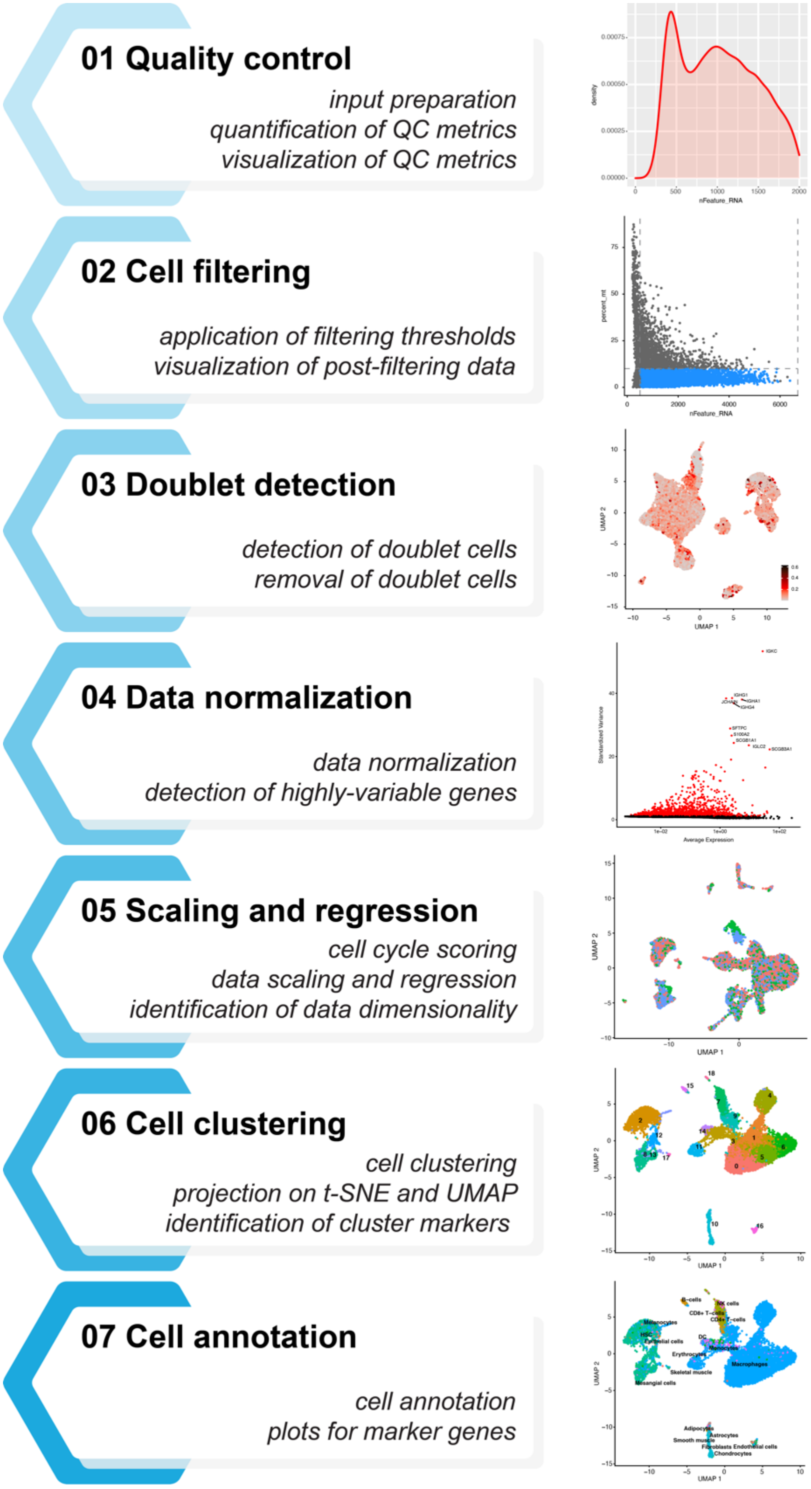
Graphical Abstract:



01 Quality control

input preparation
quantification of QC metrics
visualization of QC metrics

02 Cell filtering

application of filtering thresholds
visualization of post-filtering data

03 Doublet detection

detection of doublet cells
removal of doublet cells

04 Data normalization

data normalization
detection of highly-variable genes

05 Scaling and regression

cell cycle scoring
data scaling and regression
identification of data dimensionality

06 Cell clustering

cell clustering
projection on t-SNE and UMAP
identification of cluster markers

07 Cell annotation

cell annotation
plots for marker genes

popsicleR workflow. The package is composed of seven main functions to perform exploration of quality-control metrics, filtering of low-quality cells, identification of cell doublets, data normalization, removal of technical and biological biases, cell clustering, and cell annotation.

popsicleR: a R Package for pre-processing and quality control analysis of single cell RNA-seq data

Francesco Grandi[1*], Jimmy Caroli[2*], Oriana Romano[1], Matteo Marchionni[1], Mattia Forcato[1§] and Silvio Bicciato[1§]

1 - Department of Life Sciences, University of Modena and Reggio Emilia – Modena, Italy

2 - Department of Drug Design and Pharmacology, University of Copenhagen – Copenhagen, Denmark

[*] **Equal contribution**

[§] **Correspondence to Mattia Forcato and Silvio Bicciato**: mattia.forcato@unimore.it (Mattia Forcato), silvio.bicciato@unimore.it (Silvio Bicciato)

Abstract

The advent of single-cell sequencing is providing unprecedented opportunities to disentangle tissue complexity and investigate cell identities and functions. However, the analysis of single cell data is a challenging, multi-step process that requires both advanced computational skills and biological sensibility. When dealing with single cell RNA-seq (scRNA-seq) data, the presence of technical artifacts, noise, and biological biases imposes to first identify, and eventually remove, unreliable signals from low-quality cells and unwanted sources of variation that might affect the efficacy of subsequent downstream modules. Pre-processing and quality control (QC) of scRNA-seq data is a laborious process consisting in the manual combination of different computational strategies to quantify QC-metrics and define optimal sets of pre-processing parameters.

Here we present *popsicleR*, a R package to interactively guide skilled and unskilled command line-users in the pre-processing and QC analysis of scRNA-seq data. The package integrates, into several main wrapper functions, methods derived from widely used pipelines for the estimation of quality-control metrics, filtering of low-quality cells, data normalization, removal of technical and biological biases, and for cell clustering and annotation. *popsicleR* starts from either the output files of the Cell Ranger pipeline from 10X Genomics or from a feature-barcode matrix of raw counts generated from any scRNA-seq technology. Open-source code, installation instructions, and a case study tutorial are freely available at https://github.com/bicciatolab/popsicleR.

Highlights

- Pre-processing and quality control (QC) of single cell RNA-seq data is a laborious process, entailing both computational skills and biological sensibility
- An effective pre-processing requires the manual combination of different computational strategies to quantify QC-metrics and define optimal sets of pre-processing parameters

- Currently, no generally applicable set of methods and optimized threshold values has been unanimously defined to pre-process single cell RNA-seq data
- *popsicleR* is a R package for the interactive, standardized, and reproducible pre-processing and quality control of single cell RNA-seq data
- *popsicleR* main functions integrates, into a user-friendly environment, pre-processing methods derived from widely used computational workflows

Introduction

The constantly growing applications of single-cell technologies are flooding biologists with a new wave of high-dimensional and complex data that demands advanced computational expertise for their effective interpretation [1, 2]. As compared to bulk experiments, the analysis of single cell data is further complicated by the inherent heterogeneity of the datasets that widely vary in terms of experimental protocols and biological context [3]. This issue, combined with the limited interactivity between software tools and the lack of unanimously defined standard procedures, severely hampers the applicability of general processing pipelines, thus limiting the efficacy of analysis tasks and the reproducibility of results. Despite considerable efforts [4-8], there is still a pressing need to develop computational tools that interactively guide the user in assembling custom, dataset-specific workflows starting from different modules of widely used pipelines [9, 10]. In the analysis of single cell RNA-seq (scRNA-seq), a critical step is represented by pre-processing and quality control (QC) of input data, as, in this phase, signals from low-quality cells, technical artifacts, and unwanted sources of variation that might compromise the subsequent downstream analyses are identified and removed. Pre-processing and quality control (QC) of scRNA-seq data is a laborious process that requires the combination of different computational strategies and the manual definition of appropriate sets of parameters derived from the quantification of specific QC-metrics. Yet, no consensus has been reached, and, considering the heterogeneity of scRNA-seq data, is unlike to be reached soon, on the definition of a gold standard pipeline for scRNA-seq data pre-

preprocessing and quality control, thus increasing the difficulty to choose an optimal analysis workflow [11].

Here we introduce *popsicleR*, a R package that combines methods implemented in widely used pipelines to interactively perform all major pre-processing and QC steps of scRNA-seq data analysis. The package is composed of seven main functions capable of performing exploration of quality-control metrics, filtering of low-quality cells, data normalization, removal of technical and biological biases, and some explorative analyses as detection of differentially expressed genes, cell clustering, and cell annotation. *popsicleR* accepts as input either files from the Cell Ranger pipeline of 10X Genomics or a feature-barcode matrix of raw counts generated from any microfluidic-, microwell plates-, or droplet-based scRNA-seq technology. During each step of the analysis, *popsicleR* interactively guides the user with colored text messages and saves in dedicated folders a variety of plots to investigate several QC-metrics and assess the impact of filtering and regression parameters on the identification and classification of cell populations. To test and illustrate the functionalities of our package, we applied *popsicleR* to the analysis of the transcriptional profiles of 14,972 cells from a sample of human early-stage lung adenocarcinoma [12].

Methods

Workflow overview

popsicleR comprises seven major wrapping functions implemented in R, i.e., i) *PrePlots* to import data and quantify QC metrics; ii) *FilterPlots* to filter low-quality cells; iii) *CalculateDoublets* to detect the presence of doublet cells; iv) *Normalize* to normalize data and identify highly variable genes; v) *ApplyRegression* to calculate cell cycle phase scores and subtract unwanted sources of variation (regression); vi) *CalculateCluster* to cluster cells and identify differentially expressed features (cluster markers); and vii) *MakeAnnotation* to assign cell type identities. *popsicleR* exploits the structure of the Seurat object for scRNA-seq data, i.e., a class containing both expression data (raw and normalized count matrices) and analysis results and metadata (projections, clusters, annotations) for a single-cell sample or a dataset [6]. The major functions build on several commonly used packages for the analysis of scRNA-seq data, as *Seurat* [6], *Scater* [13], *scCancer* [14], *SingleR* [15], and *scMCA* [16]. Doublet detection is implemented using the R version of Scrublet [17] (provided at https://rdrr.io/github/ChengxiangQiu/rscrublet/) and the scDblFinder R package [18, 19]. Violin, feature, dimensionality reduction, and dot plots are based on the graphical routines of *Seurat* and *ggplot2* R packages.

Installation

popsicleR requires R version 4.0.0 or higher and depends on the following R packages: *ape*, *celldex*, *clustree*, *corrplot*, *crayon*, *dplyr*, *future*, *ggExtra*, *ggplot2*, *ggplotify*, *gtools*, *grid*, *gridExtra*, *limma*, *magrittr*, *patchwork*, *pheatmap*, *neldermead*, *RANN*, *RColorBrewer*, *reticulate*, *R.utils*, *scDblFinder*, *scMCA*, *session*, *shinythemes*, *umap*, *Seurat*, and *SingleR.* After the installation of all required dependencies, the package can be installed from Github via the *install_github* function of the *devtools* R package. In addition, an Anaconda yaml file and a detailed installation guide are available in the repository, granting the possibility to generate a complete working environment using the Anaconda platform. Details on the installation of *popsicleR* package and dependencies are described in the "Installation in R" section of https://github.com/bicciatolab/popsicleR.

Results

scRNA-seq example data

We applied *popsicleR* to the analysis of scRNA-seq data of tumor and immune cell populations from a human early-stage lung adenocarcinoma obtained using 10X Genomics [12]. The analyzed sample (sample N1) is composed of 14,972 single cells that were sequenced on the Illumina Hiseq X platform. Raw data have been downloaded from https://ngdc.cncb.ac.cn/biosample/browse/SAMC106013. A guided tutorial of the entire analysis is available at https://github.com/bicciatolab/popsicleR.

Input data preparation

Prior to the analysis with *popsicleR*, raw data has been processed with Cell Ranger software (version 3.1.0) to align reads to the hg38 version of the human genome and generate feature-barcode matrices. Cell Ranger data, comprising the gzipped TSV files of the feature-barcode matrix and of feature and barcode sequences, can be found at https://drive.google.com/drive/folders/15rpnxWik0XQIv5LIrQVYfB_9XufPVnOP?usp=sharing . To organize input data and result files, it is convenient to create a directory named as the sample name and containing a subfolder with the input data (either in the form of Cell Ranger gzipped TSV files or of a tab-delimited matrix; refer to the guided tutorial at https://github.com/bicciatolab/popsicleR for details).

Quality controls

The analysis workflow starts with the *PrePlots* function that first reads the input data and creates a unique molecular identifier (UMI) count matrix in the form of a Seurat object, i.e., a class container for expression data, analysis results, and cell metadata (**Figure 1**). The values in the count matrix represent the number of molecules for each feature (i.e., gene; rows) that are detected in each cell (columns). During the generation of the count matrix, a soft filter is applied to remove low quality cells. By default, *PrePlots* removes genes that are expressed in less than 0.1% of cells and cells that express less than 200 genes. The structure of the Seurat object is maintained throughout the whole pipeline and updated by each *popsicleR* function. Once created the Seurat object, *PrePlots* explores several QC metrics to help detecting those columns of the count matrix that contain gene expression signals from low-quality cells. Low-quality cells result from the capture of cells in stressed conditions or dead, of cell debris from broken cells, of more than one cell (cell doublets) or of no cell and represent artifacts that might deceive downstream analyses [20]. The number of unique genes detected in each cell, the total number of molecules detected within a cell (UMIs), the percentage of reads that map to the mitochondrial genome, and to ribosomal and dissociation genes are commonly used

QC metrics to detect low-quality cells [14, 20]. Low number of detected genes or low expression signals are often indicative of low-quality cells or empty captures, aberrantly high numbers of genes or of UMIs are frequently due to cell doublets, elevate fractions of mitochondrial genes are commonly indicative of loss of cytoplasmatic mRNA due to cell lysis, and high levels of dissociation-associated genes are often expressed by tumor cells stressed by the dissociation process. To guide the choice of appropriate filtering thresholds for low-quality cells, *PrePlots* generates violin, density, and scatter plots of the number of unique genes and total UMIs detected within each cell and of the percentage of reads mapped to the mitochondrial, ribosomal, and dissociation genes (**Figure 2A**). Caution must be taken when samples are composed by an heterogenous mixture of cell populations, since some populations are intrinsically characterized by ranges of unique genes, total UMIs, and mitochondrial fractions that may appear as outliers in the general distribution. As an effect, the application of thresholds on QC metrics could remove not only low-quality but also biologically relevant cells, e.g., T cells, that are intrinsically characterized by low RNA content and number of expressed genes. To help the user avoiding this pitfall, *PrePlots* also returns the cell abundance and the distributions of several features (i.e., number of unique genes, total UMIs detected, mitochondrial fraction) within specific cell populations identified by a list of user-selected marker genes (e.g., SFTPC for alveolar cells and CD3D for T cells; **Figure 2A-C**). To set more accurate thresholds on the minimum number of genes and on the minimum number of molecules detected within a cell, density and scatter plots are also provided as graphs zoomed-in on low value ranges of these features.

Cell filtering

Custom filters to discard low-quality cells can be applied using the *FilterPlots* function. Since scRNA-seq datasets vary widely in number and type of profiled cells, of investigated biological samples, and for technical aspects, no set of fixed filters is generally applicable to filter out low quality cells. Thus, *FilterPlots* allows the user to design combinations of filtering criteria based on the QC metrics quantified by *PrePlots*. Thresholds can be set on the minimum and maximum number of genes detected in each cell, on the minimum and maximum number of molecules detected within a cell, and on the maximum percentage of mitochondrial, ribosomal, and dissociation genes. Once thresholds are applied, *FilterPlots* returns summary graphs with data distributions and correlations that highlight cells passing the filtering procedure (**Figure 2D-E**). These plots can be further investigated to adaptively optimize the filtering criteria.

Doublet detection

As an additional QC and filtering step, the function *CalculateDoublets* allows detecting and, eventually, removing cell doublets, i.e., groups of cells captured and associated to the same unique barcode that typically present gene expression profiles of mixed cell populations. Doublet detection can be obtained through Scrublet [17] or scDblFinder [18, 19], i.e., two methods that use k-nearest neighbor (KNN) classifiers to distinguish doublets from singlets. Scrublet infers the likelihood for each cell to be a doublet (doublet score) by randomly sampling and combining observed transcriptomes. The distribution of the simulated doublet scores is typically bimodal, separating expected doublet cells with similar expression profiles (lower values of the doublet score) from doublet cells with separate transcriptomes (higher values of the doublet score) that are likely to affect downstream analyses. Doublets are identified as those cells with a doublet score higher than the automatically detected minimum score separating the two modes of the simulated doublet histogram. In *CalculateDoublets*, Scrublet has been coded in R starting from the script provided at https://rdrr.io/github/ChengxiangQiu/rscrublet/. The doublet detection approach implemented in scDblFinder creates artificial doublets combining cells taken from different clusters obtained by a fast clustering. Dimensionality reduction is then performed on the union of real cells and artificial doublets and a nearest neighbor network is constructed to estimate the proportion of artificial doublets among each cell nearest neighbors. This feature, together with library size and co-expression score of each cell, is used to train multiple gradient-boosted trees to distinguish artificial doublets from real cells and to assign doublet scores [19]. For both Scrublet and scDblFinder*, CalculateDoublets* returns the visualization of detected cell doublets and of doublet scores on a UMAP projection (**Figure 2F**), adds doublet classifications and scores in the metadata of the Seurat object, and eventually removes cells classified as doublets. The UMAP projections allow to visualize how cells with high doublet score are organized, since potential doublets tend to co-cluster or to form branches or bridges between separate groups of cells. If the Scrublet method is selected, *CalculateDoublets* also displays the distributions of observed and simulated doublets and the minimum score to call cell doublets. In this case, the function might be run twice, the first time to inspect the score distributions and evaluate the automatically detected minimum score and the second to select the appropriate score threshold (either automatically detected or user-defined) to mark doublet cells.

Data normalization and identification of highly variable genes

After removing low-quality cells from the dataset, data are normalized using the *Normalize* function. By default, *Normalize* applies the global-scaling normalization method implemented in the *NormalizeData* function of the Seurat package. The function normalizes the feature expression measurements for each cell by the total expression, multiplies this by a scale factor (10,000 by default), and log-transforms the result. The *Normalize* function also calculates a subset of features that exhibit high cell-to-cell variation in the dataset (i.e., that are highly expressed in some cells, and lowly expressed in others) and that are used to later perform dimensionality reduction with Principal Component Analysis (PCA). This step is based on the *FindVariableFeatures* function of the Seurat package. *Normalize* returns histograms of signal distributions before and after normalization and scatter plots highlighting the highly variable genes.

Cell cycle scoring, scaling, and regression

Before dimensionality reduction, scaling and, eventually, regression must be performed. Scaling is required for PCA and makes the data to have zero-mean and unit standard deviation (standardization). Regression is used to mitigate the effect of unwanted sources of variation, as experimental and biological biases. One source of biological variation is represented by cell cycle that is quantified, before scaling or regression, using the *ApplyRegression* function. This quantification exploits the method implemented in the Seurat *CellCycleScoring* function to assign each cell with cell cycle scores and a predicted cell cycle phase*,* based on the expression of G2/M and S phase markers. In *popsicleR*, scaling and regression are executed by the *ApplyRegression* function that uses the functionalities of Seurat *ScaleData*. *ApplyRegression* simply scales the data, if no variable to regress is indicated, or performs regression on several variables specified by the user. By default, *ApplyRegression* is first run without regressing any variable (data scaling only) to observe, in reduced embeddings as PCA, t-SNE (t-distributed stochastic neighbor embedding), or UMAP (Uniform Manifold Approximation and Projection), if any experimental or biological covariate influences the data. In case regression is needed, *ApplyRegression* can be executed a second time inputting covariates like the number of genes, the number of UMIs, the percentage of mitochondrial genes, or the cell cycle as regression variables. The function returns post-regression PCA, t-SNE, and UMAP plots to investigate the efficacy of regression in removing the effect of confounding variables. Once completed the scaling or regression process, *ApplyRegression* performs PCA on the scaled (or regressed) data and returns several graphs (i.e., heatmap, jackstraw, and elbow plots from the Seurat functions *DimHeatmap*, *JackStraw*, and *ElbowPlot*) to guide the user in selecting the optimal number of principal components for the subsequent dimensionality reduction and clustering steps.

Cell clustering

Cell clustering is used to group cells based on their transcriptomes assuming that, in a low-dimensional embedding, cells with the same identity or in the same state map close together. In *popsicleR* cell clustering is executed by the *CalculateCluster* function. *CalculateCluster* exploits the graph-based clustering approach implemented in Seurat *FindNeighbors* and *FindClusters* functions using as input the previously identified number of principal components and a resolution parameter that sets the granularity of the clustering. The resolution parameter can be either a single value or a set of values to explore cell grouping at different resolutions. Cells clusters are visualized in dimensionality reduced planes as UMAP or t-SNE embeddings, while a phylogenetic tree is used to show correlation between the identified clusters (**Figure 3A**). Beside cluster membership, in UMAP and t-SNE embeddings cells are colored according to several features that might aid in the assessment of cluster quality or in the identification of clusters of problematic cells that escaped the previous filters. For instance, an extremely high expression of MALAT1, a nuclear retained transcript, coupled with a small number of unique genes, can be a proxy for cells that lost most of their cytoplasmic mRNA. If multiple resolutions have been selected, the *clustree* package is used to produce a clustering tree that shows how cells are assigned to the clusters at the various clustering resolutions (**Figure 3B**). In case a single resolution is indicated, *CalculateCluster* uses the *FindAllMarkers* function of Seurat to determine genes differentially expressed in the comparison between cells in each cluster and those in all other clusters (cluster markers). The function returns a table, in the form of a tab-delimited file, containing, for each cluster, the list of all its putative markers and their associated statistics. Expression values of the top 10 markers of each cluster are used to generate a heatmap and the expression of the top two markers is visualized through t-SNE, UMAP, and violin plots. Finally, *CalculateCluster* returns dot plots showing, for a list of user-defined genes and in all clusters identified at the various resolutions, the fraction of cells expressing a gene (dot size) and its average expression in those cells (dot color; **Figure 3C**).

Cell annotation

In the last step of *popsicleR*, the *MakeAnnotation* function is used to automatically assign cell type identities matching the transcriptome of each single cell to gene expression signatures of known cell types [21]. *MakeAnnotation* annotates cells using *SingleR* and *scMCA* built-in tissue-specific reference data. Specifically, *SingleR* is used to annotate data from human and mouse samples with the references of the *celldex* package [15]. For human samples, *celldex* includes the Human Primary Cell Atlas (HPCA) and the Blueprint Encode (BpEn) references generated from microarrays of human primary cells and RNA-seq profiles of human stroma and immune cells, respectively. For mouse samples, references are derived from a collection of mouse bulk RNA-seq datasets (Mouse RNA-seq) and from the microarray profiles of pure mouse immune cells provided by the Immunological Genome Project (ImmGen). *scMCA* is used to annotate data from mouse samples with a reference constructed from the single-cell Mouse Cell Atlas (scMCA) and covering all mouse cell types [16]. Single cell annotations for the different references are displayed color-coding cells in t-SNE and UMAP embeddings

(**Figure 3D**). Since annotation can be performed either at the cluster or at the single cell level, cells can be colored based on clusters or on cell type identity. In the annotation at single cell level, it can be difficult to discern the different populations in the low-dimensional embeddings, as cells are overlapping, and the color scales might be inefficient in separating the various cell types. To overcome this limitation, *MakeAnnotation* produces a plot for each identified population with only cells of the given type highlighted in color on top of all other cells colored in grey. Finally, *MakeAnnotation* returns dot plots for a set of user-defined genes in the predicted cell populations and a pie-chart plot showing the clusters composition according to the various annotations (**Figure 3E**). The *MakeAnnotation* function returns a Seurat object that includes all the calculated cell features in the metadata slot and that can be used for downstream analyses.

Usage statistics

Memory usage and computation time depends on the number of cells composing each sample. As an example, *popsicleR* required about 1.5 hours to entirely process the example data (using Scrublet to detect doublets) on a personal computer with 64 GB of RAM and R version 4.1.2.

Discussion and conclusions

Pre-processing and quality control of scRNA-seq data are complex processes based on the application of different computational methods and the manual definition of appropriate sets of parameters derived from the investigation of several QC-metrics. Accordingly, several bioinformatics workflows and strategies have been developed to explore the quality of the data and to identify and remove low-quality cells, technical artifacts, and unwanted sources of variation that might compromise the subsequent identification of cell populations that are biologically meaningful [6, 8, 13, 14, 22, 23]. Despite major efforts, no consensus has been reached yet on the definition of a reference pipeline of general use for scRNA-seq data pre-processing and quality control. Consequently, the choice of appropriate thresholds to filter out low quality cells, while preserving high quality ones and biologically relevant data, is still a laborious task that involves multiple iterative runs of quantification, visualization, and application of different QC-metrics. All these issues urge the deployment of flexible computational tools that interactively guide the user along the various pre-processing and explorative steps of the scRNA-seq data analysis, thus improving their efficacy and the reproducibility of results.

In this paper we presented *popsicleR*, a R package capable of performing all major scRNA-seq data pre-processing and QC steps through the integration of methods from different workflows. The package implements functions to explore QC metrics, filter low-quality cells, normalize data, remove technical and biological biases, and perform cell clustering and annotation. *popsicleR* can be used either starting from the output files of specific pipelines (as, for instance, Cell Ranger of 10X Genomics) or from feature-barcode matrices generated from any scRNA-seq technology. During the analysis, *popsicleR* returns graphs and colored text messages to interactively guide even inexperienced command-line users in investigating several QC-metrics and in evaluating the impact of filtering parameters on the identification and classification of cell populations. These latter functionalities constitute a distinctive characteristic of our package over other tools developed in R. At each step of the analysis, *popsicleR* allows the user to adapt and store parameters in a sequential manner directly from the R console, without the need to edit external files or search for commands in precompiled menus. Moreover, *popsicleR* returns a variety of graphs that, without requiring any experience in programming graphical routines, support the user in defining the most appropriate set of parameters and in monitoring its impact on the analysis results.

We evaluated the usability of our package in comparison with other R-based workflows comprising pre-processing modules, as *scCancer* [14], *iCellR* [24], and *SingleCellTK* [25]. All tools are based on wrapper functions that allow performing data pre-processing, quality control, analysis, and visualization directly from the R console. In addition to the console analysis, *SingleCellTK* offers the possibility to analyze the data through a R Shiny graphical user interface (GUI). Although the various pipelines can pre-process datasets composed of multiple samples, they are all conceptually designed to be applied on a single sample base. *SingleCellTK* quantifies QC metrics and detects doublets separately for each sample and then performs all other pre-processing steps (as filtering or dimensionality reduction) on the aggregated dataset. *iCellR* can be run in aggregate mode to merge multiple datasets (data frames/matrices) into one file prior its analysis as a single meta-sample. Similarly, *scCancer* and *popsicleR* can perform pre-processing and quality control of a multi-sample dataset obtained through the *cellranger aggr* pipeline that can be used to aggregate cells from different replicates, tissues, or individuals into a single feature-barcode matrix. It is worth noting that pre-processing and QC analysis on a single sample is intended for small size experiments where a rigorous assessment of the quality of each single sample is considered critical for further downstream analyses (as integration with reference datasets) and might result practically untenable with datasets comprising large number of samples. Nonetheless, when the number of samples is limited, it is preferable to assess data quality on single sample base and perform multi-sample integration as a downstream analysis (through *SeuratMNN*, *Harmony*, and *Liger* algorithms), since each sample might be affected by sample-specific biases, that require dedicated pre-processing and filtering steps, and some pre-processing procedures (as doublet detection) has to be performed separately for each sample (see https://github.com/swolock/scrublet and https://github.com/plger/scDblFinder for details).

Since the pre-processing and QC approaches of the various tools are rooted in major packages for the analysis of scRNA-seq data (e.g., *Seurat* [6], *Harmony* [5], *Liger* [7], and *scater* [13]), their differences are mostly limited to functional aspects, i.e., the strategy adopted to assemble the wrapper functions, their functionalities, and the type of graphical outputs, that marginally influence the overall result of the analysis. For instance, even though *scCancer* implements, as compared to *popsicleR*, different approaches for the identification of cell doublets and cell-type annotation (based on the *scds* [26] and *gelnet* [27] packages, respectively), when used on the same scRNA-seq sample, the two pipelines returned similar results both in terms of cells passing the filtering steps (12,167 and 11,608 for *popsicleR* and *scCancer*, respectively), cell clusters (18 and 21 for *popsicleR* and *scCancer*, respectively at a resolution of 0.8), and cell types composing the lung tumor tissue (e.g., as macrophages, NK cells, T cells and B cells, and epithelial cells). Nevertheless, *popsicleR* and *scCancer* significantly differ in the design of their wrapper functions. The workflow of *scCancer* consists of three modules, i.e., *scStatistics* for pre-processing, *scAnnotation* for basic analyses, and *scCombination* for the integration of data from multiple samples. As compared to *popsicleR*, this minimal structure simplifies the overall analysis pipeline but, at the same time, limits the possibility of the user to sequentially adapt threshold values and parameters while performing each step of the analysis. As an example, the parameters for the *scAnnotation* module, which performs cell filtering, functional analyses, and visualizations, are inferred, and automatically set, by *scStatistics*, i.e., the function that quantifies thresholds based on QC metrics. Thus, to modify the QC thresholds and optimize the strength of the filters, the user is required to manually edit the output file of *scStatistics* and run the entire workflow several times. The various tools also present differences in the available functionalities as exemplified by the various options to detect doublets, ranging from the multiple algorithms proposed by *SingleCellTK* to the absence of any doublet detection module in *iCellR*; the possibility to perform cell clustering at multiple resolutions, implemented only in *popsicleR*; or the application of multiple cell annotation methods, available only in *popsicleR* and *iCellR*. As to the type of graphical outputs, all workflows return scatter and violin plots of the major QC metrics (as the number of unique genes and total UMIs and the percentage of mitochondrial, and ribosomal genes), but only *scCancer* and *popsicleR* plot their distributions within specific cell populations identified by a list of user-selected marker genes. This type of graphs is extremely helpful in guiding the selection of appropriate thresholds to remove only low-quality cell while preserving biologically relevant cells, as T cells, intrinsically characterized by low levels of some QC metrics. Conversely, only *iCellR* and *SingleCellTK* result interactive plots created using the graphical routines of *Plotly*, a R package to generate interactive web graphics. Finally, only *popsicleR* returns dot plots showing, for a list of user-defined genes, the fraction of cells expressing a gene and its average expression level across all cells within clusters identified at a given resolutions. The dot plot visualization represents a valuable tool to select the most appropriate clustering resolution, to identify the presence of clusters composed by mixed cell types, to refine cell annotation, and to enhance the exploration of cell heterogeneity.

All in all, we believe that, as single cell assays progressively become indispensable approaches to investigate the fundamental mechanisms regulating cell identity and tissue complexity, automated and user-friendly workflows must be developed to facilitate the routine analysis of single cell genomic data, standardize the computational approaches, and improve the reproducibility of results.

Availability

popsicleR is written in R language and is released under a GPL License. The package can be downloaded from GitHub (https://github.com/bicciatolab/PoPsicleR) along with an installation guide and an application tutorial. *popsicleR* is continuously maintained by the developers, providing continuous support to all external users. Users can report issues and comments directly at the GitHub repository.

CRediT authorship contribution statement

Francesco Grandi: Writing - original draft, Conceptualization, Methodology, Software.

Jimmy Caroli: Writing - original draft, Conceptualization, Methodology, Software.

Oriana Romano: Conceptualization, Methodology.

Matteo Marchionni: Methodology, Software.

Mattia Forcato: Conceptualization, Methodology, Supervision, Funding acquisition, Writing - review & editing.

Silvio Bicciato: Conceptualization, Methodology Supervision, Funding acquisition, Writing - review & editing.

Funding

This work was supported by funds from Fondazione AIRC under 5 per Mille 2019 program (ID. 22759) to S.B. and from the PRIN 2017 Project 2017HWTP2K of the Italian Ministry of Education, University and Research and the FAR 2019 (E54I19002000001) and GR-2016-02362451 of the Italian Ministry of Health to M.F.. F.G. is a recipient of a Doctoral Fellowship Progetti di formazione alla ricerca (Bando 2018) from Regione Emilia Romagna. O.R. has been supported by Fondazione Umberto Veronesi (Post-Doctoral Fellowship 2020). We thank Martina Dori and Andrea Grilli for their support in coding the graphical routines of *popsicleR*.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Keywords:

single cell RNA-sequencing

data analysis

bioinformatics

R language

software tools

Abbreviations used:

QC, Quality Controls

scRNA-seq, single cell RNA-sequencing

HVGs, highly variable genes

t-SNE, t-distributed stochastic neighbor embedding

UMAP, Uniform Manifold Approximation and Projection

HPCA, Human Primary Cell Atlas data

BpEn, Blueprint Encode data

scMCA, single-cell Mouse Cell Atlas

References

1. Stegle O, Teichmann SA, Marioni JC. (2015). Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet.* **16**, 133-45.

2. Angerer P, Simon L, Tritschler S, Wolf FA, Fischer D, Theis FJ. (2017). Single cells make big data: New challenges and opportunities in transcriptomics. *Current Opinion in Systems Biology.* **4**, 85-91.

3. Kharchenko PV. (2021). The triumphs and limitations of computational methods for scRNA-seq. *Nat Methods.* **18**, 723-32.

4. Amezquita RA, Lun ATL, Becht E, Carey VJ, Carpp LN, Geistlinger L, et al. (2020). Orchestrating single-cell analysis with Bioconductor. *Nat Methods.* **17**, 137-45.

5. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, et al. (2019). Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods.* **16**, 1289-96.

6. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, 3rd, et al. (2019). Comprehensive Integration of Single-Cell Data. *Cell.* **177**, 1888-902 e21.

7. Welch JD, Kozareva V, Ferreira A, Vanderburg C, Martin C, Macosko EZ. (2019). Single-Cell Multi-omic Integration Compares and Contrasts Features of Brain Cell Identity. *Cell.* **177**, 1873-87 e17.

8. Moreno P, Huang N, Manning JR, Mohammed S, Solovyev A, Polanski K, et al. (2021). User-friendly, scalable tools and workflows for single-cell RNA-seq analysis. *Nat Methods.* **18**, 327-8.

9. Hwang B, Lee JH, Bang D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med.* **50**, 1-14.

10. Andrews TS, Kiselev VY, McCarthy D, Hemberg M. (2021). Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nat Protoc.* **16**, 1-9.

11. Dal Molin A, Di Camillo B. (2019). How to design a single-cell RNA-sequencing experiment: pitfalls, challenges and perspectives. *Brief Bioinform.* **20**, 1384-94.

12. He D, Wang D, Lu P, Yang N, Xue Z, Zhu X, et al. (2021). Single-cell RNA sequencing reveals heterogeneous tumor and immune cell populations in early-stage lung adenocarcinomas harboring EGFR mutations. *Oncogene.* **40**, 355-68.

13. McCarthy DJ, Campbell KR, Lun AT, Wills QF. (2017). Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics.* **33**, 1179-86.

14. Guo W, Wang D, Wang S, Shan Y, Liu C, Gu J. (2021). scCancer: a package for automated processing of single-cell RNA-seq data in cancer. *Brief Bioinform.* **22**.

15. Aran D, Looney AP, Liu L, Wu E, Fong V, Hsu A, et al. (2019). Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat Immunol.* **20**, 163-72.

16. Sun H, Zhou Y, Fei L, Chen H, Guo G. (2019). scMCA: A Tool to Define Mouse Cell Types Based on Single-Cell Digital Expression. *Methods Mol Biol.* **1935**, 91-6.

17. Wolock SL, Lopez R, Klein AM. (2019). Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data. *Cell Syst.* **8**, 281-91 e9.

18. Germain PL, Sonrel A, Robinson MD. (2020). pipeComp, a general framework for the evaluation of computational pipelines, reveals performant single cell RNA-seq preprocessing tools. *Genome Biol.* **21**, 227.

19. Germain P, Lun A, Macnair W, Robinson M. (2021). Doublet identification in single-cell sequencing data using scDblFinder [version 1; peer review: 1 approved, 1 approved with reservations]. *F1000Research.* **10**.

20. Ilicic T, Kim JK, Kolodziejczyk AA, Bagger FO, McCarthy DJ, Marioni JC, et al. (2016). Classification of low quality cells from single-cell RNA-seq data. *Genome Biol.* **17**, 29.

21. Clarke ZA, Andrews TS, Atif J, Pouyabahar D, Innes BT, MacParland SA, et al. (2021). Tutorial: guidelines for annotating single-cell transcriptomic maps using automated and manual methods. *Nat Protoc.* **16**, 2749-64.

22. Garcia-Jimeno L, Fustero-Torre C, Jimenez-Santos MJ, Gomez-Lopez G, Di Domenico T, Al-Shahrour F. (2021). bollito: a flexible pipeline for comprehensive single-cell RNA-seq analyses. *Bioinformatics.*

23. Parekh S, Ziegenhain C, Vieth B, Enard W, Hellmann I. (2018). zUMIs - A fast and flexible pipeline to process RNA sequencing data with UMIs. *Gigascience.* **7**.

24. Tang KH, Li S, Khodadadi-Jamayran A, Jen J, Han H, Guidry K, et al. (2022). Combined Inhibition of SHP2 and CXCR1/2 Promotes Antitumor T-cell Response in NSCLC. *Cancer Discov.* **12**, 47-61.

25. Wang Y, Sarfraz I, Hong R, Koga Y, Alabdullatif S, Jenkins D, et al. (2021). singleCellTK: Comprehensive and Interactive Analysis of Single Cell RNA-Seq Data. *R package version 240, https://wwwcamplabnet/sctk/.*

26. Bais AS, Kostka D. (2020). scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics.* **36**, 1150-8.

27. Sokolov A, Paull EO, Stuart JM. (2016). One-Class Detection of Cell States in Tumor Subtypes. *Pac Symp Biocomput.* **21**, 405-16.

Figure legends

Figure 1. Workflow. The workflow of *popsicleR*.

Figure 2. Pre-processing and filtering. **A.** Distributions of the number of unique genes (nFeature_RNA) detected in all cells. **B.** Distributions of the number of unique genes (nFeature_RNA) detected in cells expressing SFTPC, a marker of lung alveolar cells. **C.** Distributions of the number of unique genes (nFeature_RNA) detected in cells expressing CD3D, a marker of T lymphocytes. In **A.**, **B**., and **C.** the X axis is zoomed in the range 0 to 2000. **D.** Scatter plot showing the correlation between the total number of unique genes (nFeature_RNA) and the total number of UMIs (nCount_RNA), across all single cells. Dashed grey lines indicate the thresholds set on minimum and maximum number of genes detected in each cell and on the maximum number of molecules detected within a cell. **E.** Scatter plot showing the correlation between the total number of unique genes (nFeature_RNA) and the percentage of mitochondrial genes (percent_mt), across all single cells. Dashed grey lines indicate the thresholds set on minimum and maximum number of genes detected in each cell and on the maximum percentage of mitochondrial genes. In **D.** and **E.**, cells that passed the combinations of filters are highlighted in red (**D**.) and blue (**E.**), while cell filtered out are colored in gray. **F.** Doublet score of each single cell projected in the UMAP plane.

Figure 3. Clustering and annotation. **A.** UMAP visualization of cells colored by cluster membership at resolution 0.8. Numbers indicate the cluster identity. **B.** Clustering tree displaying how cells move among clusters as resolution increases from 0.4 to 0.8. The tree is shown only for selected clusters. Cluster identity is displayed inside the circle; circle color indicates the clustering resolution; circle size is proportional to the number of cells in the cluster. Edge color is related to the number of cells whereas its transparency shows how many cells of the incoming node are assigned to the destination node. **C.** Dot plot reporting, in the clusters identified at resolution 0.8, the expression of genes commonly associated to general immune populations. Dot size is proportional to the fraction of cells with non-zero expression of the marker gene. Dots are color-coded based on the average scaled normalized expression.

D. UMAP visualization of cells colored by population label assigned using the Blueprint ENCODE reference atlas. **E.** Pie-chart plot displaying, for each cluster at resolution 0.8, the proportion of cells annotated with a specific label using the Blueprint ENCODE reference atlas.

Figure 1

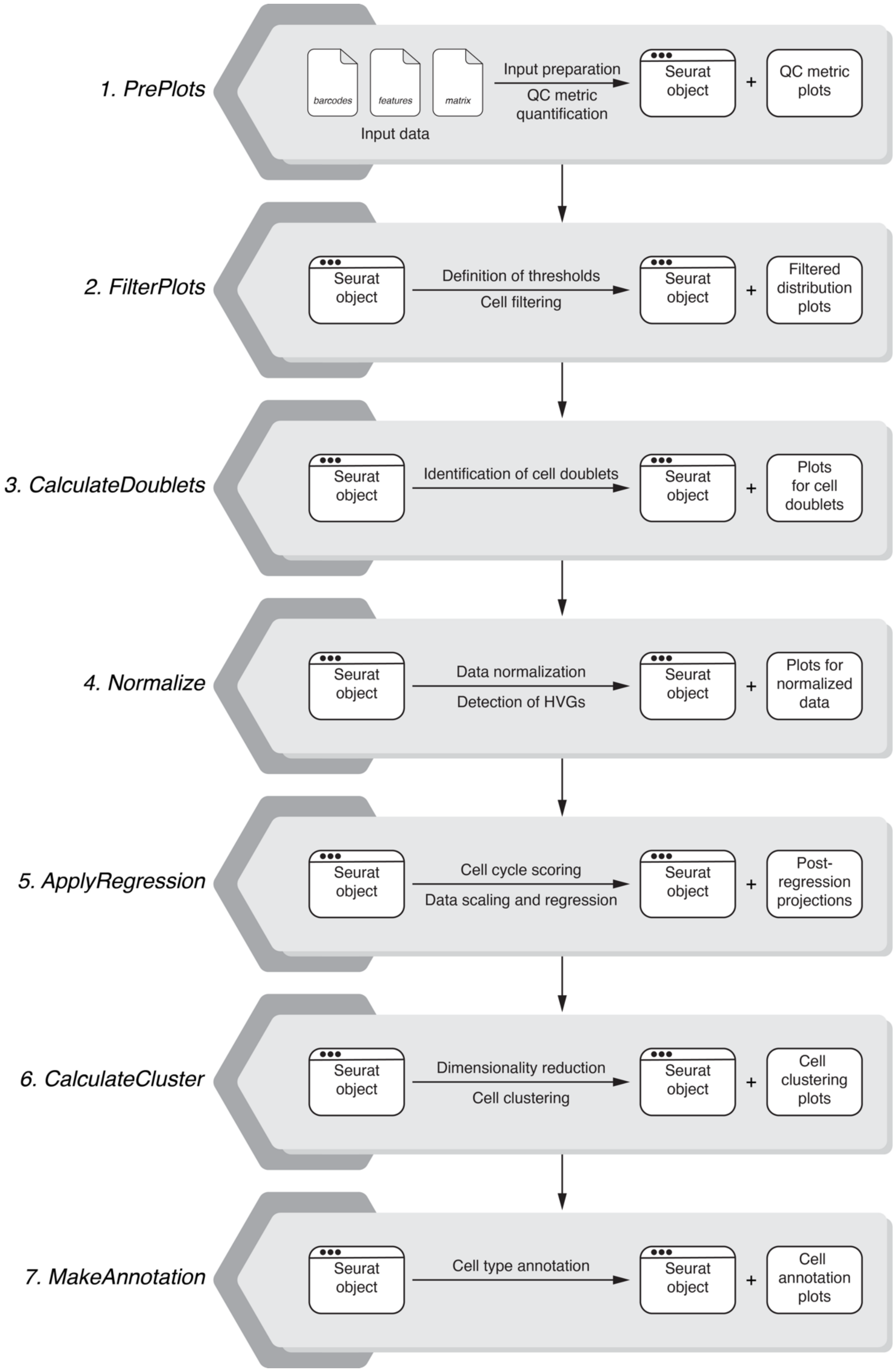

1. PrePlots
barcodes
features
matrix
Input data
Input preparation
QC metric quantification
Seurat object
+
QC metric plots
2. FilterPlots
Seurat object
Definition of thresholds
Cell filtering
Seurat object
+
Filtered distribution plots
3. CalculateDoublets
Seurat object
Identification of cell doublets
Seurat object
+
Plots for cell doublets
4. Normalize
Seurat object
Data normalization
Detection of HVGs
Seurat object
+
Plots for normalized data
5. ApplyRegression
Seurat object
Cell cycle scoring
Data scaling and regression
Seurat object
+
Post-regression projections
6. CalculateCluster
Seurat object
Dimensionality reduction
Cell clustering
Seurat object
+
Cell clustering plots
7. MakeAnnotation
Seurat object
Cell type annotation
Seurat object
+
Cell annotation plots

Figure 2

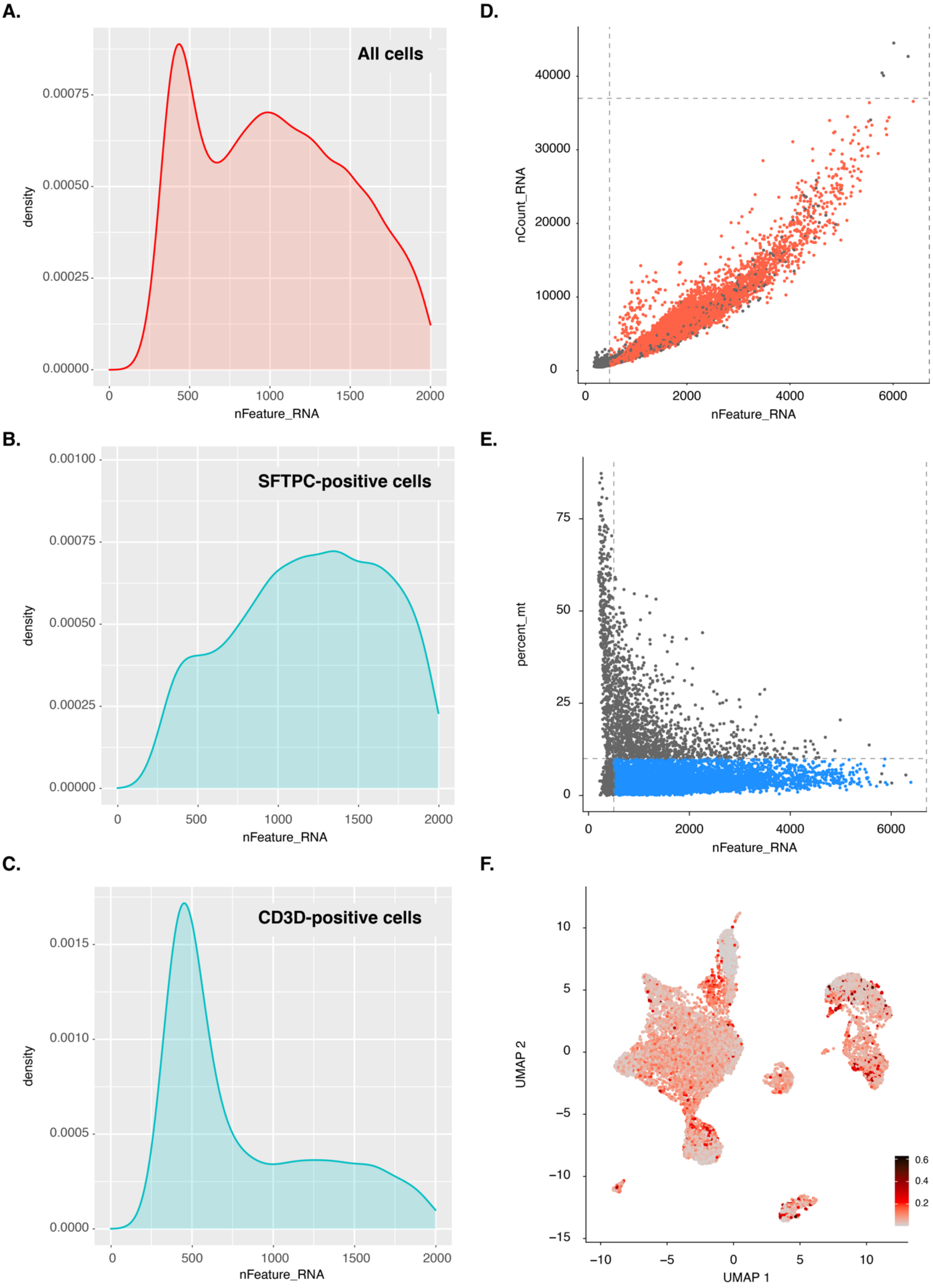
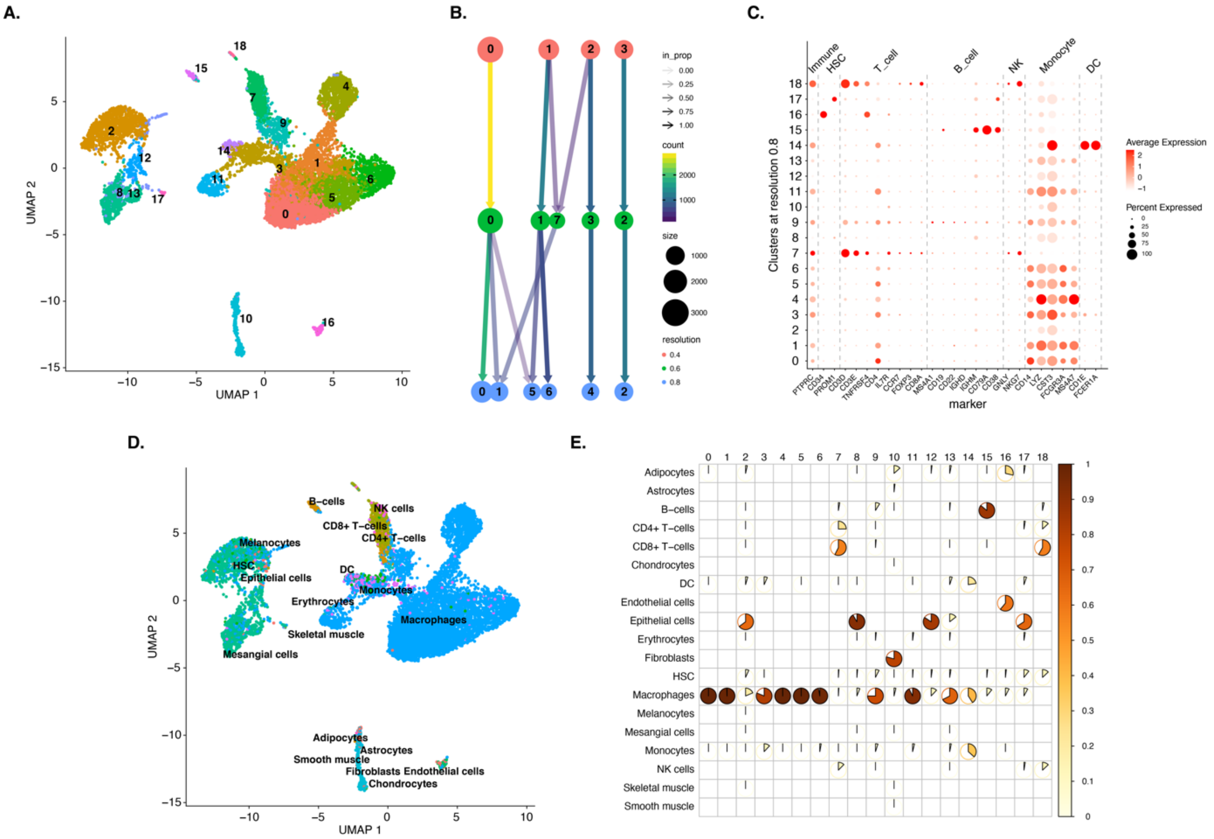

A.
All cells
density
0.00075
0.00050
0.00025
0.00000
0
500
1000
1500
2000
nFeature_RNA
B.
SFTPC-positive cells
density
0.00100
0.00075
0.00050
0.00025
0.00000
0
500
1000
1500
2000
nFeature_RNA
C.
CD3D-positive cells
density
0.0015
0.0010
0.0005
0.0000
0
500
1000
1500
2000
nFeature_RNA
D.
nCount_RNA
40000
30000
20000
10000
0
0
2000
4000
6000
nFeature_RNA
E.
percent_mt
75
50
25
0
0
2000
4000
6000
nFeature_RNA
F.
UMAP 2
10
5
0
−5
−10
−15
−10
−5
0
5
10
UMAP 1
0.6
0.4
0.2

Figure 3


A.
UMAP 1
UMAP 2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
B.
in_prop
0.00
0.25
0.50
0.75
1.00
count
2000
1000
size
1000
2000
3000
resolution
0.4
0.6
0.8
C.
Immune
HSC
T_cell
B_cell
NK
Monocyte
DC
Clusters at resolution 0.8
marker
PTPRC
CD34
PROM1
CD3D
CD3E
TNFRSF4
CD4
IL7R
CCR7
FOXP3
CD8A
MS4A1
CD19
CD22
IGHD
IGHM
CD79A
CD38
GNLY
NKG7
CD14
LYZ
CST3
FCGR3A
MS4A7
CD1E
FCER1A
Average Expression
Percent Expressed
D.
B-cells
NK cells
CD8+ T-cells
CD4+ T-cells
Melanocytes
HSC
Epithelial cells
DC
Monocytes
Erythrocytes
Macrophages
Skeletal muscle
Mesangial cells
Adipocytes
Astrocytes
Smooth muscle
Fibroblasts
Endothelial cells
Chondrocytes
E.
Adipocytes
Astrocytes
B-cells
CD4+ T-cells
CD8+ T-cells
Chondrocytes
DC
Endothelial cells
Epithelial cells
Erythrocytes
Fibroblasts
HSC
Macrophages
Melanocytes
Mesangial cells
Monocytes
NK cells
Skeletal muscle
Smooth muscle