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 Institute of Clinical Medicine, University of Oslo, Oslo, Norway
2 Department of Pathology, Oslo University Hospital-Rikshospitalet, Oslo, Norway
3 Institute of Oral Biology, University of Oslo, Oslo, Norway
4 Department of Microbiology, University of Oslo, Rikshospitalet, Oslo, Norway

*corresponding author: sinanuu@uio.no

Abstract

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

Introduction

Single-cell RNA sequencing (scRNA-seq) is a method used to study gene expression at the single cell level. This is in contrast to bulk RNA sequencing, which only provides information on average transcript expression in 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 (Saliba *et al.*, 2014; Bacher and Kendziorski, 2016; Nayak and Hasija, 2021; Jovic *et al.*, 2022).

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 (Lähnemann *et al.*, 2020; Nayak and Hasija, 2021). 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 (Wolf, Angerer and Theis, 2018; Amezquita *et al.*, 2020; Hao *et al.*, 2021; Jovic *et al.*, 2022). Popular workflows like Seurat (Satija *et al.*, 2015), SingleCellExperiment (of Bioconductor) (Amezquita *et al.*, 2020) and Scanpy (Wolf, Angerer and Theis, 2018) 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 (Lähnemann *et al.*, 2020). 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 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 (Prieto, Barrios and Villaverde, 2022), the R package scTyper (Choi, In Kim and Woo, 2020) and the web application Cellenics (proprietary software of Biomage). Cellranger from 10x genomics also provides dataset clustering and basic differential expression analysis (Zheng *et al.*, 2017) 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 and 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 resolution, doublet filtering, visualization of marker genes, enrichment analysis and pathway analysis. Cellsnake also allows parallelization and readily utilizes high performance computing (HPC) platforms. In addition to that, cellsnake provides metagenome analysis capabilities 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.

Results

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

Cellsnake utilizes a variety of tools and algorithms (Table 1) and it is composed of two main parts: 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) was primarily designed using the Seurat pipeline and the Snakemake workflow manager. When necessary, the workflow integrates various algorithms to enhance the basic functionality of Seurat. For instance, when one droplet encapsulates more than one cell, it appears as a single cell and can affect the downstream analysis. Therefore, this issue must be addressed early in a workflow (Xi and Li, 2021). One unique feature of cellsnake is its default doublet filtering option, which is 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 enables 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 individual samples. Analysis results 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 artifacts, those 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 single sample processing. Cellsnake automatically detects the input format and runs accordingly with minimal user intervention and with minimal lines of input commands (Table 2). Cellsnake workflow offers three main modes with different 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).

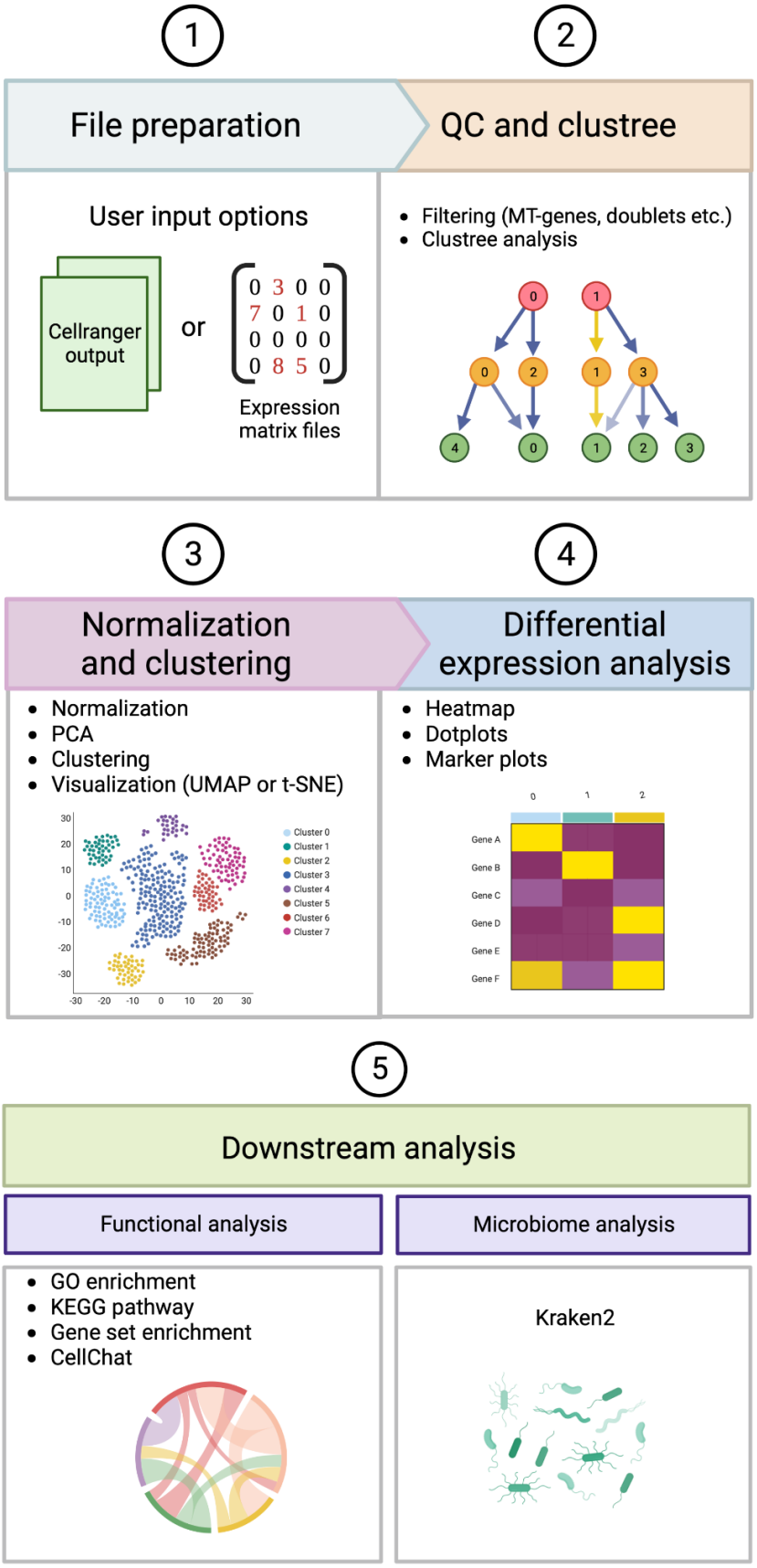

Figure 1. Overview of the scRNA-seq pipeline in cellsnake. (1) Cellsnake can accept the output files from cellranger in addition to raw expression matrix files if provided in an

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 PCA analysis and visualized by UMAP or t-SNE. (4) To find the differences in gene expression levels within the dataset differential gene expression analysis is performed with several outputs such as heatmaps, dotplots 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 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	(Hao *et al.*, 2021)	Main analysis platform
SeuratDisk	0.9020	github.com/mojaveazure/seurat-disk/	Format converter
Clustree	0.5.0	(Zappia and Oshlack, 2018)	Clustering interrogation
MultiK	1.0	(Liu *et al.*, 2021)	Optimal cluster detection
miQC	1.6.0	(Hippen *et al.*, 2021)	Auto MT gene trimming
DoubletFinder	2.0.3	(McGinnis, Murrow and Gartner, 2019)	Doublet detection
SingleR	2.0.0	(Aran *et al.*, 2019)	Cell type annotation
CellTypist	1.2.0	(Domínguez Conde *et al.*, 2022)	Cell type annotation
Kraken2	2.1.2	(Wood, Lu and Langmead, 2019)	Metagenomics
CellChat	1.6.1	(Jin *et al.*, 2021)	Ligand-receptor analysis and miscellaneous
clusterProfiler	4.4.4	(Yu *et al.*, 2012)	KEGG, GO and module enrichment
Monocle3	1.0.0	(Cao *et al.*, 2019)	Cell trajectory and velocity

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,	`cellsnake minimal data`

	feature), clustree plot	
cellsnake standard	All minimal outputs and CellTypist, singleR annotations, enrichment analyses tables, trajectory plots, summarized marker plots	`cellsnake standard data`
cellsnake advanced	All standard outputs and CellChat results, detailed top markers per cluster plots	`cellsnake advanced data`
cellsnake integrate	A single integrated object for analysis.	`cellsnake integrate data`

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

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 two samples (Fig. 2). We processed two samples using the default settings (e.g. MT filtering threshold 10 percent and resolution parameter 0.8). Minimal mode only takes four minutes in a laptop for two samples of the fetal brain dataset. Another five minutes is enough for both integration and processing of the integrated sample with minimal mode. The user can decide the parameters and a standard mode will finish in 50 minutes without parallel processing.

The second dataset is from the fetal liver containing three CD45+ and three CD45- FACS sorted samples from three different donors (Fig. 3A). This time we selected automatic filtering of MT gene abundant cells rather than a hard cut off when pre-processing the samples. In total, 29045 cells passed the filtering threshold. The standard workflow took 3 hours with only two cpu cores on a standard laptop, which is enough for most use cases. The samples were later integrated and the optimal number of clusters were predicted automatically. The separation of two groups (Fig. 3A and 3B) in the integrated dataset is similar to what was reported in the original study (Popescu *et al.*, 2019), 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.

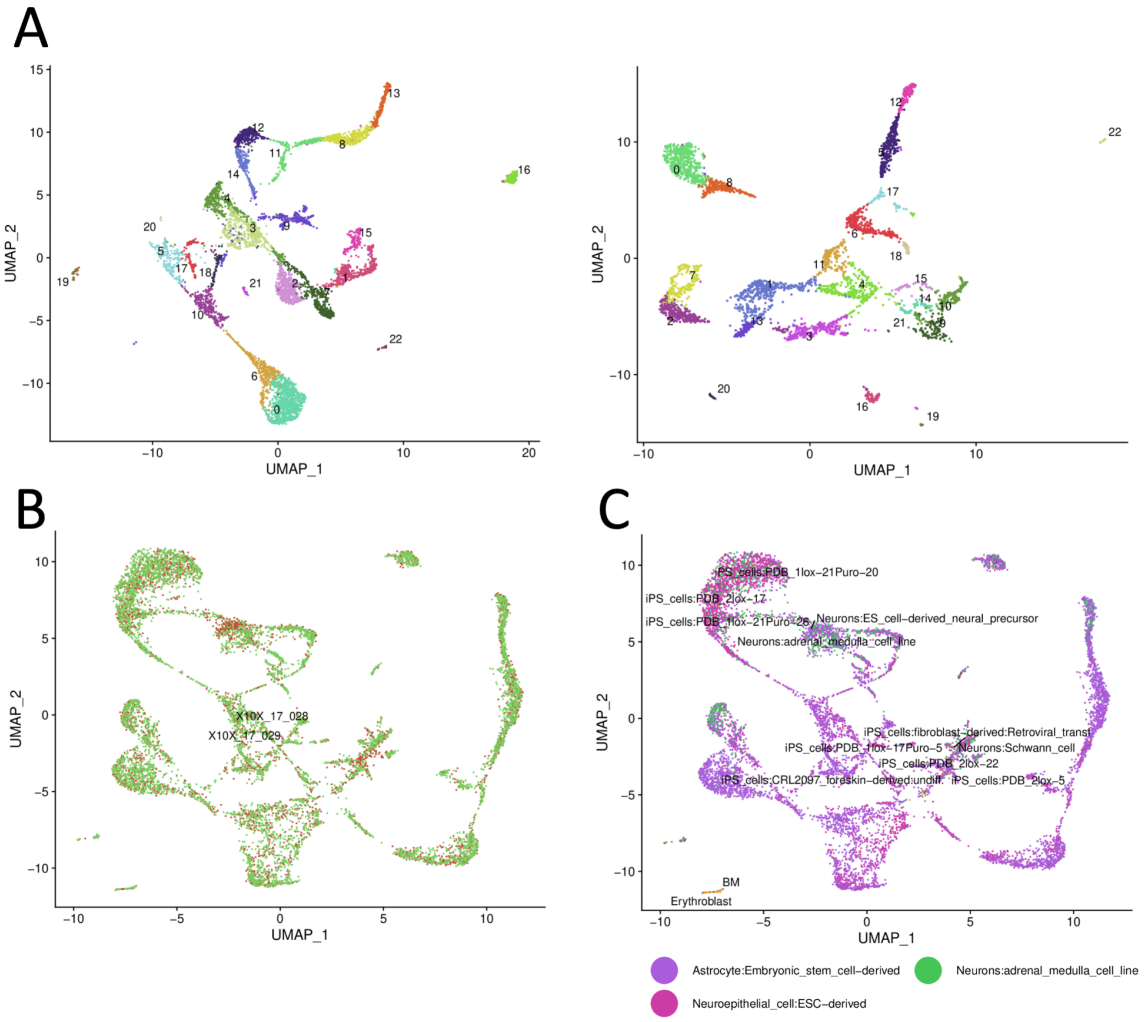

Figure 2. The fetal brain dataset: cellsnake quickly generates standard output plots that include clustering and technical information. Besides visualization, the results are written as Excel tables. A) The UMAP plots were generated for two samples from the fetal brain dataset, seen in the first and second panels. B) The UMAP plot shows clusters for the integrated samples. C) The UMAP plot shows cluster annotation based on the singleR package “HumanPrimaryCellAtlasData” model predictions. The results showed the cells mostly predicted as neurons.

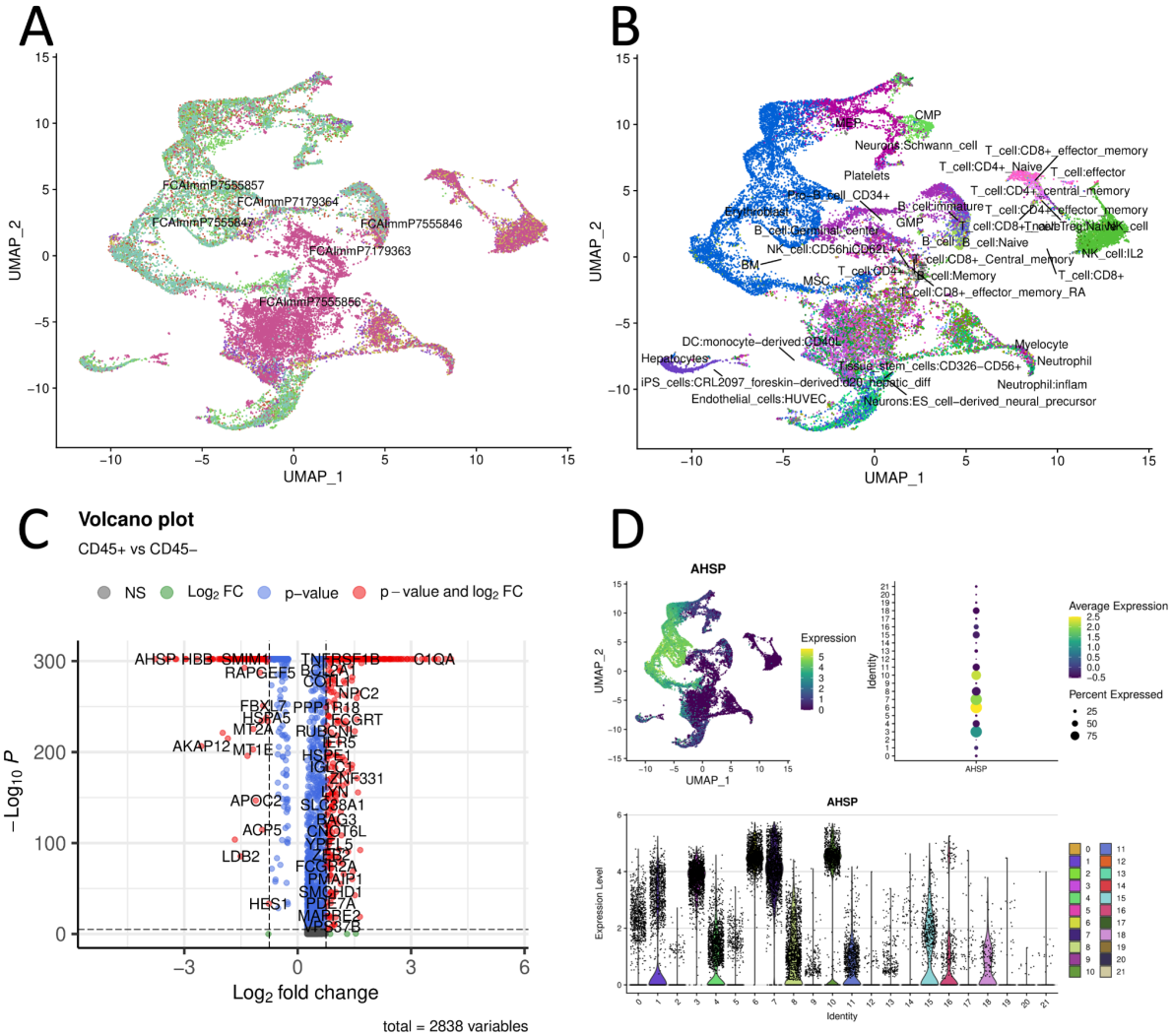

Figure 3. The fetal liver dataset consists of six FACS sorted samples: A) Cellsnake successfully integrates six samples originating from the fetal liver and B) annotates the clusters. C) The user can provide the clinical information which shows differentially expressed genes. D) It is also possible to visualize selected marker genes. For example, the AHSP gene is upregulated in CD45+ samples compared to CD45- samples.

Cellsnake can analyze metagenomics from single cell data

Another 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 four samples from mucosal macrophages, with automatic trimming of MT genes and selection of resolution (Fig. 4). Cellsnake reported results based

on the optimal number of clusters, and non-human material detected by Kraken2 is visualized on integrated UMAP plots (Fig. 4A,B). Users can also obtain a detailed list of results based on the selected taxonomic level in an Excel file.

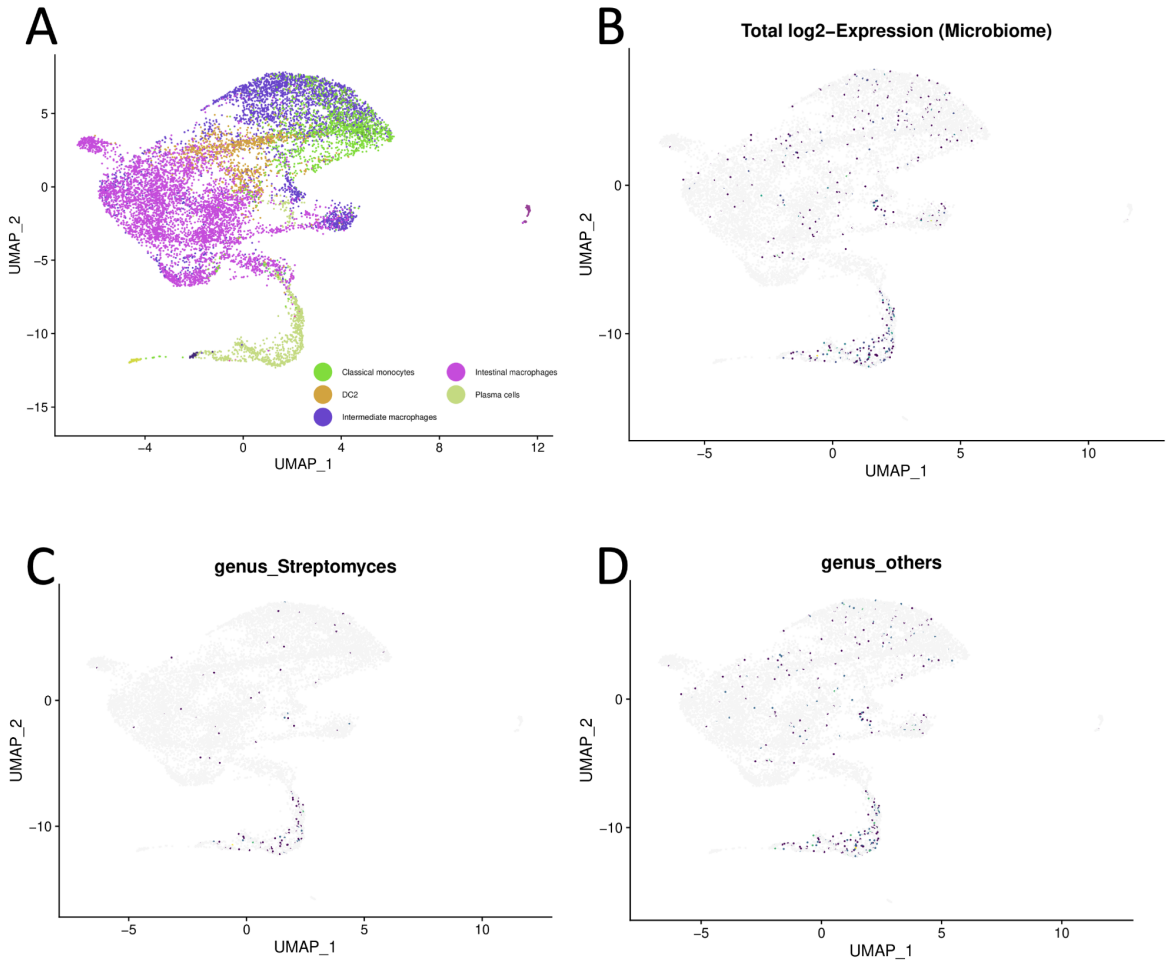

Figure 4. We tested cellsnake's metagenomics feature on mucosal macrophages. A) Four samples were merged and the clusters were predicted and were annotated using the CellTypist immune model. Some of the cells were annotated as plasma cells. B) The cells annotated as "plasma cells" contain the highest 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 are collapsed and reported as "others." The user can select the desired taxonomic level (in this case, it was genus).

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, scRNA-seq data analysis can be complex and time-consuming, cellsnake was designed to simplify this process, allowing 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, performing 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.

Recent studies have shown that the heterogeneity in microbiota and the present cell types along with their functions are co-dependent (Mahmoudabadi, Tabula Sapiens Consortium and Quake, 2022). Cell-associated microbial reads can be identified in scRNA-seq data (Galeano Niño *et al.*, 2022). Cellsnake uses Kraken2 (Wood, Lu and Langmead, 2019) to analyze this 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. They do not necessarily reflect real biological associations but the results may provide valuable insights for QC (e.g. to determine contamination in the lab).

There are some limitations of the workflow that need to be addressed. Firstly, 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. Secondly, the fully-featured workflow relies on Cellranger outputs from 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 materials and 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 were set to default values. This approach was preferred to make the workflow more user-friendly.

In conclusion, cellsnake is a convenient and adaptable tool that enables researchers to analyze scRNA-seq data in a reproducible and customizable manner. With its advanced features and streamlined workflow, cellsnake is a valuable bioinformatics resource for

investigating cellular heterogeneity and gene expression patterns at single-cell resolution in tissues.

Future Directions

Accurate bioinformatics software requires long-term development and commitment to the project (Gardner *et al.*, 2022). It is also a major problem in the field that many projects are abandoned after publication, becoming unusable and outdated. For instance, cerebroApp (Hillje, Pelicci and Luzi, 2020), 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 to other platforms and offer additional support for various input formats. Our goal is for cellsnake to become an essential toolkit for fast, accurate, tunable and comprehensive scRNA data analysis.

Methods

Cellsnake workflow and tools

The cellsnake wrapper was written in Python, while the main workflow was implemented in Snakemake. To detect optimal cluster resolution, we utilized clustree (Zappia and Oshlack, 2018). Seurat analysis pipeline (Hao *et al.*, 2021) 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 (Domínguez Conde *et al.*, 2022) requires AnnData format, and the workflow converts the files back to the required file format in R. By default, cellsnake stores files into two 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 some fundamental parameters such as min.cells or min.features, and also non-default parameters can be provided using a YAML file and a YAML file template can be printed and edited. To filter MT genes, cellsnake uses the miQC tool (Hippen *et al.*, 2021). If that fails, it uses the median absolute deviation of the MT gene expression as an alternative. MultiK algorithm (Liu *et al.*, 2021) is used to determine optimal resolution detection and doublet filtering is done using the DoubletFinder tool (McGinnis, Murrow and Gartner, 2019). 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 the results will be saved in different folders named after the selected parameters. These results are not overwritten and can be reviewed later, or the parameters can be changed for further investigation.

Cellsnake testing and benchmarks
To test the effectiveness of cellsnake, we obtained four macrophage-only samples from gut mucosal tissue (Domanska *et al.*, 2022), as well as two fetal brain datasets (La Manno *et al.*, 2018) and six fetal liver datasets (Popescu *et al.*, 2019). The fetal brain datasets were provided in matrix file format, while the other datasets were in FASTQ format and were processed by cellranger (v.7.0.0) using the default settings and the default databases.

Data availability
The publicly available datasets for the fetal brain and liver are available under accessions SRP129388 and E-MTAB-7407, respectively. Macrophage-only samples from gut mucosal tissue are deposited in the European Genome-Phenome Archive under the following accession numbers: EGAD00001007765 and EGAS00001005377. The cellsnake analysis results and link to a Dropbox folder is available on our GitHub repository.

Code availability
The full source code and the pipeline is available on GitHub (github.com/sinanugur/cellsnake). Cellsnake is also available as a Bioconda package (anaconda.org/bioconda/cellsnake) and a Docker image (sinanugur/cellsnake:latest).

Author contributions
SUU and DD devised the project. SUU created the snakemake workflow and wrapper. SUU, KRV, VTK, MC and DD wrote the R scripts. SUU drafted the manuscript with KRV, VTK, MC and DD in consultation with ESB and FLJ. ESB, FLJ and DD acquired the financial support.

Competing interests
None declared.

References

Amezquita, R.A. *et al.* (2020) ‘Orchestrating single-cell analysis with Bioconductor’, *Nature methods*, 17(2), pp. 137–145.

Aran, D. *et al.* (2019) ‘Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage’, *Nature immunology*, 20(2), pp. 163–172.

Bacher, R. and Kendziorski, C. (2016) ‘Design and computational analysis of single-cell RNA-sequencing experiments’, *Genome biology*, 17, p. 63.

Cao, J. *et al.* (2019) ‘The single-cell transcriptional landscape of mammalian organogenesis’, *Nature*, 566(7745), pp. 496–502.

Choi, J.-H., In Kim, H. and Woo, H.G. (2020) ‘scTyper: a comprehensive pipeline for

the cell typing analysis of single-cell RNA-seq data', *BMC bioinformatics*, 21(1), p. 342.

Domanska, D. *et al.* (2022) 'Single-cell transcriptomic analysis of human colonic macrophages reveals niche-specific subsets', *The Journal of experimental medicine*, 219(3). Available at: https://doi.org/10.1084/jem.20211846.

Domínguez Conde, C. *et al.* (2022) 'Cross-tissue immune cell analysis reveals tissue-specific features in humans', *Science*, 376(6594), p. eabl5197.

Galeano Niño, J.L. *et al.* (2022) 'Effect of the intratumoral microbiota on spatial and cellular heterogeneity in cancer', *Nature*, 611(7937), pp. 810–817.

Gardner, P.P. *et al.* (2022) 'Sustained software development, not number of citations or journal choice, is indicative of accurate bioinformatic software', *Genome biology* [Preprint]. Available at: https://doi.org/10.1186/s13059-022-02625-x.

Hao, Y. *et al.* (2021) 'Integrated analysis of multimodal single-cell data', *Cell*, 184(13), pp. 3573–3587.e29.

Hillje, R., Pelicci, P.G. and Luzi, L. (2020) 'Cerebro: interactive visualization of scRNA-seq data', *Bioinformatics* , 36(7), pp. 2311–2313.

Hippen, A.A. *et al.* (2021) 'miQC: An adaptive probabilistic framework for quality control of single-cell RNA-sequencing data', *PLoS computational biology*, 17(8), p. e1009290.

Jin, S. *et al.* (2021) 'Inference and analysis of cell-cell communication using CellChat', *Nature communications*, 12(1), p. 1088.

Jovic, D. *et al.* (2022) 'Single-cell RNA sequencing technologies and applications: A brief overview', *Clinical and translational medicine*, 12(3), p. e694.

Lähnemann, D. *et al.* (2020) 'Eleven grand challenges in single-cell data science', *Genome biology*, 21(1), p. 31.

La Manno, G. *et al.* (2018) 'RNA velocity of single cells', *Nature*, 560(7719), pp. 494–498.

Liu, S. *et al.* (2021) 'MultiK: an automated tool to determine optimal cluster numbers in single-cell RNA sequencing data', *Genome biology*, 22(1), p. 232.

Mahmoudabadi, G., Tabula Sapiens Consortium and Quake, S.R. (2022) 'Single Cell Transcriptomics Reveals the Hidden Microbiomes of Human Tissues', *bioRxiv*. Available at: https://doi.org/10.1101/2022.10.11.511790.

McGinnis, C.S., Murrow, L.M. and Gartner, Z.J. (2019) 'DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors', *Cell systems*, 8(4), pp. 329–337.e4.

Nayak, R. and Hasija, Y. (2021) 'A hitchhiker's guide to single-cell transcriptomics and data analysis pipelines', *Genomics*, 113(2), pp. 606–619.

Popescu, D.-M. *et al.* (2019) ‘Decoding human fetal liver haematopoiesis’, *Nature*, 574(7778), pp. 365–371.

Prieto, C., Barrios, D. and Villaverde, A. (2022) ‘SingleCAnalyzer: Interactive Analysis of Single Cell RNA-Seq Data on the Cloud’, *Frontiers in bioinformatics*, 2, p. 793309.

Saliba, A.-E. *et al.* (2014) ‘Single-cell RNA-seq: advances and future challenges’, *Nucleic Acids Research*, pp. 8845–8860. Available at: https://doi.org/10.1093/nar/gku555.

Satija, R. *et al.* (2015) ‘Spatial reconstruction of single-cell gene expression data’, *Nature biotechnology*, 33(5), pp. 495–502.

Wolf, F.A., Angerer, P. and Theis, F.J. (2018) ‘SCANPY: large-scale single-cell gene expression data analysis’, *Genome biology*, 19(1), p. 15.

Wood, D.E., Lu, J. and Langmead, B. (2019) ‘Improved metagenomic analysis with Kraken 2’, *Genome biology*, 20(1), p. 257.

Xi, N.M. and Li, J.J. (2021) ‘Benchmarking Computational Doublet-Detection Methods for Single-Cell RNA Sequencing Data’, *Cell systems*, 12(2), pp. 176–194.e6.

Yu, G. *et al.* (2012) ‘clusterProfiler: an R package for comparing biological themes among gene clusters’, *Omics: a journal of integrative biology*, 16(5), pp. 284–287.

Zappia, L. and Oshlack, A. (2018) ‘Clustering trees: a visualization for evaluating clusterings at multiple resolutions’, *GigaScience*, 7(7). Available at: https://doi.org/10.1093/gigascience/giy083.

Zheng, G.X.Y. *et al.* (2017) ‘Massively parallel digital transcriptional profiling of single cells’, *Nature communications*, 8, p. 14049.