an automated tool to determine optimal cluster numbers in single-cell RNA sequencing data. Genome Biol 2021;22:232. https://doi.org/10.1186/s13059-021-02445-5.
20. McGinnis CS, Murrow LM, Gartner ZJ. DoubletFinder: doublet detection in single-cell RNA sequencing data using artificial nearest neighbors. Cell Syst 2019;8:329–37.e4. https://doi.org/10.1016/j.cels.2019.03.003.
21. Domanska D, Majid U, Karlsen VT, et al. Single-cell transcriptomic analysis of human colonic macrophages reveals niche-specific subsets. J Exp Med 2022;219:e20211846. https://doi.org/10.1084/jem.20211846.
22. La Manno G, Soldatov R, Zeisel A, et al. RNA velocity of single cells. Nature 2018;560:494–8. https://doi.org/10.1038/s41586-018-0414-6.
23. Popescu D-M, Botting RA, Stephenson E, et al. Decoding human fetal liver haematopoiesis. Nature 2019;574:365–71. https://doi.org/10.1038/s41586-019-1652-y.
24. Cellenics. https://scp.biomage.net/. Accessed 4 October 2023.
25. Biomage. https://biomage.net/ Accessed 4 October 2023.
26. Xi NM, Li JJ. Benchmarking computational doublet-detection methods for single-cell RNA sequencing data. Cell Syst 2021;12:176–94.e6. https://doi.org/10.1016/j.cels.2020.11.008.
27. Seurat Disk. github.com/mojaveazure/seurat-disk/. Accessed 4 October 2023.
28. Aran D, Looney AP, Liu L, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nat Immunol 2019;20:163–72. https://doi.org/10.1038/s41590-018-0276-y.
29. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biol 2019;20:257. https://doi.org/10.1186/s13059-019-1891-0.
30. Jin S, Guerrero-Juarez CF, Zhang L, et al. Inference and analysis of cell-cell communication using CellChat. Nat Commun 2021;12:1088. https://doi.org/10.1038/s41467-021-21246-9.
31. Yu G, Wang L-G, Han Y, et al. clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS 2012;16:284–7. https://doi.org/10.1089/omi.2011.0118.
32. Cao J, Spielmann M, Qiu X, et al. The single-cell transcriptional landscape of mammalian organogenesis. Nature 2019;566:496–502. https://doi.org/10.1038/s41586-019-0969-x.
33. Mahmoudabadi G,Tabula Sapiens Consortium, Quake SR. Single cell transcriptomics reveals the hidden microbiomes of human tissues. Biorxiv. 2022. https://doi.org/10.1101/2022.10.11.511790.
34. Galeano Niño JL, Wu H, LaCourse KD, et al. Effect of the intratumoral microbiota on spatial and cellular heterogeneity in cancer. Nature 2022;611:810–7. https://doi.org/10.1038/s41586-022-05435-0.
35. Gardner PP, Paterson JM, McGimpsey S, et al. Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software. Genome Biol 2022;23:56. https://doi.org/10.1186/s13059-022-02625-x.
36. Hillje R, Pelicci PG, Luzi L. Cerebro: interactive visualization of scRNA-seq data. Bioinformatics 2020;36:2311–3. https://doi.org/10.1093/bioinformatics/btz877.
37. Cellsnake main GitHub repository. https://github.com/sinanugur/cellsnake. Accessed 4 October 2023.
38. Cellsnake documentation. https://cellsnake.readthedocs.io/en/latest/. Accessed 4 October 2023.
39. Cellsnake PyPi repository. https://pypi.org/project/cellsnake. Accessed 4 October 2023.
40. Cellsnake Bioconda repository. 2023. bioconda. https://anaconda.org/bioconda/cellsnake. Accessed 4 Ocobert 2023.
41. Cellsnake Docker Hub. 2023. Docker Hub. https://hub.docker.com/r/sinanugur/cellsnake. Accessed 4 October 2023.
42. Cellsnake workflow GitHub repository. 2023. https://github.com/sinanugur/scrna-workflow. Accessed 4 October 2023.
43. Umu S. 2023 cellsnake: a user-friendly tool for single-cell RNA sequencing analysis. Zenodo. https://doi.org/10.5281/zenodo.8282676
44. Umu SU, Vander-Elst KR, Karlsen VT, et al. Supporting data for "Cellsnake: A User-Friendly Tool for Single-Cell RNA Sequencing Analysis." GigaScience Database. 2023. https://doi.org/10.5524/102453.

Received: June 1, 2023. **Revised:** August 25, 2023. **Accepted:** October 5, 2023