bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Title: Comprehensive generation, visualization, and reporting of quality control metrics for single-cell RNA sequencing data

Authors: Rui Hong[1,2]*, Yusuke Koga[1,2]*, Shruthi Bandyadka[1,5], Anastasia Leshchyk[1,2], Yichen Wang[2], Vidya Akavoor[2,3], Xinyun Cao[3], Irzam Sarfraz[2], Zhe Wang[1,2], Salam Alabdullatif[2], Frederick Jansen[3], Masanao Yajima[4], W. Evan Johnson[1,2], Joshua D. Campbell[1,2].

1. Bioinformatics Program, Boston University, Boston, MA, USA.
2. Section of Computational Biomedicine, Boston University School of Medicine, Boston, MA, USA
3. Software & Application Innovation Lab, Rafik B. Hariri Institute for Computing and Computational Science and Engineering, Boston, MA, USA
4. Department of Mathematics and Statistics, Boston University, Boston, MA, USA
5. Department of Biology, Boston University, Boston, MA, USA
* Contributed equally to this work

Corresponding author:
Joshua D. Campbell
camp@bu.edu

Abstract

Single-cell RNA sequencing (scRNA-seq) can be used to gain insights into cellular heterogeneity within complex tissues. However, a variety of technical artifacts can be present in scRNA-seq data and need to be assessed before downstream analyses can be performed. While several algorithms and tools have been developed to perform individual quality control (QC) tasks, they are scattered in different packages across several programming environments. Comprehensive pipelines to streamline the process of generating and visualizing QC metrics are lacking. To address this need, we built the SCTK-QC pipeline within the *singleCellTK* R package (https://github.com/compbiomed/singleCellTK). Features in this pipeline include the ability to import data from 11 different preprocessing tools or file formats, perform empty droplet detection with 2 different algorithms, generate standard quality control metrics such as number of UMIs per cell or the percentage of mitochondrial counts, predict doublets using 6 different algorithms, and estimate ambient RNA. QC data can be exported to R and/or Python objects used in popular down-stream workflows. Results are visualized in an easy-to-read HTML report. This pipeline can also be used by non-computational users with an interactive graphical user interface developed with R/Shiny. Overall, the SCTK-QC pipeline will streamline and standardize QC analysis for scRNA-seq data across a variety of different single-cell transcriptomic platforms and preprocessing tools.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Introduction

Single-cell RNA-sequencing (scRNA-seq) has been instrumental in providing detailed insights into cellular heterogeneity related to tissue development and disease pathogenesis[1]. With the advent of microfluidic devices, the transcriptome for thousands of individual cells can be measured in a single run[2]. These devices work by partitioning cells into droplets along with beads containing oligonucleotide primers with unique barcodes. Within each droplet, reverse transcription is initially used to create barcoded cDNA and then additional amplification steps are used to create final sequencing libraries depending on the protocol[3]. Other approaches such as SMART-seq2 and CEL-seq2 can be used to profile cells that have been sorted into 96- or 384-well plates[4,5]. Many of these protocols use unique molecular indices (UMIs) to barcode each individual mRNA molecule and correct for biases in amplification[6].

Despite the advances in scRNA-seq protocols, poor-quality cells can still be present in high-quality runs. Technical artifacts related to the cell dissociation process, cell encapsulation, library preparation, or sequencing can affect various aspects of data quality. Low quality cells need to be excluded and technical artifacts need to be systematically assessed in each sample before downstream analyses can be performed. We briefly describe five types of QC analyses and metrics that are commonly utilized in scRNA-seq data analysis: 1) Cells in which barcoding or amplification reactions were not successful will have lower numbers of UMIs and genes detected. Lower numbers of detected UMIs and genes can hinder downstream analyses such as clustering because the genes that are able to distinguish cell populations may not be adequately measured. Often, these cells are excluded by setting a minimum threshold on the number of detected UMI and/or genes. 2) Another aspect unique to droplet-based microfluidic devices is that the majority of the droplets will not contain an actual cell[7]. Despite the absence of a cell, these “empty droplets” may contain low levels of background ambient RNA that was present in the cell solution[8]. An algorithm is needed to determine which droplets likely contained a real cell versus those that just contain ambient RNA[8]. Only droplets predicted to contain an actual cell are used in downstream analyses. 3) Doublets and multiplets occur when two or more cells are partitioned into a single droplet or well and will result in an artificial hybrid expression profile of each individual cell[2]. Several algorithms have been developed to identify potential doublets by combining expression profiles of randomly selected cells and then scoring each cell against the *in silico* doublets[9,10]. These tools can

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

be used to flag potentially problematic cell clusters that are actually combinations of two different cell types. 4) Ambient RNA in the cell suspension can also be present in droplets containing a cell as well as empty droplets. These ambient transcripts will be counted along with a cell's native RNA and result in contamination of highly-expressed genes from other cell types. Tools such as DecontX can be used to estimate contamination levels and deconvolute each cell into counts derived from native RNA and counts from contaminating ambient RNA[11]. 5) Perturbations during sample preparation can lead to biological artifacts. For example, cells that become stressed during tissue dissociation may express abnormally large proportions of mitochondrial genes in their transcriptome[12]. These cells may appear as a unique cluster in the scRNA-seq data even though they were not present in the original tissue sample. If not taken into account, these factors can confound downstream analyses or produce erroneous findings. Therefore, performing comprehensive QC is a crucial step in scRNA-seq data analysis to ensure valid results.

While a large number of QC algorithms and software tools have been produced to address the specific challenges inherent in scRNA-seq data, these tools are implemented in different packages across various programming environments. In order to generate a comprehensive set of QC metrics, users need to separately download, install, and run each tool for each sample and independently assess the results[13]. Currently, there is a lack of standardized workflows that can streamline the process of generating QC metrics from different tools. In order to address these limitations, we have developed the SCTK-QC pipeline within the *singleCellTK* R package. This pipeline can import single-cell RNA-seq data from a variety of preprocessing tools, run a multitude of different tools to generate comprehensive sets of QC metrics, visualize the results within detailed HTML reports, and export the results in an organized manner in various formats.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Results

Overview of SCTK-QC Pipeline

The SCTK-QC pipeline is accessible through the *singleCellTK* package in R/Bioconductor. This pipeline assumes that the raw sequencing reads have been aligned, a correction for UMI and cell barcodes has been applied, and a count matrix containing genes and cells has been created by an upstream preprocessing tool such as CellRanger[7] or STARsolo[14]. For data generated with microfluidic devices, the first major step after UMI counting is to detect cell barcodes that represent droplets containing a true cell and exclude empty droplets that only contain ambient RNA[8]. We use the terms "Droplet" matrix to denote a count matrix that still contains empty droplets, "Cell" matrix to denote a count matrix of cells where empty droplets have been excluded but no other filtering has been performed, and "FilteredCell" matrix to indicate a count matrix where poor quality cells have also been excluded. The Droplet and Cell matrices have also been called "raw" and "filtered" matrices, respectively, by tools such as CellRanger. However, using the term "filtered" can be ambiguous as other forms of cell filtering can be applied beyond empty droplets (e.g. excluding poor-quality cells based on low number of UMIs). Additionally, even after excluding empty droplets and poor-quality cells, the matrix will still contain unnormalized counts, which is also commonly referred to as the "raw" count matrix. To eliminate ambiguity of these terms, we adopt the nomenclature of "Droplet", "Cell", and "FilteredCell" to describe the level of filtering on the dimensions of the count matrix while we prefer the terms "Raw", "Normalized", and "Scaled" to denote the level of processing for the counts within the matrix.

The major steps in the SCTK-QC pipeline include: 1) importing of the Droplet matrix, 2) detection and exclusion of empty droplets to create the Cell matrix, 3) calculation of a comprehensive set of QC metrics on the Cell matrix, 4) visualization of results in HTML format, and 5) exporting the data to formats used in downstream analysis workflows (**Figure 1**). Note that several preprocessing tools automatically exclude empty droplets and create a Cell matrix. The SCTK-QC pipeline also has the ability to import a Cell matrix and start with the calculation of QC metrics in step 3 or import both the Droplet and Cell matrices and perform QC on each matrix independently. The single-cell data is stored within the pipeline as a *SingleCellExperiment* object[13]. Cell-level metrics generated by QC tools are stored in the *colData* slot alongside other imported cell-level annotations and corrected raw counts matrices created by any QC tool are stored in the *assays* slot. For

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

reproducibility, the parameters and seeds used to run the functions within the pipeline will be also stored in the *metadata* slot. Overall, the pipeline supports importing data from 11 different preprocessing tools or file formats, empty droplet detection with 2 algorithms, generation of standard QC metrics, doublet detection using 6 algorithms, and estimation of ambient RNA (**Table 1**). SCTK-QC integrates numerous tools across different programming environments such as R and Python. To streamline installation and minimize challenges with package dependencies, we have built Docker and Singularity images which are available through DockerHub (campbio/sctk_qc). The specific algorithms and tools used in each step of the pipeline are described in more detail below.

1. Data import

SCTK-QC can automatically import data from a variety of preprocessing tools and file formats. Supported preprocessing tools include CellRanger[7], BUStools[15], STARSolo[14], SEQC[16], Optimus[17], and dropEST[18]. Generally, users will only need to specify the top-level directories for one or more samples and SCTK-QC will import and combine each sample into a single matrix. Alternatively, specific file formats such as Market Exchange Format (MEX) or a file containing comma-separated values (.csv) can be specified along with separate files for feature and cell annotation. By default, SCTK-QC will run QC analysis on both Droplet matrix and Cell matrix if both of them are provided. However, users can also choose to run QC only on the Droplet or only on the Cell matrix. The sample labels for each cell are stored in a variable called “sample” within the *colData* slot of the *SingleCellExperiment* object. Each QC algorithm will be applied to cells from each sample separately.

2. Empty droplet detection

Detection of empty droplets within the Droplet matrix is accomplished using the algorithms *barcodeRanks* and *EmptyDrops* from the *dropletUtils* package[8]. These algorithms are incorporated within the wrapper function *runDropletQC()*, implemented within SCTK-QC. *barcodeRanks* ranks all barcodes within the Droplet matrix based on total UMI counts per barcode. The knee and inflection points are computed from the log-log plot of the rank against the total counts. Under the assumption that cells will have a higher number of total UMI counts than empty droplets, barcodes with total counts under the knee or inflection points are flagged as empty droplets. Rank, total counts, knee and inflection point are all outputted from the algorithm and stored within the *SingleCellExperiment* object. In contrast, *emptyDrops* differentiates between empty droplets containing

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

ambient RNA from true cells by employing a probabilistic model that assumes a "pool" of ambient RNA from the Droplet matrix randomly contaminates each droplet. By comparison with a pool of ambient RNA simulated from the counts data, the algorithm determines droplets containing true cells to be those only containing ambient RNA. Metrics generated from this model include the total UMI counts per barcode, the log-probability, Monte Carlo p-value, and the q-value for a droplet containing a real cell, and a value signaling whether increasing the number of iterations within the algorithm will increase the likelihood of identifying a lower p-value. The *SingleCellExperiment* object containing the Droplet matrix can be automatically filtered on either the *barcodeRanks* or *emptyDrops* output to create a new *SingleCellExperiment* object containing the Cell matrix if this matrix was not originally supplied as input to the SCTK-QC pipeline.

3. Generation of QC metrics

Wrapper functions for each QC algorithm or tool are included in SCTK-QC. Additionally, the wrapper function *runCellQC()* is capable of executing these algorithms all at once within SCTK-QC. *runCellQC()* applies algorithms available from the *scater*[13] package to the Cell matrix to compute standard metrics. This includes the total UMI counts per cell, total number of features detected per cell, and the percentage of library size occupied by the most highly expressed genes in each cell. Users may also supply any gene set of their choice to calculate the aggregate expression of the gene set per cell. As a specific use case, a list of mitochondrial genes may be supplied to *runCellQC()* to compute the mitochondrial gene expression per cell. Mitochondrial gene sets for mouse and human in Gene Symbol, Ensembl and Entrez formats are stored in the package and can be supplied to the SCTK-QC pipeline by setting "-M" parameter. The *runCellQC()* function employs the following algorithms for doublet identification in the Cell matrix: *scrublet*[9], *scDblFinder*[19], *DoubletFinder*[10], and the *cxds*, *bcds* and *cxds_bcds_hybrid* models from *SCDS*[20] package. All of these algorithms output a doublet score. Most of these tools also derive a threshold and make a call as to whether each cell is a doublet or a singlet. Running multiple algorithms allows users to set their own criteria for flagging potential doublets that works best for their dataset. Finally, the *runCellQC()* function runs DecontX[10] to detect ambient RNA contamination for each cell within the Cell matrix. The percentage of estimated contamination is stored within the *colData* and the decontaminated count matrix is stored as an assay in the *SingleCellExperiment* which can be optionally used in downstream analysis. After completion of *runCellQC()*, users can use any combination of these QC metrics to filter the Cell matrix and create a FilteredCell matrix for use in downstream analyses.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

4. Generation of comprehensive QC HTML reports

Rmarkdown documents can be used to create dynamic HTML or PDF reports useful for systematic display and evaluation of data[21]. We include the functions *reportDropletQC(), reportCellQC(), and reportQCTool()* which make use of algorithm-specific Rmarkdown document templates to generate HTML reports with the visualizations of QC metrics from all algorithms (**Figure 2**). *reportDropletQC()* generates a report including a scatterplot annotating all empty droplets flagged by the EmptyDrops algorithm as well as a curve visualizing the knee and inflection points identified by BarcodeRanks. For each set of doublet detection algorithm executed, *reportCellQC()* generates a report which visualizes the doublet score and call through violin plots, density plots, and dimensionality reduction plots. These plots are also created to visualize the contamination percentage of ambient RNAs computed by DecontX if the algorithm has been applied to the data. Additionally, both reports include a summary table detailing the outputted quality control metrics for all algorithms run.

5. Export to common data structures

Different software packages utilize varying data containers to store and retrieve scRNA-seq data[22]. To facilitate downstream analysis in multiple platforms, the SCTK-QC pipeline provides several functions to export the data in one or more data structures or file formats. The *exportSCEtoFlatFile* function writes assays to MEX files and the *colData*, *rowData*, *reducedDims* slots into tab-delimited flat files. The metadata is exported as a list in an RDS file. All exported files can be optionally saved in a gzipped format. The *exportSCEtoAnnData()* function exports the data into a Python annotated data matrix (*AnnData*)[23] object. The function stores assay, rowData, colData and reducedDims slots into X, var, obs and obsm groups of the AnnData object, respectively. The AnnData object can be written into a .h5ad file format and can subsequently be compressed in a “gzip” or “lzf” format. These functions can be run by setting the “-F” or “--outputFormat” parameter in the SCTK-QC pipeline.

R/Shiny user interface

Shiny is a R package developed for building interactive web applications. The *singleCellTK* package incorporates Shiny Graphical User Interface (GUI) for the interactive analysis of single-cell data. Users are able to access the user interface by simply executing the *singleCellTK*() function in the R console. Upon loading the data, QC algorithms to be run are chosen on the “Data QC & Filtering” page by selecting checkboxes in the user interface (**Figure 3**). Upon the completion of the algorithms, QC plots will appear within tabs for each of

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

the algorithms selected. The “Filtering” tab can be used to set criteria for filtering. After QC, users are able to interactively perform other downstream analyses such as batch correction, feature selection, dimensionality reduction, clustering, differential expression, and pathway analysis.

Comparison to other tools

Several other tools that can perform single-cell RNA sequencing data analysis and quality control have been created. While many packages only support input data stored in structured format (*SingleCellExperiment* object, *Seurat*[24,25] object or count matrix stored in csv/txt/mtx file), SCTK-QC also accepts data generated from different preprocessing tools and .h5ad files. Although other packages can perform general QC metrics including number of reads and features detected per cell, SCTK-QC includes comprehensive QC analysis including empty droplet detection, doublet detection and ambient RNA correction (**Table 2**). Furthermore, no other software package currently runs multiple doublet detection methods and allows users to easily compare results. SCTK-QC also visualizes QC metrics in standardized html reports and stores results in several data formats, which facilitates downstream analysis in different analysis workflows. Currently, SCTK-QC does not support RSEM as an input format and it does not support other Python objects such as *pickle* and *joblib* as these are not commonly used.

Application of SCTK-QC pipeline to PBMC datasets

To demonstrate the utility of SCTK-QC, we apply the pipeline to the 10x Genomics 1K healthy donor Peripheral Blood Mononuclear Cell (PBMC) dataset generated with v2 or v3 Chromium chemistries. Each dataset was processed with two different versions of Gencode GTF files (Gencode v27 and Gencode v34). The resulting four count matrices (Gencode v27 PBMC 1K v2, Gencode v27 PBMC 1K v3, Gencode v34 PBMC 1K v2, Gencode v34 PBMC 1K v2) were then processed by the SCTK-QC pipeline. Specifically, the pipeline used the *importCellRangerV2()* and *importCellRangerV3()* function by setting the “-cellRangerDirs” as the path of input data , and the “-dataType” parameter as “filtered” and the *runCellQC()* function was called to generate the QC metrics. All of the QC metrics are summarized for each of the four samples in **Table 3**. The distributions for some of the general QC metrics and the decontX decontamination scores are displayed in violin plots. (**Figure 4**). As expected, the median counts and features detected in the alignments from v3 chemistry PBMC datasets were almost double than those detected from v2 chemistry indicating the higher capture sensitivity of the 10x v3 chemistry. No significant difference was observed in the total read counts (p =

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

0.93; *t*-test) and the number of features detected per cell between the PBMC datasets aligned to different versions of Gencode references (p-value: 0.69; *t*-test). The predicted doublet rate of each dataset varied among different doublet detection methods. With the exception of DoubletFinder, all other methods consistently predicted higher doublets rate for v3 chemistry dataset than those for v2 chemistry dataset. Finally, lower decontX contamination scores suggest improved processing for the samples profiled with the v3 chemistry.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Discussion

The wide applicability of single-cell approaches has led to the development of novel computational tools that allow for clustering and identification of new cell types and trajectory inference of cell populations in development. Despite the improvements of scRNA-seq platforms and protocols, low-quality cells and technical artifacts such as empty droplets, doublets, and ambient RNA still remain present to some degree in most datasets. Thus, rigorous QC measures are needed to evaluate the quality of individual experiments. The SCTK-QC pipeline streamlines and standardizes the generation and visualization of metrics important for assessing data quality. Previous tools like *FastQC* and *RSeQC*[26,27] have enabled extensive quality assessment and visualizing of FASTQ and aligned BAM files. Similarly, the SCTK-QC pipeline enables comprehensive generation and visualization of QC metrics for the initial UMI-corrected count matrix by integrating several algorithms and tools into a common, easy-to-run framework. Importantly, SCTK-QC pipeline provides a framework with standardized data structures for computing and storing QC metrics. This modular architecture of SCTK-QC will allow for easy integration of new tools as they are made available in the future. SCTK-QC is able to generate HTML reports with publication-ready figures and contains a GUI for interactive QC of single-cell data. These features will enable users without in-depth programming backgrounds to run these tools and perform QC on their data. Finally, the SCKT-QC pipeline can export to both R and Python-compatible data structures enabling easy integration with other popular analysis frameworks such as *Seurat*[24,25] and *Scanpy*[23].

Methods

Accessibility

The SCTK-QC pipeline is executable on the R console, Rstudio or on the Unix command-line with an Rscript command. The *singleCellTK* package and quality control pipeline is open sourced through GitHub (https://github.com/compbiomed/singleCellTK) and the Bioconductor repository. Additionally, we have included scripts to set up the Conda or Python virtual environments that meet all cross-platform dependency requirements for convenient portability of the pipeline between operating systems. To encourage reproducibility and make the computing environment independent, the *singleCellTK* package and SCTK-QC pipeline is included in Docker image[28] (https://hub.docker.com/r/campbio/sctk_qc). All dependencies of the *singleCellTK* package are included in the Docker image and the quality control pipeline can be executed with a single

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

docker run. Users can specify parameters used for each QC function by providing a YAML file to the pipeline with argument “-y” or “--yamlFile”. We have created several vignettes and in-depth walkthroughs for installation and analysis workflows which are available on the GitHub repository and at https://www.sctk.science.

Quality control of PBMC datasets

The raw reads in the FASTQ format were downloaded from the 10x Genomics Dataset portal and the human reference genome sequence GRCh38 release versions 27 and 34 in the FASTQ and GTF formats from the GENCODE website. The “mkref” command in *CellRanger* v3.1.0 was used to build separate custom references for Gencode GRCh38 v27 and v34. Droplet and Cell matrices for both PBMC 1k v2 and v3 samples were then obtained by aligning the raw reads to the reference genomes using *CellRanger* v3.1.0 running *bcl2fastq* v2.20. The summary of the dataset size is summarized in Table 3. Quality control on both matrices was conducted with SCTK-QC under default parameters for all QC algorithms.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Figures and Tables

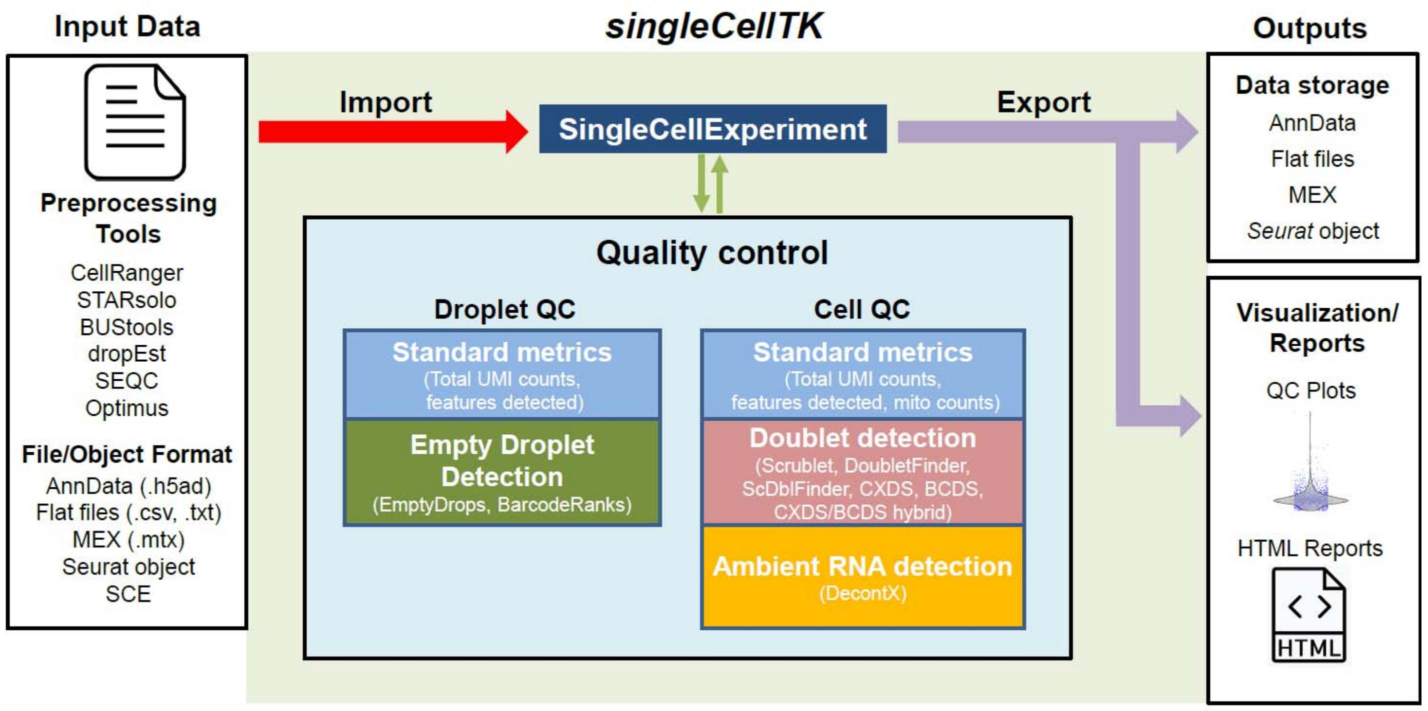
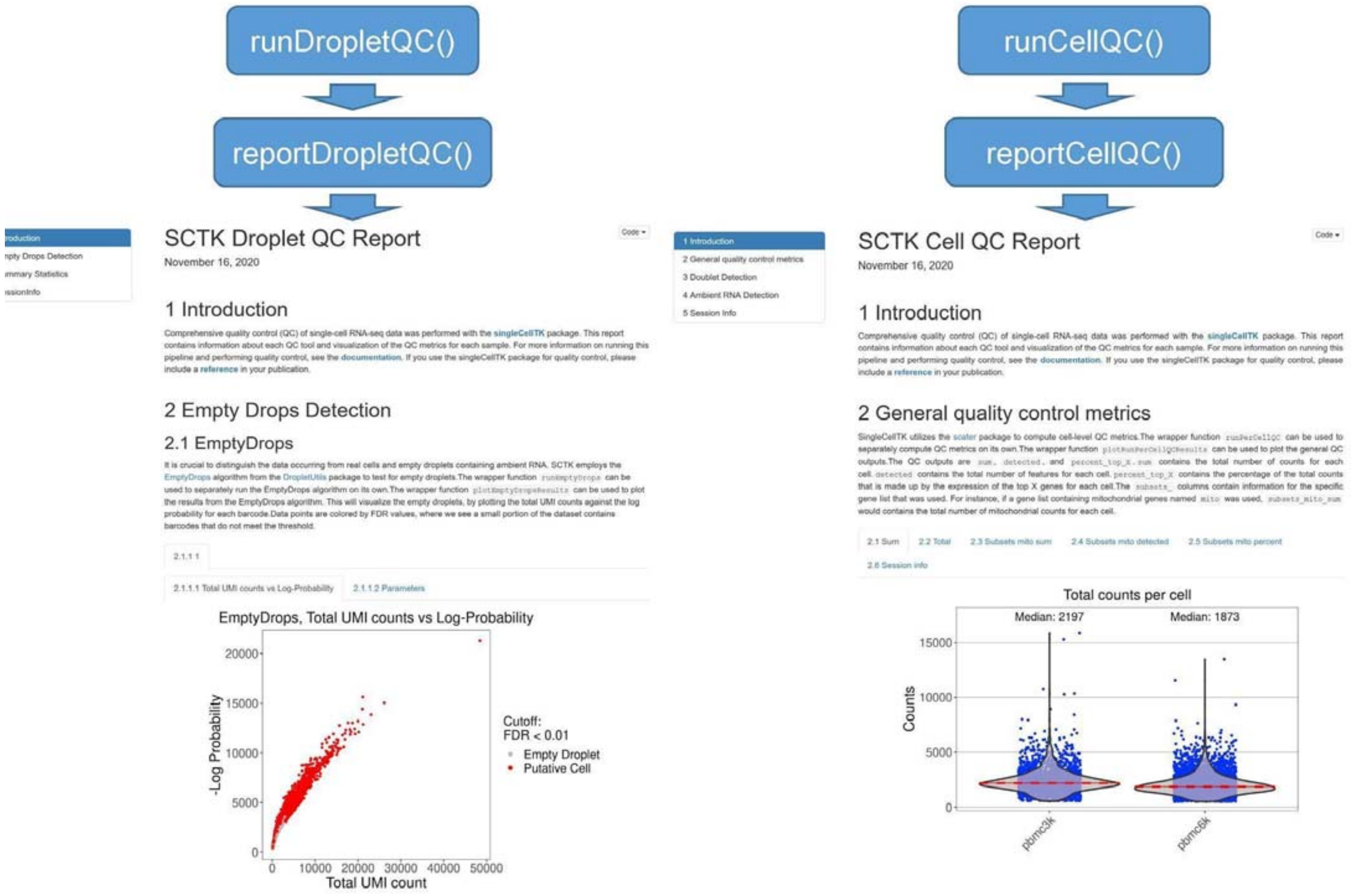


Figure 1. Overview of the SCTK-QC pipeline. The SCTK-QC pipeline is developed in R and can import datasets generated from various preprocessing tools. The pipeline incorporates various software and tools to perform QC for Droplet and/or Cell matrices within each sample. Tools are included for calculation of standard metrics such as the number of UMIs per cell, detection of empty droplets, prediction of doublets, and estimation of contamination from ambient RNA. The pipeline utilizes the *SingleCellExperiment* R object to store assay data and the derived QC metrics. Data visualization and report generation can be subsequently performed on the imported dataset based on user specified parameters. All data can be exported to *Seurat* object, a Python AnnData object, or as MEX and .txt flat files to facilitate analysis in downstream workflows.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



SCTK Droplet QC Report

November 16, 2020

1 Introduction

Comprehensive quality control (QC) of single-cell RNA-seq data was performed with the singleCellTK package. This report contains information about each QC tool and visualization of the QC metrics for each sample. For more information on running this pipeline and performing quality control, see the documentation. If you use the singleCellTK package for quality control, please include a reference in your publication.

2 Empty Drops Detection

2.1 EmptyDrops

It is crucial to distinguish the data occurring from real cells and empty droplets containing ambient RNA. SCTK employs the EmptyDrops algorithm from the DropletUtils package to test for empty droplets.The wrapper function `runEmptyDrops` can be used to separately run the EmptyDrops algorithm on its own.The wrapper function `plotEmptyDropsResults` can be used to plot the results from the EmptyDrops algorithm. This will visualize the empty droplets, by plotting the total UMI counts against the log probability for each barcode.Data points are colored by FDR values, where we see a small portion of the dataset contains barcodes that do not meet the threshold.

SCTK Cell QC Report

November 16, 2020

1 Introduction

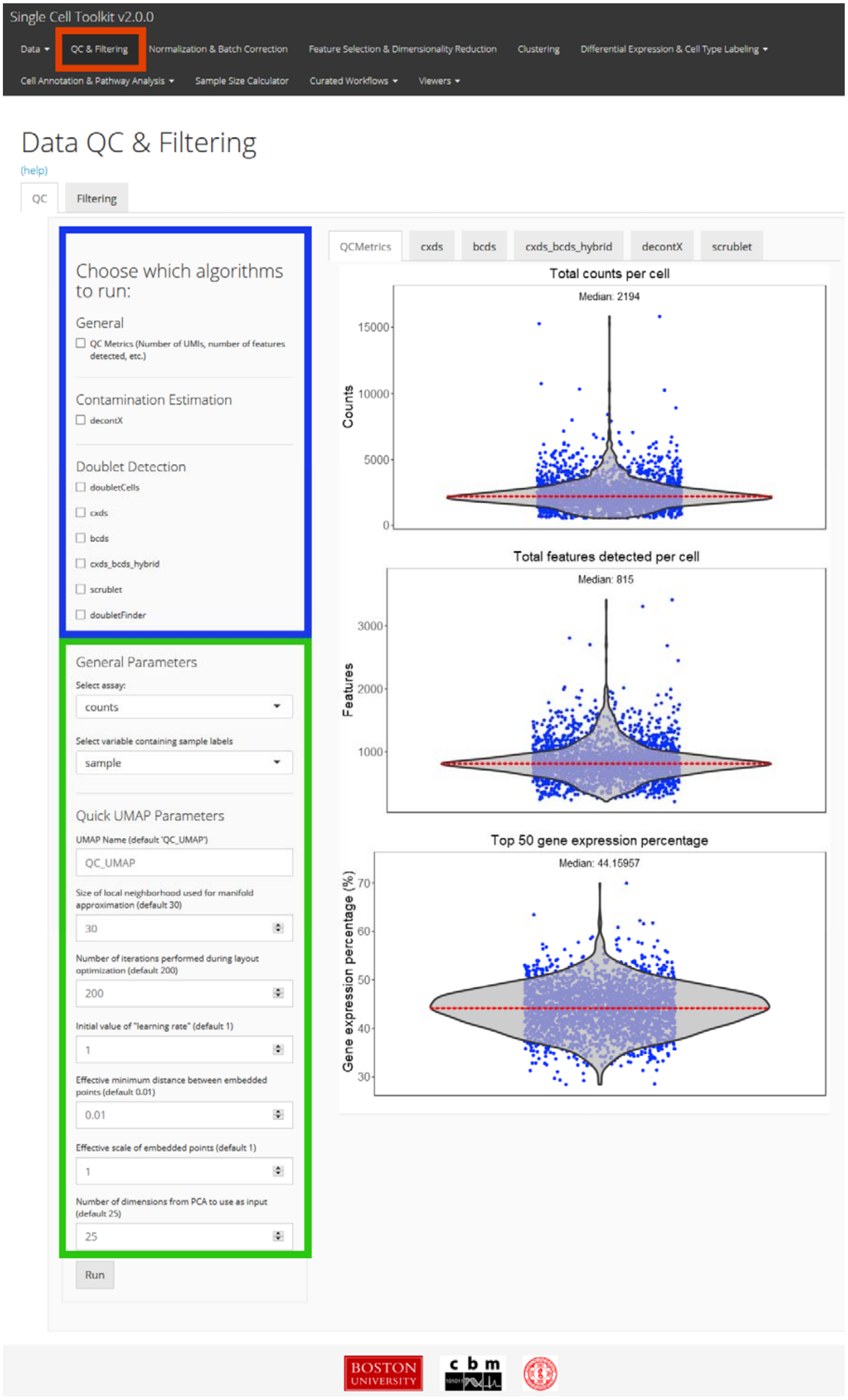
Comprehensive quality control (QC) of single-cell RNA-seq data was performed with the singleCellTK package. This report contains information about each QC tool and visualization of the QC metrics for each sample. For more information on running this pipeline and performing quality control, see the documentation. If you use the singleCellTK package for quality control, please include a reference in your publication.

2 General quality control metrics

SingleCellTK utilizes the scater package to compute cell-level QC metrics.The wrapper function `runPerCellQC` can be used to separately compute QC metrics on its own.The wrapper function `plotRunPerCellQCResults` can be used to plot the general QC outputs.The QC outputs are `sum`, `detected`, and `percent_top_X.sum` contains the total number of counts for each cell. `detected` contains the total number of features for each cell. `percent_top_X` contains the percentage of the total counts that is made up by the expression of the top X genes for each cell.The `subsets_` columns contain information for the specific gene list that was used. For instance, if a gene list containing mitochondrial genes named `mito` was used, `subsets_mito_sum` would contains the total number of mitochondrial counts for each cell.

Figure 2. Generation of HTML reports for visualization and assessment of QC metrics. The functions *reportDropletQC() and reportCellQC()* generate the extensive HTML reports to display data generated by the various QC tools applied by the functions *runDropletQC()* and *runCellQC()*, respectively. The *reportDropletQC()* report contains figures visualizing identified empty droplets. The *reportCellQC()* report contains visualizations of total read counts, total detected, doublet scores, doublet calls, percentages of ambient RNA detected, and cell clusters identified by DecontX. These reports are run automatically by the SCTK-QC pipeline. Examples of *runDropletQC() (on the left)* and *runCellQC() (on the right)* reports are shown.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.


Single Cell Toolkit v2.0.0
Data
QC & Filtering
Normalization & Batch Correction
Feature Selection & Dimensionality Reduction
Clustering
Differential Expression & Cell Type Labeling
Cell Annotation & Pathway Analysis
Sample Size Calculator
Curated Workflows
Viewers
QC
Filtering
QCMetrics
cxds
bcds
cxds_bcds_hybrid
decontX
scrublet
Run
BOSTON UNIVERSITY
c b m

Data QC & Filtering

(help)

Choose which algorithms to run:

General

- ☐ QC Metrics (Number of UMIs, number of features detected, etc.)

Contamination Estimation

- ☐ decontX

Doublet Detection

- ☐ doubletCells
- ☐ cxds
- ☐ bcds
- ☐ cxds_bcds_hybrid
- ☐ scrublet
- ☐ doubletFinder

General Parameters

Select assay: counts

Select variable containing sample labels: sample

Quick UMAP Parameters

UMAP Name (default 'QC_UMAP'): QC_UMAP

Size of local neighborhood used for manifold approximation (default 30): 30

Number of iterations performed during layout optimization (default 200): 200

Initial value of "learning rate" (default 1): 1

Effective minimum distance between embedded points (default 0.01): 0.01

Effective scale of embedded points (default 1): 1

Number of dimensions from PCA to use as input (default 25): 25

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Figure 3. Interactive QC of single cell data using a Graphical User Interface (GUI). An R/Shiny GUI can be used to interactively run QC algorithms in the *singleCellTK* package. A screenshot of the “Data QC & Filtering” tab from the interactive GUI is shown. After importing the data, quality control is performed within the “QC & Filtering” tab (red) of the user interface. QC algorithms are chosen from a list (blue), while specific parameters may be specified as well (green). Plots displaying metrics generated by each QC tool will appear to the right in a tab.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

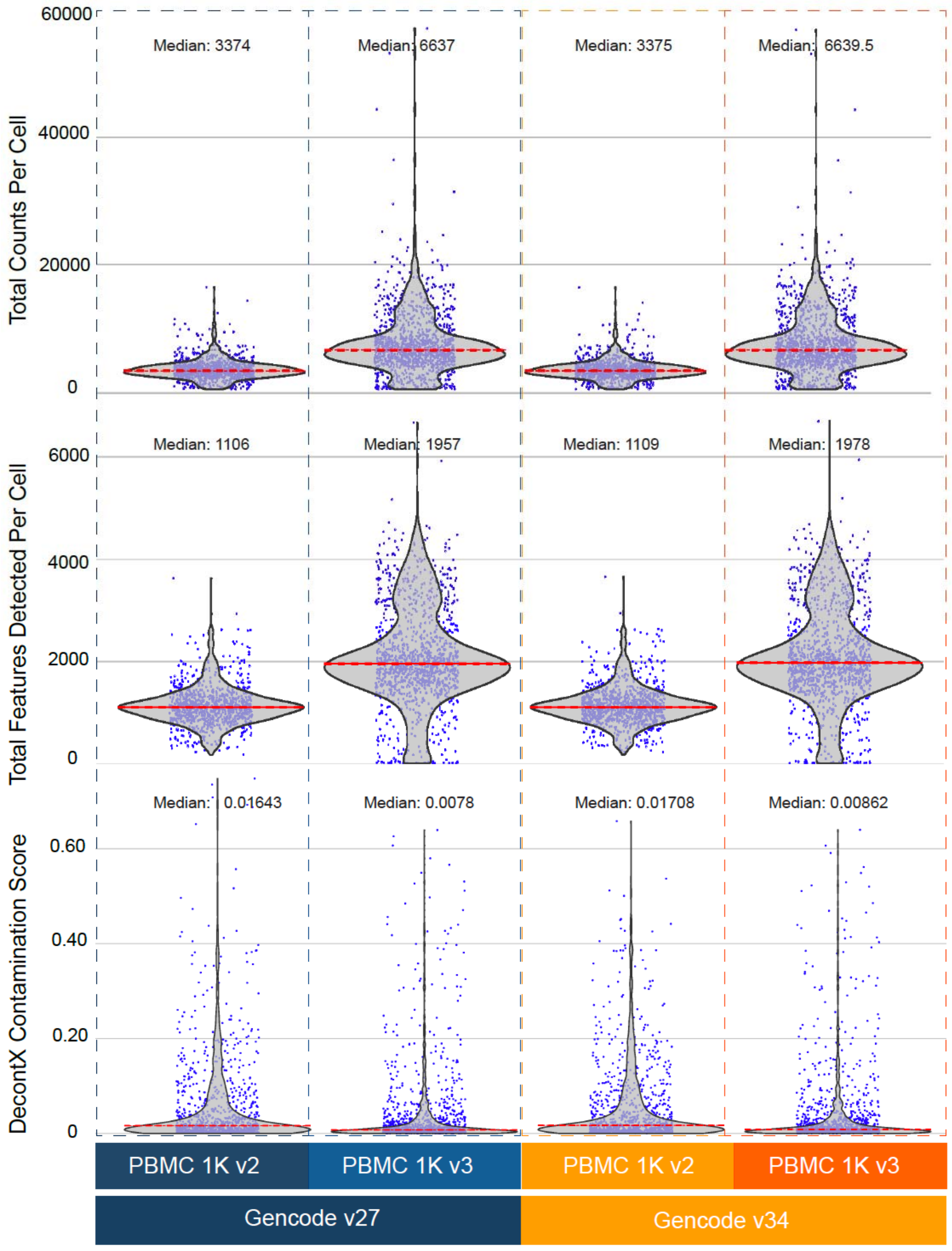


Figure 4. Application of SCKT-QC to PBMC datasets. QC metrics were generated by the SCTK-QC pipeline for 1K healthy donor Peripheral Blood Mononuclear Cell (PBMC) datasets from 10X Genomics. Violin plots generated by the pipeline demonstrate higher capture sensitivity of the 10x v3 Chromium chemistry. Furthermore, lower ambient RNA contamination was observed in the samples run with v3 chemistry compared to samples profiled with the v2 chemistry.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

SCTK QC modules	Methods	Goal	Packages integrated	Function
runDropletQC	runBarcodeRankDrops	Calculate barcode ranks	DropletUtils	barcodeRanks
	runEmptyDrop	Detection of empty droplets	DropletUtils	emptyDrops
runCellQC	runPerCellQC	Compute general quality control metrics	scater	addPerCellQC
	runScrublet	Doublet detection	Scrublet	scrub_doublets*
	runScDblFinder		scran	scDblFinder
	runDoubletFinder		DoubletFinder	doubletFinder_v3
	runCxds		scds	cxds
	runBcds		scds	bcds
	runCxdsBcdsHybrid		scds	cxds_bcds_hybrid
	runDecontX	Detect ambient RNA contamination	celda	decontX

Table 1. Functions available in the singleCellTK package and the SCTK-QC pipeline along with the corresponding wrapper functions. The diverse algorithms and their corresponding SCTK-QC wrapper functions that are used to generate quality control QC metrics in SCTK-QC pipeline. The asterisk denotes Python functions.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

	SCTK	PIVOT	Seurat	ascend	scRNABatch QC	Adobo	SCONE	SCHNAPPs	iS-CellR	Ganatum	ASAP browser
Input format											
10x CellRanger	✓		✓		✓				✓		
SCE Object	✓				✓		✓	✓	✓		
Seurat Object	✓		✓								
AnnData	✓										✓
LOOM											
BUStools	✓										
SEQC	✓										
STARSolo	✓										
Optimus	✓										
DropEst	✓										
TXT, CVN and MTX	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
RSEM							✓				
Ambient droplets detection	✓										
General QC Metrics											
Total counts	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Number of features detected	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Gene set count (e.g mitochondrial)	✓	✓	✓	✓	✓	✓			✓		✓
Doublet detection											
scDblFinder	✓										
Scrublet	✓										
doubletFinder	✓										
cxds	✓										
bcds	✓										
cxds/bcds hybrid	✓										

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Shiny App / interactive	✓	✓					✓	✓	✓	✓	✓
docker	✓	✓							✓		✓
HTML Report	✓			✓	✓						✓
Output format											
RDS	✓			✓	✓			✓			
AnnData	✓										
hdf5	✓										✓
.txt Flatfile	✓							✓			
pickle						✓					
joblib						✓					

Table 2. Comparison of features in the SCTK-QC pipeline with other single-cell analysis toolkits. SCTK-QC pipeline supports various types of input, full scRNA-seq quality control pipeline and supports common data structures for data storage.

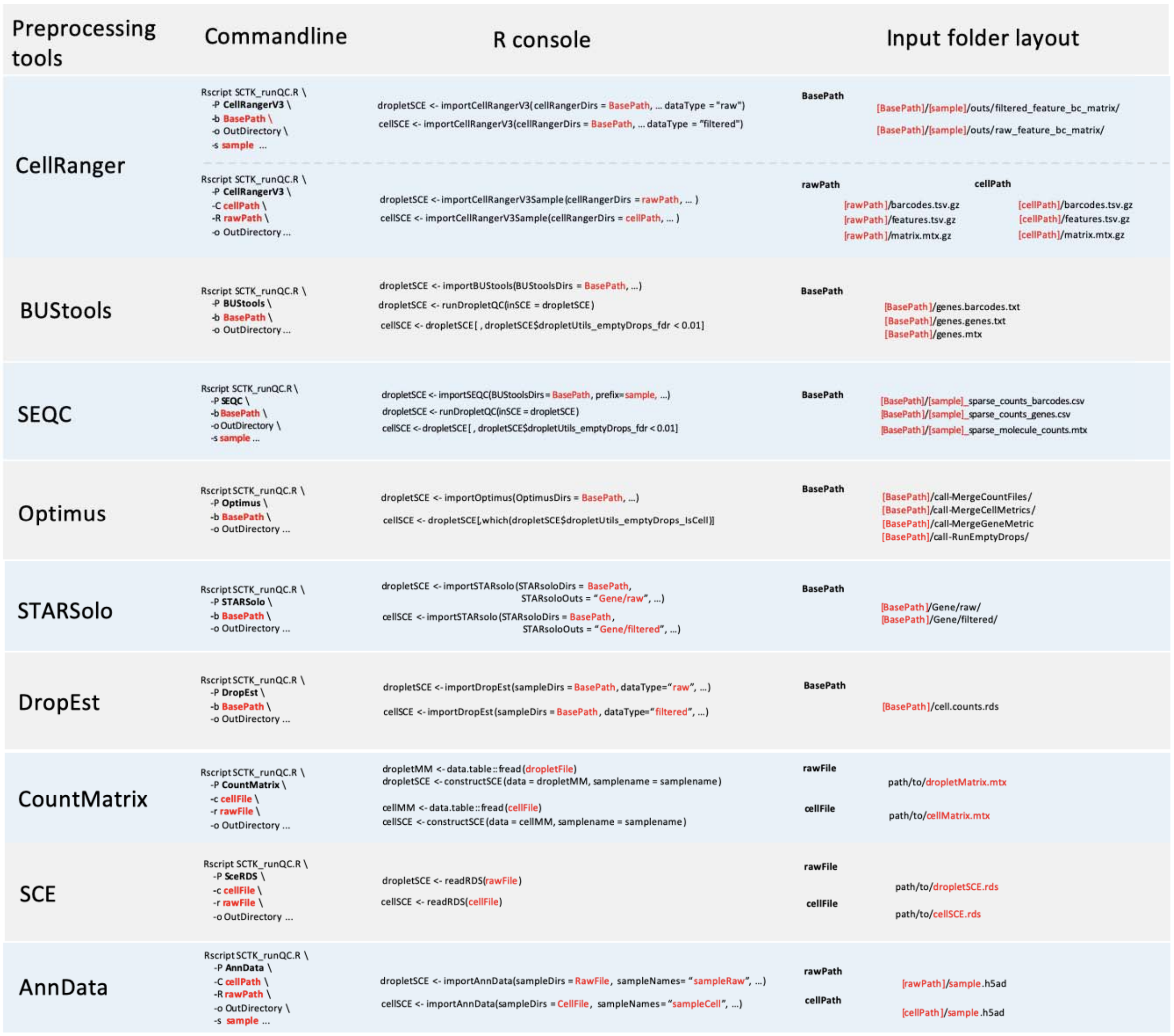
bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

	GENCODE GRCh38 v27		GENCODE GRCh38 v34	
	PBMC1k V2	PBMC1k V3	PBMC1k V2	PBMC 1k V3
Total number of genes detected	58347	60669	58347	60669
Number of droplets, Droplet matrix	737280	6794880	737280	6794880
Number of Cells, Cell matrix	995	1223	996	1222
Mean counts	3560	7580	3550	7580
Median counts	3370	6640	3380	6640
Mean features detected	1130	2090	1140	2100
Median features detected	1110	1960	1110	1980
Scrublet, Number of doublets	12	16	12	18
Scrublet, Percentage of doublets	1.21	1.31	1.2	1.47
ScDblFinder, Number of doublets	13	16	14	20
ScDblFinder, Percentage of doublets	1.31	1.31	1.41	1.64
DoubletFinder, Number of doublets, Resolution 1.5	75	92	75	92
DoubletFinder, Percentage of doublets, Resolution 1.5	7.54	7.52	7.53	7.53
CXDS - Number of doublets	51	195	52	183
CXDS - Percentage of doublets	5.13	15.9	5.22	15
BCDS - Number of doublets	55	104	64	96
BCDS - Percentage of doublets	5.53	8.5	6.43	7.86
SCDS Hybrid - Number of doublets	56	94	71	117
SCDS Hybrid - Percentage of doublets	5.63	7.69	7.13	9.57
DecontX - Mean contamination	0.06	0.04	0.06	0.03
DecontX - Median contamination	0.02	0.01	0.02	0.01

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Table 3. Summary of QC metrics for each PBMC sample. GENCODE PBMC 1k datasets were analyzed with the SCTK-QC pipeline. Two datasets of differing 10x Chemistry were taken from GENCODE v27 and v34, resulting in a total of four datasets. A per-sample summary table is automatically generated by the pipeline.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.



Preprocessing tools	Commandline	R console	Input folder layout
CellRanger	Rscript SCTK_runQC.R \ -P **CellRangerV3** \ -b **BasePath** \ -o OutDirectory \ -s **sample** ...	dropletSCE <- importCellRangerV3(cellRangerDirs = BasePath, ... dataType = "raw") cellSCE <- importCellRangerV3(cellRangerDirs = BasePath, ... dataType = "filtered")	**BasePath** [BasePath]/[sample]/outs/filtered_feature_bc_matrix/ [BasePath]/[sample]/outs/raw_feature_bc_matrix/
	Rscript SCTK_runQC.R \ -P **CellRangerV3** \ -C **cellPath** \ -R **rawPath** \ -o OutDirectory ...	dropletSCE <- importCellRangerV3Sample(cellRangerDirs = rawPath, ...) cellSCE <- importCellRangerV3Sample(cellRangerDirs = cellPath, ...)	**rawPath** [rawPath]/barcodes.tsv.gz [rawPath]/features.tsv.gz [rawPath]/matrix.mtx.gz **cellPath** [cellPath]/barcodes.tsv.gz [cellPath]/features.tsv.gz [cellPath]/matrix.mtx.gz
BUStools	Rscript SCTK_runQC.R \ -P **BUStools** \ -b **BasePath** \ -o OutDirectory ...	dropletSCE <- importBUStools(BUStoolsDirs = BasePath, ...) dropletSCE <- runDropletQC(inSCE = dropletSCE) cellSCE <- dropletSCE[, dropletSCE$dropletUtils_emptyDrops_fdr < 0.01]	**BasePath** [BasePath]/genes.barcodes.txt [BasePath]/genes.genes.txt [BasePath]/genes.mtx
SEQC	Rscript SCTK_runQC.R \ -P **SEQC** \ -b **BasePath** \ -o OutDirectory \ -s **sample** ...	dropletSCE <- importSEQC(BUStoolsDirs = BasePath, prefix=sample, ...) dropletSCE <- runDropletQC(inSCE = dropletSCE) cellSCE <- dropletSCE[, dropletSCE$dropletUtils_emptyDrops_fdr < 0.01]	**BasePath** [BasePath]/[sample]_sparse_counts_barcodes.csv [BasePath]/[sample]_sparse_counts_genes.csv [BasePath]/[sample]_sparse_molecule_counts.mtx
Optimus	Rscript SCTK_runQC.R \ -P **Optimus** \ -b **BasePath** \ -o OutDirectory ...	dropletSCE <- importOptimus(OptimusDirs = BasePath, ...) cellSCE <- dropletSCE[,which(dropletSCE$dropletUtils_emptyDrops_IsCell)]	**BasePath** [BasePath]/call-MergeCountFiles/ [BasePath]/call-MergeCellMetrics/ [BasePath]/call-MergeGeneMetric [BasePath]/call-RunEmptyDrops/
STARSolo	Rscript SCTK_runQC.R \ -P **STARSolo** \ -b **BasePath** \ -o OutDirectory ...	dropletSCE <- importSTARsolo(STARsoloDirs = BasePath, STARsoloOuts = "Gene/raw", ...) cellSCE <- importSTARsolo(STARsoloDirs = BasePath, STARsoloOuts = "Gene/filtered", ...)	**BasePath** [BasePath]/Gene/raw/ [BasePath]/Gene/filtered/
DropEst	Rscript SCTK_runQC.R \ -P **DropEst** \ -b **BasePath** \ -o OutDirectory ...	dropletSCE <- importDropEst(sampleDirs = BasePath, dataType="raw", ...) cellSCE <- importDropEst(sampleDirs = BasePath, dataType="filtered", ...)	**BasePath** [BasePath]/cell.counts.rds
CountMatrix	Rscript SCTK_runQC.R \ -P **CountMatrix** \ -c **cellFile** \ -r **rawFile** \ -o OutDirectory ...	dropletMM <- data.table::fread(dropletFile) dropletSCE <- constructSCE(data = dropletMM, samplename = samplename) cellMM <- data.table::fread(cellFile) cellSCE <- constructSCE(data = cellMM, samplename = samplename)	**rawFile** path/to/dropletMatrix.mtx **cellFile** path/to/cellMatrix.mtx
SCE	Rscript SCTK_runQC.R \ -P **SceRDS** \ -c **cellFile** \ -r **rawFile** \ -o OutDirectory ...	dropletSCE <- readRDS(rawFile) cellSCE <- readRDS(cellFile)	**rawFile** path/to/dropletSCE.rds **cellFile** path/to/cellSCE.rds
AnnData	Rscript SCTK_runQC.R \ -P **AnnData** \ -C **cellPath** \ -R **rawPath** \ -o OutDirectory \ -s **sample** ...	dropletSCE <- importAnnData(sampleDirs = RawFile, sampleNames= "sampleRaw", ...) cellSCE <- importAnnData(sampleDirs = CellFile, sampleNames= "sampleCell", ...)	**rawPath** [rawPath]/sample.h5ad **cellPath** [cellPath]/sample.h5ad

Supplementary Figure 1. **Import strategies of the SCTK-QC pipeline used to import data.** The last column demonstrates folder structure that is recognized by SCTK-QC pipeline for the dataset generated by each preprocessing tool. The first column shows the command-line implementation of the pipeline. The second column shows the script used to run the pipeline in the R console.

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

References

1. Hwang, B., Lee, J. H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp. Mol. Med.* **50**, 96 (2018).
2. Streets, A. M. *et al.* Microfluidic single-cell whole-transcriptome sequencing. *Proc. Natl. Acad. Sci.* **111**, 7048–7053 (2014).
3. Grün, D., Kester, L. & van Oudenaarden, A. Validation of noise models for single-cell transcriptomics. *Nat. Methods* **11**, 637–640 (2014).
4. Hashimshony, T. *et al.* CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. *Genome Biol.* **17**, 77 (2016).
5. Picelli, S. *et al.* Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nat. Methods* **10**, 1096–1098 (2013).
6. Klein, A. M. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161**, 1187–1201 (2015).
7. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
8. Lun, A. T. L. *et al.* EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. *Genome Biol.* **20**, 63 (2019).
9. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data. *Cell Syst.* **8**, 281-291.e9 (2019).
10. McGinnis, C. S., Murrow, L. M. & Gartner, Z. J. DoubletFinder: Doublet Detection in Single-Cell RNA Sequencing Data Using Artificial Nearest Neighbors. *Cell Syst.* **8**, 329-337.e4 (2019).
11. Yang, S. *et al.* Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome Biol.* **21**, 57 (2020).
12. Zhao, Q. A mitochondrial specific stress response in mammalian cells. *EMBO J.* **21**, 4411–4419 (2002).
13. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* **15**, e8746 (2019).
14. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
15. Melsted, P. *et al.* Modular and efficient pre-processing of single-cell RNA-seq. (2019).
16. Azizi, E. *et al.* Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment. *Cell* **174**, 1293-1308.e36 (2018).
17. Regev, A. *et al.* The Human Cell Atlas. *Elife* **6**, (2017).
18. Petukhov, V. *et al.* dropEst: pipeline for accurate estimation of molecular counts in droplet-based single-cell RNA-seq experiments. *Genome Biol* **19**, 78 (2018).
19. Germain, P.-L. & Lun, A. scDblFinder: scDblFinder. R package version 1.4.0. (2020).
20. Bais, A. S. & Kostka, D. scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* **36**, 1150–1158 (2020).
21. Allaire, J. *et al.* rmarkdown: Dynamic Documents for R. R package version 2.6. (2020).
22. Zappia, L., Phipson, B. & Oshlack, A. Exploring the single-cell RNA-seq analysis landscape with the scRNA-tools database. *PLoS Comput Biol* **14**, e1006245 (2018).
23. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
24. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat. Biotechnol.* **36**, 411–420 (2018).
25. Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888-1902.e21 (2019).
26. Andrews, S. A Quality Control Tool for High Throughput Sequence Data. (2010).
27. Wang, L., Wang, S. & Li, W. RSeQC: quality control of RNA-seq experiments. *Bioinformatics* **28**, 2184–2185 (2012).
28. Merkel, D. Docker: lightweight linux containers for consistent development and deployment. vol. 239 2 (2014).

Acknowledgements

This work was funded by the National Library of Medicine (NLM) R01LM013154-01 (JDC, MY), the National Cancer Institute (NCI) Informatics Technology for Cancer Research (ITCR) 1U01 CA220413-01 (WEJ), the National Institute of General Medical Sciences of the National Institutes of Health (NIGMS) T32GM100842

bioRxiv preprint doi: https://doi.org/10.1101/2020.11.16.385328; this version posted February 14, 2021. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

(YK, SB, AL) and the NCI Human Tumor Atlas Network (HTAN) Pre-Cancer Atlas (PCA) 1U2CCA233238 (JDC).

Code Availability

All code used to generate the analysis is available at https://github.com/campbio/Manuscripts/SCTK-QC.