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], Zhe Wang[1,2], Salam Alabdullatif[2], Yichen Wang[2], Vidya Akavoor[2,3], Xinyun Cao[3], Irzam Sarfraz[2], Frederick Jansen[3], W. Evan Johnson[1,2], Masanao Yajima[4], Joshua D. Campbell[1,2].

1. Bioinformatics Graduate Program, Boston University, Boston, MA, USA.
2. 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

Abstract

Performing comprehensive quality control is necessary to remove technical or biological artifacts in single-cell RNA sequencing (scRNA-seq) data. Artifacts in the scRNA-seq data, such as doublets or ambient RNA, can also hinder downstream clustering and marker selection and need to be assessed. While several algorithms have been developed to perform various quality control tasks, they are only available in different packages across various programming environments. No standardized workflow has been developed to streamline the generation and reporting of all quality control metrics from these tools. We have built an easy-to-use pipeline, named SCTK-QC, in the *singleCellTK* package that generates a comprehensive set of quality control metrics from a plethora of packages for quality control. We are able to import data from several preprocessing tools including *CellRanger*, *STARSolo*, *BUSTools*, *dropEST*, *Optimus,* and *SEQC*. Standard quality control metrics for each cell are calculated including the total number of UMIs, total number of genes detected, and the percentage of counts mapping to predefined gene sets such as mitochondrial genes. Doublet detection algorithms employed include *scrublet*, *scds*, *doubletCells*, and *doubletFinder*. *DecontX* is used to identify contamination in each individual cell. To make the data accessible in downstream analysis workflows, the results can be exported to common data structures in R and Python or to text files for use in any generic workflow. Overall, this pipeline will streamline and standardize quality control analyses for single cell RNA-seq data across different platforms.

Introduction

Single-cell RNA-sequencing (scRNA-seq) has been instrumental in providing detailed insights into cellular heterogeneity, which is key for tissue development and disease pathogenesis, at a resolution that was previously unattainable with bulk RNA sequencing[1]. By dissociation of tissue prior to sequencing, it is now possible to discern the genetic profile of an individual cell.

Despite its utility, scRNA-seq data is susceptible to technical noise. The library size of each cell, determined by its number of unique molecular identifiers (UMIs), may be diminished due to cell lysis or faulty amplification during sequencing. Additionally, the number of features expressed per cell, measured as the total number of features with non-zero expression per cell, could be reduced as a result of defective capture of cDNA during the library preparation protocol. Furthermore, a high degree of expression of mitochondrial genes in a scRNA-seq library may indicate apoptosis or damage to cell membranes, or poor sample quality. Multiplets, which arise from multiple cells being incorrectly sorted into a single droplet, can result in an artificial hybrid expression profile and are a common issue as well in scRNA-seq data, with doublets making up more than 97% of all multiplets[2]. A common strategy is to correct for existing doublets within the data by simulating doublets in silico through random combination of expression profiles and removing barcodes which cluster with the simulated doublets in the PCA space[2, 3]. Ambient RNA, which are RNA molecules that have been released from cells that could have been damaged or undergone apoptosis during sequencing, may also get incorporated into another droplet, leading to contamination[4]. A Bayesian approach has been previously developed to separate the expression of a cell into two separate multinomial distributions of the true cell population and the contamination[4]. It is also possible for ambient RNA to be incorporated into an empty droplet not containing a cell, which will need to be removed. One approach to distinguish empty droplets from cell containing droplets is to characterize the contamination from ambient RNA in the dataset and determine if each barcode deviates from the contamination model[5]. If not controlled for, these factors will cause non-viable cells to appear as a distinct cell type, and confound the identification of true cell types. As such, quality control is a crucial step in scRNA-seq data analysis.

An increase in the number and type of quality control methods, which are implemented in different packages across various programming environments, has necessitated the creation of a standardized, easy to use system for running quality control[6]. Currently, there are no standardized workflows that can streamline the process of generating quality control metrics from all of these tools. In order to address these limitations, we have developed a novel pipeline, called SCTK-QC, within the *singleCellTK* R package which can import data from multiple samples from a variety of preprocessing tools, apply a multitude of different tools to generate comprehensive sets of QC metrics, and visualize these data as intuitive plots and detailed accessible reports.

Results

Overview

The SCTK-QC quality control pipeline accessible through the *singleCellTK* package in R/Bioconductor (**Figure 1**). After alignment of the raw sequencing data and correcting for UMIs, the major steps in analysis of droplet-based scRNA-seq data include: 1) import of the raw gene-barcode matrix and/or the filtered matrix assumed to only contain cells, which we will term as the Droplet Matrix and the Cell Matrix, respectively, 2) detection and exclusion of empty droplets in the Droplet Matrix, 3) calculation of quality control metrics on the Cell Matrix, 4) visualization of the quality control metrics, and 5) export of the data.

Data import

Import of scRNA-seq data from external preprocessing tools is carried out through a set of functions implemented within the SingleCellTK package. Supported algorithms include CellRanger from 10X Genomics, BUStools[7], STARSolo[8], as well SEQC[9], Optimus[10], and dropEST[11]. The dataset is stored within the pipeline as a *SingleCellExperiment* S4 object[12], where the quality control metrics generated in the pipeline will be stored in the “colData” slot alongside other cell-level annotations. For reproducibility, the parameters used to run the functions within the pipeline will be stored in the “metadata” slot. Additionally, the expression data will go into the “assays” slot, while the feature-level information will be contained in the “rowData” slot.

Generation of quality control metrics

Many quality control algorithms have been included in SCTK-QC as R functions. emptyDrops and barcodeRanks from the dropletUtils[5] package are used for the detection of empty droplets in the Droplet Matrix. The *addPerCellQC* function from *scater*[13] will compute general quality control metrics on a *SingleCellExperiment* object including the total UMI and feature count per cell. Additionally, *addPerCellQC* is able to compute the expression of gene sets supplied by the user, which may be useful in cases such as the measurement of mitochondrial genes. *scrublet*[2], *scDblFinder*[14], *DoubletFinder*[3], and the *cxds*, *bcds* and *cxds_bcds_hybrid* models from *SCDS*[15] are utilized for doublet detection in the Cell Matrix. The *decontX*[4] algorithm in the *Celda* package is utilized to determine the level of ambient RNA contamination in the dataset (**Table 1**).

Comparison to other tools

The pipeline supports various types of input, including data generated from different preprocessing tools, SingleCellExperiment objects, h5 files and count matrices. While other QC tools perform specific quality steps, the singleCellTK quality control pipeline supports full scRNA-seq analysis workflow, including general quality metric, doublet detection and ambient RNA corrections (**Table 2**). Besides, the pipelines stores result in common data structures, which facilitates downstream analysis in different analysis workflows.

Generation of comprehensive quality control reports

Rmarkdown reports are reproducible documents that support a variety of both static and dynamic output formats. They use the markdown syntax that allows their conversion to many other types of documents (e.g., .html, .pdf formats). Rmarkdown reports have been widely used in the bioinformatics community as they facilitate ease of sharing and executing the embedded R code. SCTK-QC supports the export of the QC output into comprehensive Rmarkdown reports. The functions *reportDropletQC, reportCellQC, and reportQCTool* make use of algorithm-specific Rmarkdown templates to generate HTML reports with the visualizations of QC metrics. These reports provide a detailed annotation of the QC algorithms and the output results (**Figure 2**).

Export to common data structures

Different software packages utilize varying data containers to store and retrieve scRNA-seq data[17]. To facilitate downstream analysis in multiple platforms, the *SingleCellTK* package provides several options to export the data in one or more formats.

The *exportSCEtoFlatFile* function writes all slots of the *SingleCellExperient* object - colData, rowData, reducedDims, altExps - into text files, while the metadata slot of the object is exported as an RDS file in a list data structure. All exported files can be optionally zipped into a “gz.txt” format. The *exportSCEtoAnnData* function exports the data into a Python annotated data matrix (AnnData) object, which is analogous to the *SingleCellExperiment* object in the Python language. The function calls the *AnnData.write_h5ad* function, which exports the *SingleCellExperimentObject* into a .h5ad file format which can subsequently be compressed in a “gzip” or “lzf” format. Additionally, the user can specify which assay to set as the primary matrix in the output AnnData object.

Example quality control of PBMC datasets using SCTK_QC

To demonstrate the utility of SCTK-QC, we used the 10x Genomics 1K healthy donor Peripheral Blood Mononuclear Cell (PBMC) dataset obtained using both v2 and v3 Chromium chemistries. We downloaded the raw reads in the FASTQ format 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. We then followed instructions from the 10x Genomics portal to build custom references for Gencode GRCh38 v27 and v34 separately. Read counts 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 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 imported into SCTK using the importCellRanger function and their quality metrics were obtained by running the SCTK runCellQC method. The general QC metrics and decontX decontamination score for four 10x PBMC 1k data sets was visualized as violin plots across 4 data sets. (Figure 3) No significant difference was observed in the total read counts and the number of features detected per cell between the PBMC datasets aligned to different versions of Gencode references. However, 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. Additionally, SCTK-QC revealed the improved ability of the v3 chemistry in controlling ambient RNA contamination as evidenced by the lower DecontX contamination scores.

Discussion

With SCTK-QC, we have sought to streamline and standardize the quality control and visualization steps that are vital to triaging the health of single-cell sequencing runs. 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. However, limitations of scRNA-seq platforms create technical artifacts and challenges such as empty or multiplet droplets, ambient RNA, and poor quality cells. Thus, rigorous quality control measures are needed to evaluate the quality of individual experiments. Tools like Fastqc and MultiQC have previously enabled extensive quality assessment of preceding genomic data types. Similarly, SCTK enables a holistic approach to single-cell data analysis by integrating several publicly available tools to provide a common entry point for performing the critical task of estimating and visualizing droplet and cell quality metrics.

Here, we present a novel pipeline within the singleCellTK R package, SCTK-QC, that provides comprehensive sets of QC metrics for scRNA-seq analysis. This pipeline introduces a set of import functions that are able to import scRNA-seq data output generated with different preprocessing tools. SCTK-QC integrates a vast number of existing tools and provides various QC metrics, including general summary statistics of data quality, doublet detection and ambient RNA contamination and correction.

By leveraging the widely adapted SingleCellExperiment object, SCTK-QC provides a standardized method to compute, store, visualize, and export QC metrics and associated metadata, which can subsequently be interfaced with other downstream tools by exporting the sample data and the metrics produced by SCTK-QC in R and Python-compatible data structures. SCTK-QC also offers rich reporting of results that includes publication-ready figures and tabulated summaries from the outputs of various functions in the HTML format.

We have created several vignettes and in-depth walkthroughs for installation and analysis workflows for multifarious use-cases, including a comprehensive reference library on Bioconductor and the accompanying singleCellTK website (https://www.sctk.science/). For convenient portability of the pipeline between operating systems, we have included scripts to set up the Conda or Python virtual environments that meet all cross-platform dependency requirements. Because SCTK-QC integrates numerous tools written in different languages, we have worked to resolve potential package dependency and versioning issues by building Docker and Singularity images of SCTK-QC, freely available through DockerHub.

The modular architecture of SCTK-QC will allow for easy integration of new tools as they are made available. Further, we envision SCTK-QC enabling multi-modal quality, as single-cell approaches capacitate simultaneous quantitative measurements of RNA with proteins and other cellular moieties. We therefore aim to expand the capabilities of SCTK-QC to support the newer flavors of single-cell sequencing, including but not limited to scATACseq, CITEseq, and SMARTseq. SCTK-QC was developed with the needs of the non-computational scientist in mind. While the command-line workflow is flexible and simple to use, future versions of SCTK-QC will also include an R Shiny-based graphical user interface with intuitive plug-and-play modules.

Methods

Accessibility and Reproducibility

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) or Bioconductor (https://www.bioconductor.org/packages/release/bioc/html/singleCellTK.html). 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 (https://hub.docker.com/r/campbio/sctk_qc)[16]. All dependencies of the *singleCellTK* package are included in the Docker image and the quality control pipeline can be executed with a single docker run. Users may specify parameters used for each QC function by providing a YAML file to the pipeline with argument -y. In addition, the same docker image is deployed on Terra to enable computation on cloud clusters.

1. Hwang, B., Lee, J.H. & Bang, D. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Exp Mol Med* **50**, 96 (2018).
2. Wolock, S.L., Lopez, R. & Klein, A.M. Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data. *Cell Syst* **8**, 281-291.e289 (2019).
3. 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.e324 (2019).
4. Yang, S. et al. Decontamination of ambient RNA in single-cell RNA-seq with DecontX. *Genome Biol* **21**, 57 (2020).
5. 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).
6. Luecken, M.D. & Theis, F.J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol Syst Biol* **15**, e8746 (2019).
7. Melsted, P. et al. (BioRxiv; 2019).
8. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15-21 (2013).
9. Azizi, E. et al. Single-Cell Map of Diverse Immune Phenotypes in the Breast Tumor Microenvironment. *Cell* **174**, 1293-1308.e1236 (2018).
10. Regev, A. et al. The Human Cell Atlas. *Elife* **6** (2017).
11. 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).
12. Amezquita, R.A. et al. Orchestrating single-cell analysis with Bioconductor. *Nat Methods* **17**, 137-145 (2020).

13. McCarthy, D.J., Campbell, K.R., Lun, A.T. & Wills, Q.F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics* **33**, 1179-1186 (2017).
14. Germain, P.-L. & Lun, A. (https://github.com/plger/scDblFinder.; 2020).
15. Bais, A.S. & Kostka, D. scds: computational annotation of doublets in single-cell RNA sequencing data. *Bioinformatics* **36**, 1150-1158 (2020).
16. Merkel, D., Vol. 239 2 (*Linux Journal*; 2014).
17. 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).

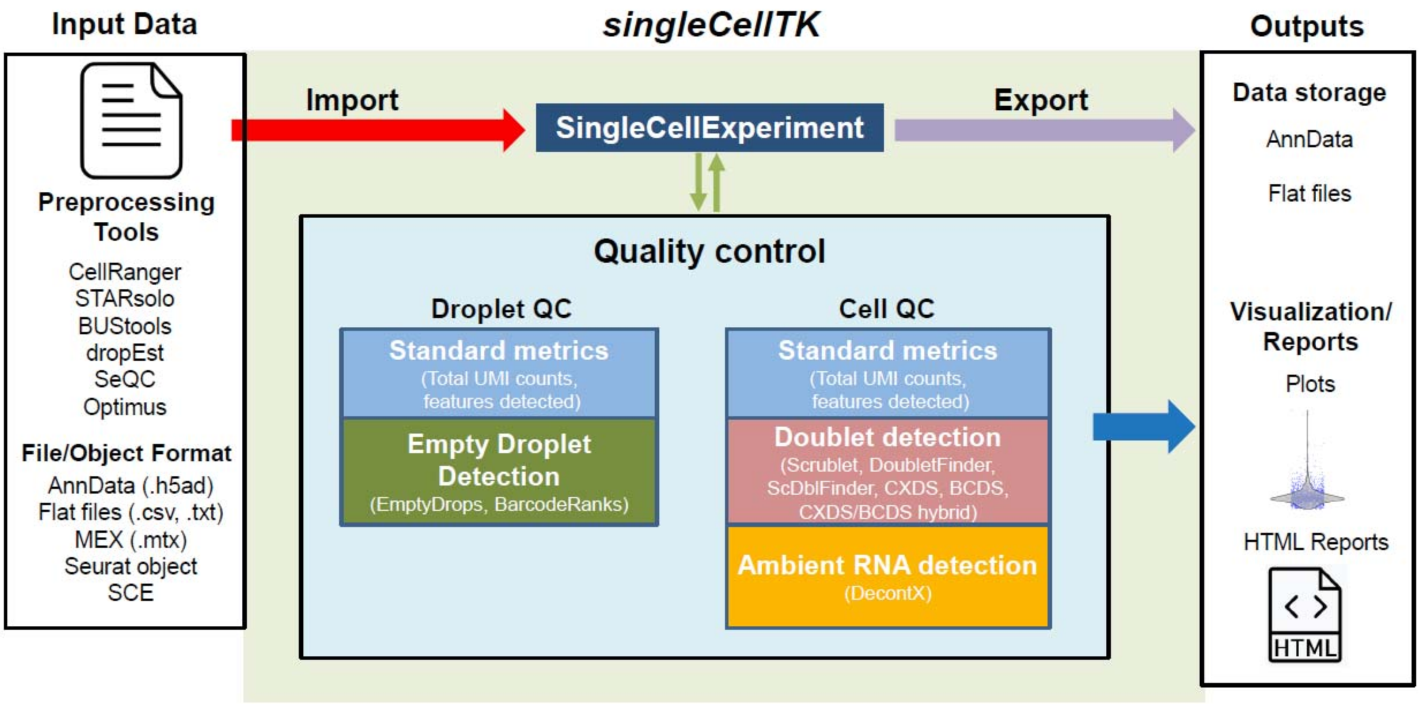
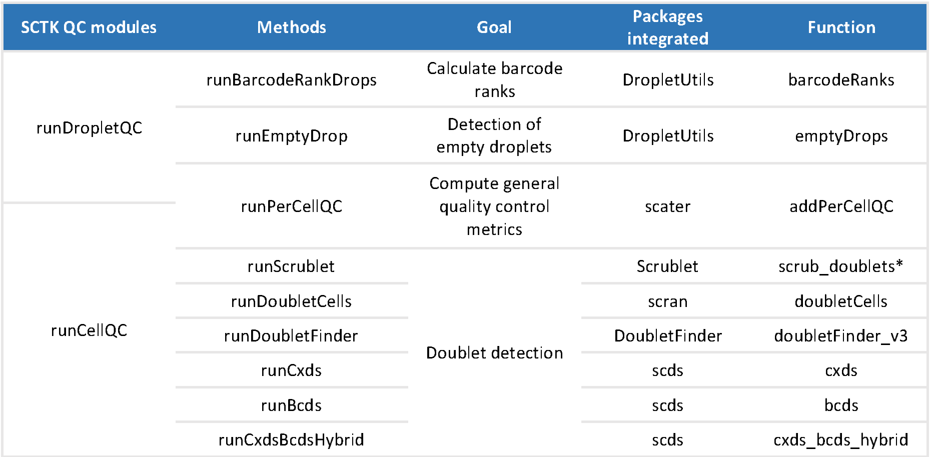


Figure 1. The SCTK-QC pipeline is developed in R and can take as input datasets generated from various preprocessing tools. The pipeline incorporates various third-party softwares to perform quality control, which includes calculation of general quality control metrics and the detection of empty droplets, doublets, and ambient RNA contamination. Data visualization, and report generation can be subsequently performed on the imported dataset based on user specified parameters. SingleCellTK utilizes the SingleCellExperiment R object to store the imported data and the metrics thus computed, which may be exported as a Python AnnData object, or as .txt flat files.



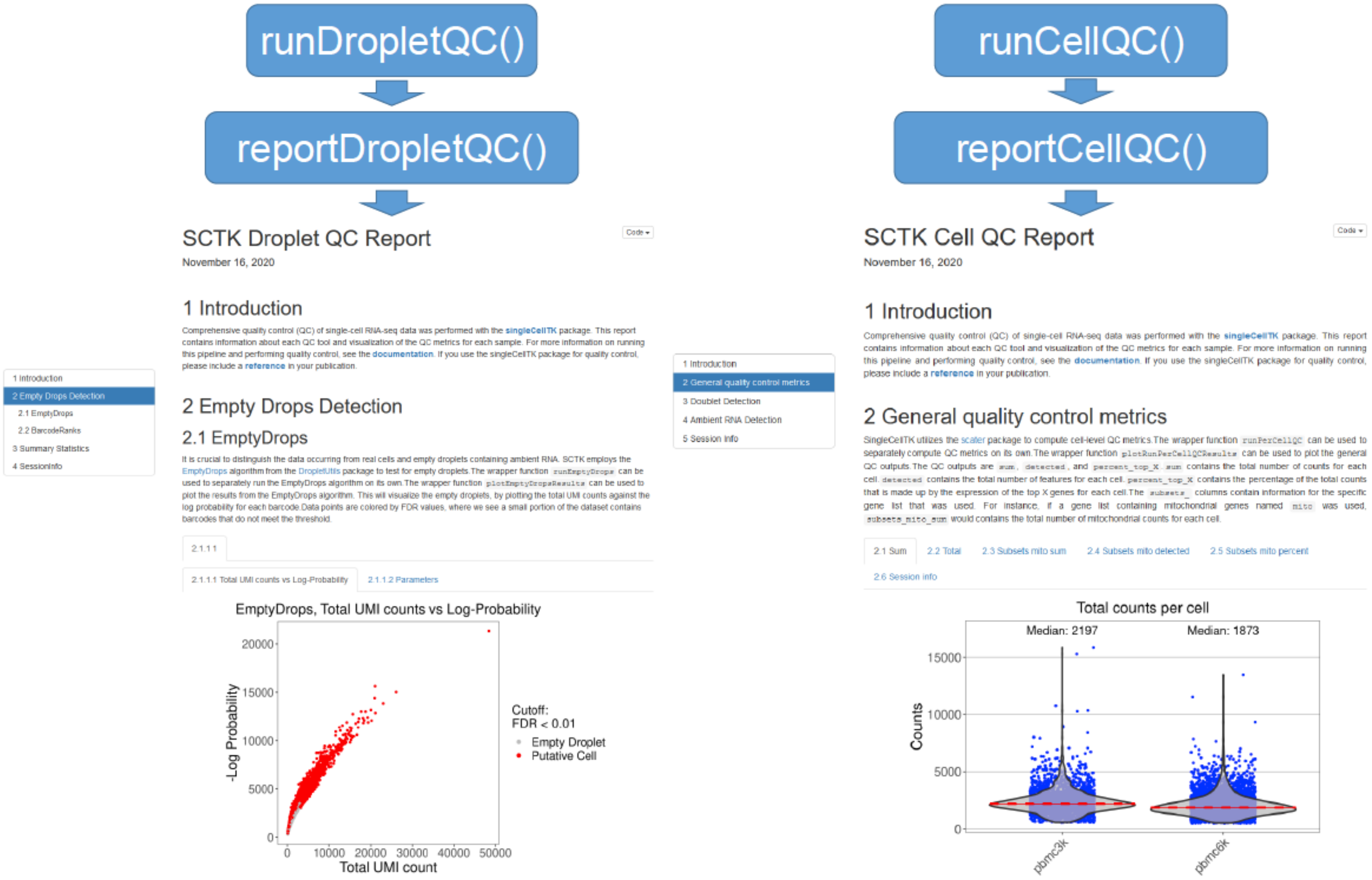
SCTK QC modules	Methods	Goal	Packages integrated	Function
runDropletQC	runBarcodeRankDrops	Calculate barcode ranks	DropletUtils	barcodeRanks
	runEmptyDrop	Detection of empty droplets	DropletUtils	emptyDrops
runCellQC	runPerCellQC	Compute general quality control metrics	scater	addPerCellQC
	runScrublet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

	runDecontX	Detect ambient RNA contamination	celda	decontX

Table 1. The diverse algorithms and their corresponding SCTK-QC wrapper functions that are used to generate quality control metrics in SCTK-QC pipeline.

	SCTK	PIVOT	Seurat	ascend	scRNABatchQC	Adobo	SCONE	SCHNAPPs	iS-CellR	Ganatum	ASAP browser
Input format											
•SCE Object	✓				✓		✓	✓	✓		
•Seurat Object	✓		✓								
•h5	✓										✓
•LOOM											✓
•BUStools	✓										
•SEQC	✓										
•STARSolo	✓										
•Optimus	✓										
•DropEst	✓										
•10x Genomics	✓		✓		✓				✓		
•Count matrix	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
•RSEM							✓				
Ambient droplets detection	✓										
General QC Metrics											
•Total counts	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
•Number of features detected	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
•Mitochondrial gene count	✓	✓	✓	✓	✓	✓			✓		✓
Doublet detection											
•doubletCells	✓										
•Scrublet	✓										
•doubletFinder	✓										
•scds	✓										
Shiny App / interactive	✓	✓					✓	✓	✓	✓	✓
docker	✓	✓							✓		✓
HTML Report	✓			✓	✓						✓
Output format											
•RDS	✓			✓	✓			✓			
•hdf5	✓										✓
•.txt Flatfile	✓							✓			
•pickle						✓					
•joblib						✓					

Table 2. Comparison of singleCellTK quality control pipeline with other popular QC tools. SCTK-QC pipeline supports various types of input, full scRNA-seq quality control pipeline and supports common data structures for data storage.



SCTK Droplet QC Report

November 16, 2020

1 Introduction
2 Empty Drops Detection
2.1 EmptyDrops
2.2 BarcodeRanks
3 Summary Statistics
4 SessionInfo

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.

2.1.1

2.1.1.1 Total UMI counts vs Log-Probability | 2.1.1.2 Parameters

SCTK Cell QC Report

November 16, 2020

1 Introduction
2 General quality control metrics
3 Doublet Detection
4 Ambient RNA Detection
5 Session Info

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.

2.1 Sum | 2.2 Total | 2.3 Subsets mito sum | 2.4 Subsets mito detected | 2.5 Subsets mito percent | 2.6 Session info

Figure 2. Reporting architecture in *singleCellTK*. The functions *runDropletQC()* and *runCellQC()* apply the corresponding algorithms on the input data. The functions *reportDropletQC(), reportCellQC*() generate the reports in the .html format. Examples of *runDropletQC (on the left)* and *runCellQC (on the right)* reports are presented.

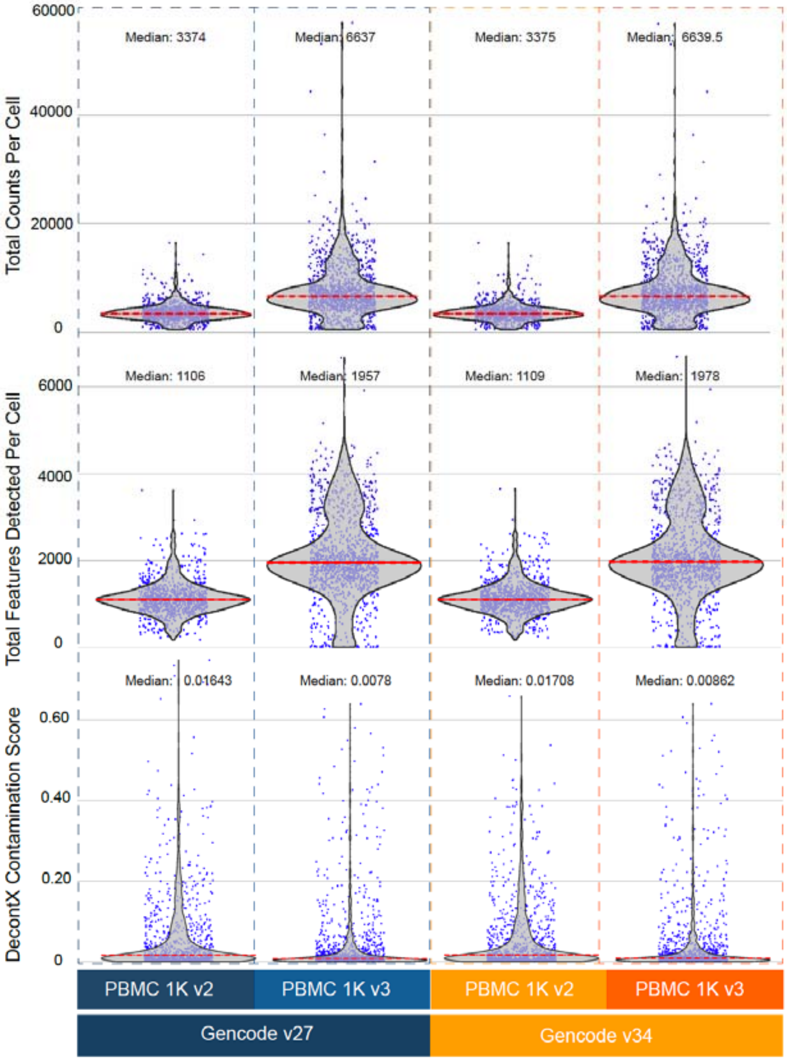
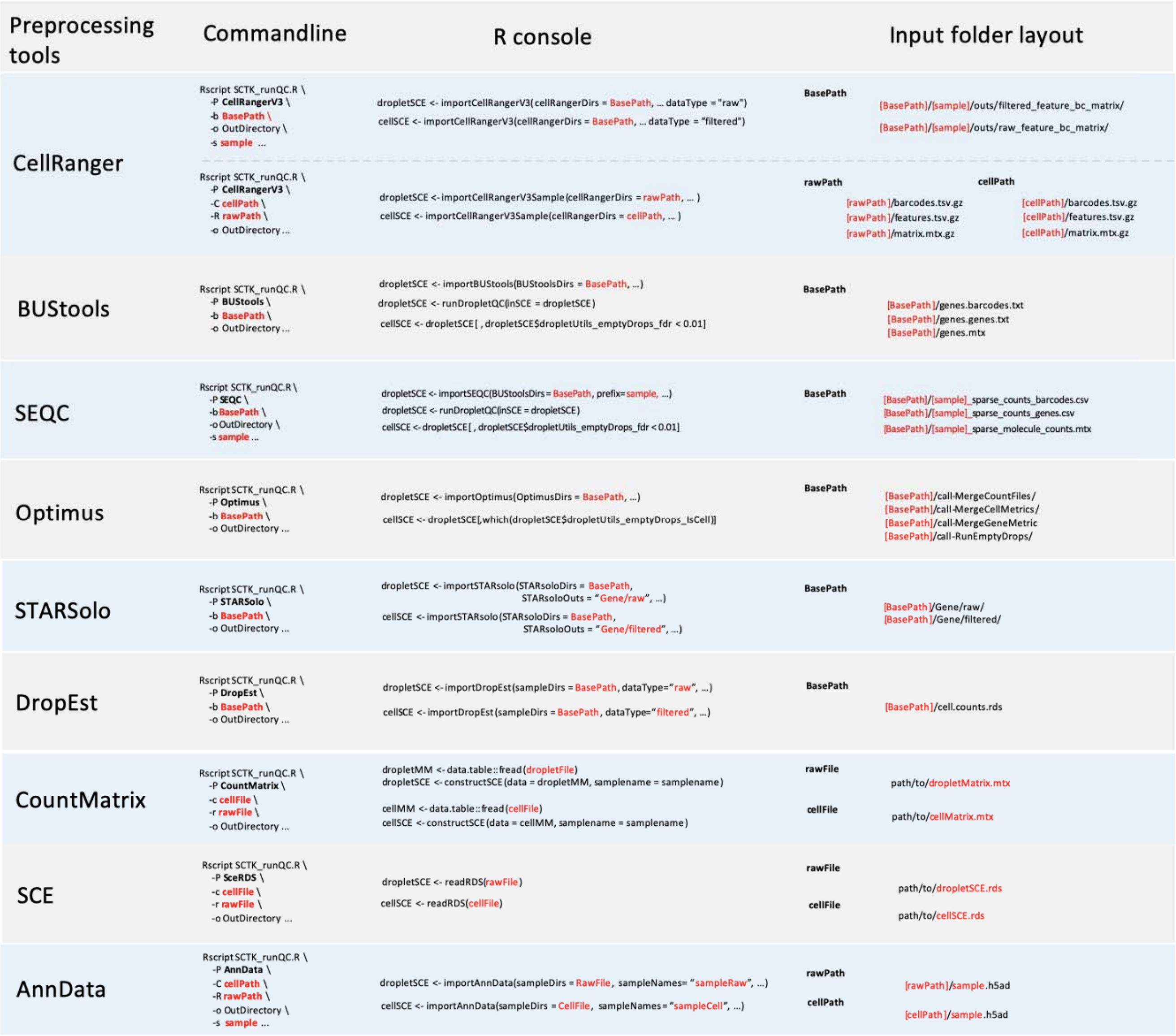


Figure 3. Violin plots of QC metrics generated by SCTK-QC from the 10x Genomics 1K healthy donor Peripheral Blood Mononuclear Cell (PBMC) datasets. SCTK-QC reveals the higher capture sensitivity of the 10x v3 Chromium chemistry as well as its ability to minimize ambient RNA contamination in comparison with the v2 Chromium chemistry.



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.