1) Creating seurat object and QC

Kane Toh*

2023-12-18

Contents

1 Overview

In this first notebook, we load in the results from the split-pipe ParseBio pipeline, convert all sublibraries from all samples into individual Seurat objects, and perform QC on all samples.

Next, we run the standard Seurat pipeline, followed by inferring the cell cycle phase of each cell in the dataset. Finally, we explore the data further by plotting several useful QC metrics to showcase the outcome of our QC, and examine how cells are distributed according to their condition (KO/WT) and sample_labels (KO_BL, WT_BR etc.).

*Genomics and Data Analytics Core (GeDaC), Cancer Science Institute of Singapore, National University of Singapore; kanetoh@nus.edu.sg

2 Loading libraries

```
library(Seurat)  # For scRNAseq analysis
library(dplyr)   # For data cleaning/wrangling
library(tidyr)   # For extra data cleaning tasks
library(stringr) # string manipulation
library(biomaRt) # To convert ensembl_ids to gene symbols
library(ggplot2) # For plotting
library(cowplot) #For placing plots in a grid, similar to patchwork
library(ArchR) #Library for scATAC-seq analysis, but with awesome color palettes
#packageurl <- "https://cran.r-project.org/src/contrib/Archive/Seurat/Seurat_4.4.0.tar.gz"
#packageurl <- "https://cran.r-project.org/src/contrib/Archive/SeuratObject/SeuratObject_4.1.4.tar.gz"
#install.packages(packageurl, repos=NULL, type="source")
#devtools::install_version("Matrix",version = "1.6.1.1")
packageVersion("SeuratObject")
```

```
## [1] '4.1.4'
```

```
packageVersion("Matrix")
```

```
## [1] '1.6.1.1'
```

```
packageVersion("Seurat")
```

```
## [1] '4.4.0'
```

```
## Installation commands
# install.packages(c("dplyr", "tidyr", "stringr", "ggplot2", "cowplot", "remotes"))
# remotes::install_github("satijalab/seurat", "seurat5", quiet = TRUE)
# if (!requireNamespace("devtools", quietly = TRUE)) install.packages("devtools")
# devtools::install_github("GreenleafLab/ArchR", ref="master", repos = BiocManager::repositories())
```

3 Define global variables

```
# Loads in custom functions that I've written to assist with plotting/saving tasks
source("helper.R")

# Path to saving the results fo the analysis
res_path <- "../results/01_qcSeurat/"
if (!dir.exists(res_path)){

  dir.create(res_path)
}else{
  print("Results directory exists")
}
```

```
## [1] "Results directory exists"
```

```r
# Data path specifes where all the outputs from the parseBio split-pipe pipeline is stored.
data_path <- "../data/"
if (!dir.exists(data_path)){
  warning("Directory does not exist: ", data_path)
}else{
  print("Data directory exists")
}
```

```
## [1] "Data directory exists"
```

```r
# RDS path specifies where all the intermediate generated R objects will be saved.
rds_path <- "../rds/"
if (!dir.exists(rds_path)){
  warning("Directory does not exist: ", rds_path)
}else{
  print("RDS directory exists")
}
```

```
## [1] "RDS directory exists"
```

4 Creating QC'ed seurat objects

4.1 Specify file paths to parsebio outputs

```r
sample_conditions <- c("KO-BL/", "KO-TL/", "KO-NN/", "KO-TR/", "WT-TR/", "WT-BR/", "WT-TRBR/")
sublibraries <- c("Sublibrary1/", "Sublibrary2/")
# This creates all combinations to the correct paths
path_df <- expand.grid(data_path, sublibraries, sample_conditions, "DGE_filtered/")
# Creates the full paths to the DGE_filtered/ folder from parsebio
path_df_full <- with(path_df, paste(Var1,Var2,Var3,Var4, sep=""))
```

4.2 Generate seurat objects and QC

Next, we loop over all file paths to generate the seurat objects. Update (18/12/23): On package incompatibitilies. I recently faced an issue with running this code because of incompatibilites arising from the SeuratObject package and the latest Matrix package. See Github Issue. The solution provided in the comment of the issue, which involves downgrading the Matrix package to v1.6.1.1 does not work as the latest SeuratObject packages requires Matrix >=1.6.3. The solution found ultimately involved downgrading to Seurat v4.40, SeuratObject

Update (18/12/23): On capturing all relevant metadata. One of the later QC decisions made was to remove the KO-BL and WT-BR samples as they appeared as outliers: the KO_BL sample does not contain tumor cells whereas the WT_BR samples does. Further discussion with the wet lab biologists suggest that this is likely due to differences in the tumor burden across mice, as shown in the mice bladder weights. Removal of these samples would lead to all KO samples having tumor cells, whilst all normal samples wouldn't. Consequently, the analysis of WT vs KO would be confounded by the presence/absence of tumor cells. A practical lesson learnt here is to *always* ask for all the metadata available before proceeding with any analysis.

```r
# Specifying the QC thresholds
nFeature_RNA <- 300
nCount_RNA <- 300
percent.mt <- 10

res_seurat_original <- c()
res_seurat_subset <- c()
cell_ids <- c()

start.time <- Sys.time()
for (i in 1:length(path_df_full)){
  seuratObj <- loadIntoSeurat(path_df_full[i])
  cell_ids <- append(cell_ids, unique(Idents(seuratObj)))
  res_seurat_original <- append(res_seurat_original, seuratObj)
  seuratObj.2 <- generateQCmetrics(seuratObj, res_path)
  SeuratObj.subset <- seuratObj.2[,seuratObj.2$nCount_RNA > nCount_RNA &
                                   seuratObj.2$nFeature_RNA > nFeature_RNA &
                                   seuratObj.2$percent.mt < 10]
  res_seurat_subset <- append(res_seurat_subset, SeuratObj.subset)
}
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken #Time difference of 26 secs
# SeuratMerged contains the QC'ed (subsetted) object
seuratMerged <- merge(res_seurat_subset[[1]], y = unlist(res_seurat_subset[-1]),
                      add.cell.ids = cell_ids,
                      project = "mbladder_2022")
seuratMerged@meta.data %>% dplyr::group_by(Condition, Type) %>% dplyr::summarise(n=n())

# SeuratMergedOriginal contains the original, non-QC'ed object.
seuratMergedOriginal <- merge(res_seurat_original[[1]], y = unlist(res_seurat_original[-1]),
                      add.cell.ids = cell_ids,
                      project = "mbladder_2022")
```

4.3 Save seurat objects as RDS objects for reproducibility

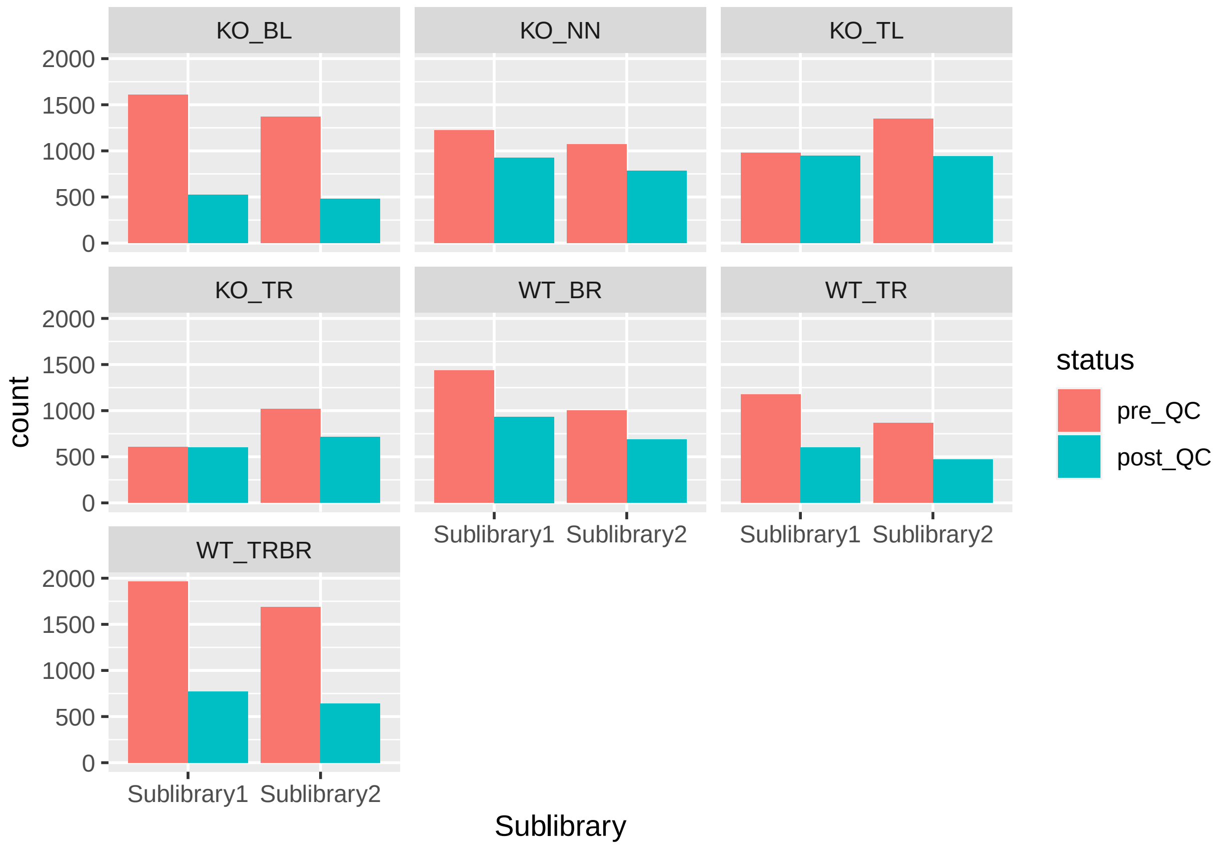
```r
SaveObject(rds_path, seuratMerged, "01_seuratMerged_postQC")
SaveObject(rds_path, seuratMergedOriginal, "01_seuratMerged_beforeQC")
```

5 Extra QC plots

5.1 Number of cells before and after QC

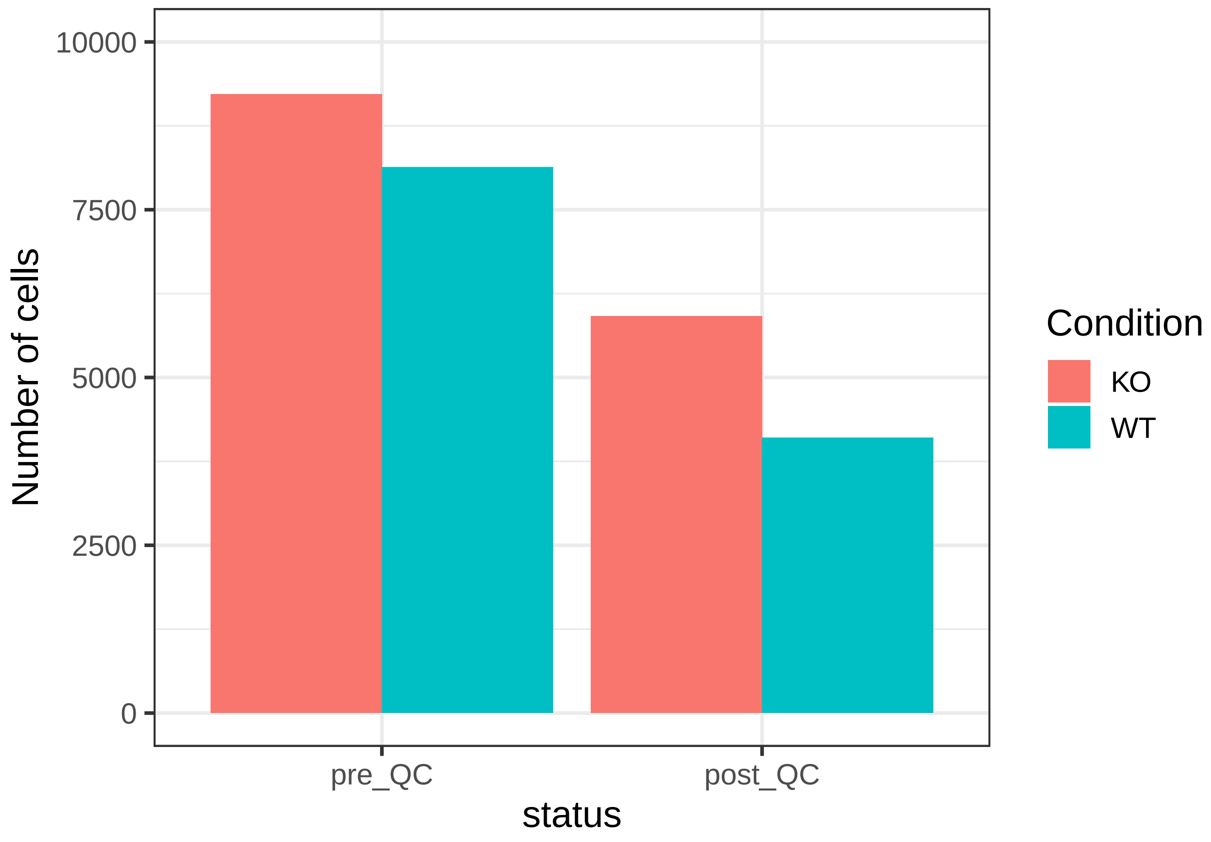
5.2 Number of KO and WT cells before and after QC across samples

```r
subset_metadata <- seuratMerged@meta.data
original_metadata <- seuratMergedOriginal@meta.data
subset_metadata_summary <- subset_metadata %>%
  dplyr::group_by(Condition, Type, Sublibrary) %>%
  dplyr::summarize(post_QC=n())
original_metadata_summary <- original_metadata %>%
  dplyr::group_by(Condition, Type, Sublibrary) %>%
  dplyr::summarize(pre_QC=n())

metadata_summary_combined <- dplyr::left_join(original_metadata_summary, subset_metadata_summary,
                                              by = c("Condition", "Type", "Sublibrary")) %>%
  tidyr::pivot_longer(cols = c("pre_QC", "post_QC"),
                      names_to = "status",
                      values_to = "count")  %>%
  dplyr::mutate(id = paste(Condition,Type,sep="_"))

metadata_summary_combined$status <- factor(metadata_summary_combined$status ,
                                           levels=c("pre_QC", "post_QC"))

cell_counts_by_sample<- ggplot(metadata_summary_combined,
       aes(x=Sublibrary, y= count, fill = status)) +
         geom_col(position='dodge') +
  facet_wrap(~id)
cell_counts_by_sample
```



5.3 Total cells before and after qc

```r
total_cell_counts_qc <- metadata_summary_combined %>%
  dplyr::group_by(status, Condition) %>%
  dplyr::summarise(n=sum(count)) %>%
  ggplot(aes(x = status, y = n))+
  geom_col(aes(fill = Condition), position = "dodge")+
  labs(y = "Number of cells")+
  lims(y = c(0,10000))+
  theme_bw(base_size = 14)
total_cell_counts_qc
```



## Combine

```r
combined <- cowplot::plot_grid(cell_counts_by_sample, total_cell_counts_qc, nrow = 1)
SaveFigure(res_path = res_path,
           plots = combined,
           name = "cell_subset",
           width = 12,
           height = 6)
```

```
## pdf
##   2
```