# Single Cell RNASeq Walkthrough, R v4.2

Table of Contents

[Single Cell RNASeq Walkthrough, R v4.2 1](#_Toc128756030)

[Background/System setup 1](#_Toc128756031)

[R packages: 1](#_Toc128756032)

[Per sample QC 1](#_Toc128756033)

[Import workflow directory from GitHub 1](#_Toc128756034)

[cellRangerOut directory 2](#_Toc128756035)

[groups.tab 3](#_Toc128756036)

[contrasts.tab 3](#_Toc128756037)

[Additional directories: 3](#_Toc128756038)

[Set up bash and swarm scripts 3](#_Toc128756039)

[Outputs 4](#_Toc128756040)

[Sample combination and batch correction 5](#_Toc128756041)

[Bash scripts 5](#_Toc128756042)

[Outputs 5](#_Toc128756043)

[Preliminary differential expression 5](#_Toc128756044)

## Background/System setup

* Biowulf HPC
* R version 4.2.0 or 4.2.2

## R packages:

* Seurat

## Per sample QC

* Import workflow directory from Github
* Prepare cellRangerOut directory
* Prepare groups.tab
* Prepare contrasts.tab
* Additional directories
* Set up bash and swarm scripts
* Outputs

### Import workflow directory from GitHub

Retrieve the current workflow from <https://github.com/wong-nw/scRNASeq_2023>

Move the directory titled “workflow” into the current working director. The working directory should also contain the subsequent setup files.

### cellRangerOut directory

Stores most of the 10X CellRanger output directory. Directory structure should be designed as follows:

CellRangerOut

---SampleOutput (GEX +/- CITE, VDJ, ATAC)

------outs

---------mandatory output files

In the event that output files are being provide directly from the collaborator, as opposed to from the sequencing teams, certain mandatory outputs should be arranged to simulate their locations within a 10X CellRanger output

* GEX and CITESeq: filtered\_feature\_bc\_matrix.h5.
* VDJ (e.g. TCR and BCR): filtered\_contig\_annotations.csv.
* ATACSeq: (per Signac) filtered\_peak\_bc\_matrix.h5 and fragments.tsv.gz.

An example of a sample directory setup is shown here.

cellRangerOut/

├── Sample1\_GEX

│   └── outs

│   ├── analysis

│   │   ├── ...

│   │   │

│   ├── cloupe.cloupe

│   ├── filtered\_feature\_bc\_matrix

│   │   ├── barcodes.tsv.gz

│   │   ├── features.tsv.gz

│   │   └── matrix.mtx.gz

│   ├── filtered\_feature\_bc\_matrix.h5

│   ├── ...

├── Sample1\_TCR

│   └── outs

│   ├── ...

│   ├── filtered\_contig\_annotations.csv

│   ├── filtered\_contig.fasta

│   ├── filtered\_contig.fastq

│   ├── ...

├── Sample1\_ATAC

│   └── outs

│   ├── analysis

│   │   ├── ...

│   │   │

│   ├── cloupe.cloupe

│   ├── filtered\_peak\_bc\_matrix

│   │   ├── barcodes.tsv.gz

│   │   ├── features.tsv.gz

│   │   └── matrix.mtx.gz

│   ├── filtered\_peak\_bc\_matrix.h5

│   ├── ...

│   ├── fragments.tsv.gz

│   ├── ...

├── ...

### groups.tab

Tab delimited file with each line including the relevant files for each sample. Can include lines to account for gene expression, CITESeq, V(D)J, and ATACSeq data. For a sample that contains all possible data types:

Sample1\_GEX Group1 Sample1Name gex

Sample1\_GEX Group1 Sample1Name cite

Sample1\_ATAC Group1 Sample1Name atac

Sample1\_VDJ Group1 Sample1Name vdj

Sample2\_GEX Group2 Sample2Name gex

…

Sample2\_VDJ Group2 Sample2Name vdj

### contrasts.tab

Tab-delimited file containing each pairwise comparison.

Group2 Group1

### Additional directories:

The following directories need to be prepared to store the results:

QC

QC/images

filtered

### Set up bash and swarm scripts

Two primary bash files are required to run the sample QC stage. The per sample QC call is in <github>/bashScripts/setupQC.bash. The script calls R and needs to have its parameters adjusted to reflect the experiment. The primary call within the script needs to be adjusted:

Rscript workflow/scripts/scrna\_QC.R cellRangerOut/ QC/ $1 <commaSepResolutionString> <species> <annotReference>

The parameters are as follows:

* commaSepResolutionString: A comma separated string of resolutions to perform clustering. Smaller resolutions produce fewer and larger clusters. E.g. “0.4,0.6,0.1,1.2”
* species: This is actually more used to determine the annotation references. Current references are either “mm10” or “hg38”
* annotReference: This calls a basic annotation reference for SingleR to run from celldex. Human references include:
  + HPCA: Human Primary Cell Atlas
  + BP\_encode: Blueprint ENCODE
  + monaco: Monaco immune dataset
  + immu\_cell\_exp: Database of Immune Cell Expression

The swarm script qcRun.swarm should call the following for each sample, as identified in groups.tab and the cellRangerOut directory:

bash setupQC.bash <sample>

The swarm job should be run with at least 64 GB of memory on 4 threads for 6 hours. Example:

swarm -t 4 -g 64 --time 6:00:00 -f qcRun.swarm

### Outputs

The primary outputs from the per sample QC are a sample QC report in html format in the “QC” directory and the filtered Seurat files for each object stored in the “filtered” directory.

.

├── cellRangerOut

│   ├── ...

├── contrasts.tab

├── filtered

│   ├── Sample1.rds

│   ├── Sample2.rds

│   ├── ...

│   └── Sample\_n.rds

├── groups.tab

├── ...

├── QC

│   ├── images

│   │   ├── Sample1

│   │   │   ├── cellsRemovedVenn\_Sample1.png

│   │   │   ├── ...

│   │   │   └──

│   │   ├── Sample2

│   │   ├── ...

│   ├── QC\_Report\_Sample1.html

│   ├── ...

│   └── QC\_Report\_Sample\_n.html

├── qcRun.swarm

├── setup\_QC.bash

└── workflow

├── ...

## Sample combination and batch correction

### Setup

Prior to running the bash scripts, instantiate the following directories:

integration

integration/images

### Bash scripts

Three bash scripts are used to create 5 separate files where the samples are combined. For each script, the Rscript call should be modified to represent the sample contrast, among other parameters.

runHarmony.bash

Rscript workflow/scripts/harmony.R filtered integration/<contrast>\_merged.rds integration/<contrast>\_harmonySample.rds integration/<contrast>\_harmonyGroup.rds <species> <contrast> <resolution\_string>

runCCA.bash

Rscript workflow/scripts/integrateBatches.R filtered integration/<contrast>\_integrated\_cca.rds <species> <contrast> <resolution\_string>

runRPCA.bash

Rscript workflow/scripts/rpca.R filtered integration/<contrast>\_rpca.rds <species> <contrast> <resolution\_string>

For each of these scripts, the contrast is normally inputted as a hyphen-delimited pair of groups, e.g. Group2-Group1. As with the per sample QC, the species options are limited to mm10 and hg38. The resolution string should be the same comma-separated list of clustering resolutions used in the per sample QC.

### 

All three scripts can then be called in parallel with

runIntegration.swarm

We recommend running the integration scripts with a minimum of 8 threads, 12 hours runtime, and 150 GB memory. This memory allocation and runtime may need to be increased depending on the number of samples included. An example of a call for the swarm script would be:

swarm -t 8 -g 150 --time 12:00:00 -f runIntegration.swarm

runIntegrationReport.bash

After the three scripts are run, which can be confirmed by the presence of 5 RDS files in the “integration” directory, generate the integration report by running runIntegrationReport.bash. The Rscript call in the bash script should be modified as follows:

Rscript workflow/scripts/integrationReport.R <contrast>\_merged.rds <contrast>\_integrated\_cca.rds <contrast>\_harmonyGroup.rds <contrast>\_harmonySample.rds <contrast>\_rpca.rds $PWD/integration/images $PWD/integration <resolution\_string> <contrast> YES NO

Since this script will be importing all the combined RDS files, it is recommended to run this with a minimum of 8 threads and 200GB of memory. As before, this can be adjusted depending on the number of samples included. Example bash call:

bash --mem=200g –cpus-per-task=8 --time=6:00:00 runIntegrationReport.bash

### Outputs

## 

The outputs are all stored in the “integration” directory. Each of the “rds” files specified in the bash scripts should be produced, as well as a QC report containing images of the combination of samples after applying each batch correction technique.

integration

├── <contrast>\_harmonyGroup.rds

├── <contrast>\_harmonySample.rds

├── <contrast>\_integrated\_cca.rds

├── <contrast>\_merged.rds

├── <contrast>\_rpca.rds

├── images

│   ├── ...

└── QC\_Report\_<contrast>.html

## Preliminary differential expression

Adapted from <https://satijalab.org/seurat/articles/sctransform_v2_vignette.html#identify-differential-expressed-genes-across-conditions-1>

Based on the selection of the batch correction technique, the user can then select an identity or pair of identities on which to perform differential expression. These identities can be experimental groups, clusters, cell types, or even a user-selected population. When choosing a single identity, the selected identity will be compared against all other cells in the dataset; when a pair of identities is chosen, a pairwise comparison is performed.

The Satija group now provides code to correct the SCT assay prior to differential expression for more accurate sample-to-sample comparison. An example run of differential expression would look like:

Idents(seuratObj)=”groups”

DefaultAssay(seuratObj)=”SCT”

seuratObj = PrepSCTFindMarkers(seuratObj)

deList = FindMarkers(seuratObj, ident.1=”group1”, ident.2=”group2”, test.use=”MAST”, min.pct=0.1, logfc.threshold = 0.25)

When running differential expression, it must be ensured that the identities being tested are reflected in the “Idents” call on the object. If necessary, a new metadata column can be generated containing the desired cell categories, and then the metadata title should be used in the “Idents” call. Additional parameters to keep in mind are the differential expression test being used (MAST is single-cell specific, but ANOVA and Wilcox, among others can also be used), the minimum percentage (at least one group should have the fraction indicated of cells that express a gene being tested), and the logfc.threshold (the minimum average logFC difference between groups before the test is run). When a full DE list of all available genes is desired, the min.pct and logfc.threshold parameters should be set to 0; the tradeoff will be a significantly increased runtime, as those thresholds are used to limit how many DE calculations are being run.