

Long Covid study scRNA-seq, 4 month time point, data preprocessing

Emily Kibbler

2025-08-10

This pipeline is for analysis of scRNA-seq data from a publicly available data set.

Data source citation:

Phetsouphanh C, [...] Matthews GV. Improvement of immune dysregulation in individuals with long COVID at 24-months following SARS-CoV-2 infection. Nat Commun. 2024 Apr 17;15(1):3315. doi: 10.1038/s41467-024-47720-8. PMID: 38632311; PMCID: PMC11024141.

[Click here to access the data from GEO](#)

This specific script reads in the libraries from samples collected at 4 months post-infection.

The GSM8180643_1_4mths library (pool 1, replicate 1) has a corrupted data file in the 10X data and I cannot read it in, so unfortunately, this object will only have two libraries. There is no pool 1, replicate 2 at all; therefore, this combined object has only 10 patients (5 long COVID, 5 matched controls) with the two technical replicates.

```
library(Seurat)
library(ggpubr)
library(tidyverse)

# This function takes a data set (Seurat object) and a list of desired features and returns a ggarrange
# Features must be something that can be retrieved by Seurat's FetchData function
# Heavy lifting performed by Seurat's VlnPlot function; formatting and tweaking by E. Kibbler

myVlnPlot <- function(dat, feats = c("nFeature_RNA", "nCount_RNA", "percent.mt")) {
  # Structure as a list even if only one feature is given
  if (length(feats) == 1) {
    feats <- c(feats)
  }
  # Initiate empty list to put the plot(s) in
  plots <- c()
  # Plot each feature separately
  for (i in 1:length(feats)) {
    # Use suppressWarnings because the VlnPlot() has a default argument which uses a deprecated argument
    temp <- suppressWarnings(VlnPlot(dat,
      features = feats[i])) +
      theme(axis.text.x = element_blank(),
            axis.title.x = element_blank(),
            legend.position = "none")
    plots <- c(plots, list(temp))
  }
  # Generate and return a plot of panel(s)
  res <- ggarrange(plotlist = plots, nrow = 1)
```

```

    return(res)
}

# Define libraries to read in
ids <- c("GSM8180644_3_4mths",
        "GSM8180645_4_4mths")
mywd <- "./data/covid/GSE262861_RAW/"

Read in data
for (library_id in ids) {
  results <- read.table(paste0(mywd,
                                library_id,
                                "/",
                                library_id,
                                "_combined_results_w_combined_assignments.tsv.gz"),
                        header = T)
  pbmc.data <- Read10X(data.dir = paste0(mywd, library_id, "/10x_dat"))
  pbmc <- CreateSeuratObject(counts = pbmc.data,
                             project = "longcovid",
                             min.cells = 3,
                             min.features = 200)
  pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^\u00d7\u00d7")
  pbmc[["UMI"]] <- rownames(pbmc@meta.data)

  # Assign timepoints
  if (grepl("4mths", library_id)) {timepoint = "4months"}
  if (grepl("8mths", library_id)) {timepoint = "8months"}
  if (grepl("^\u00d7\u00d7", library_id)) {timepoint = "24months"}
  pbmc[["timepoint"]] <- timepoint

  # Plot initial QC
  p <- myVlnPlot(pbmc) %>%
    annotate_figure(top = text_grob(paste(library_id, "initial QC")))
  print(p)
  ggsave(filename = paste0(mywd, library_id, "/qc.png"), p)

  # Assign cells
  pbmc@meta.data <- left_join(pbmc@meta.data,
                                results,
                                by = join_by(UMI == Barcode))
  rownames(pbmc@meta.data) <- pbmc@meta.data$UMI

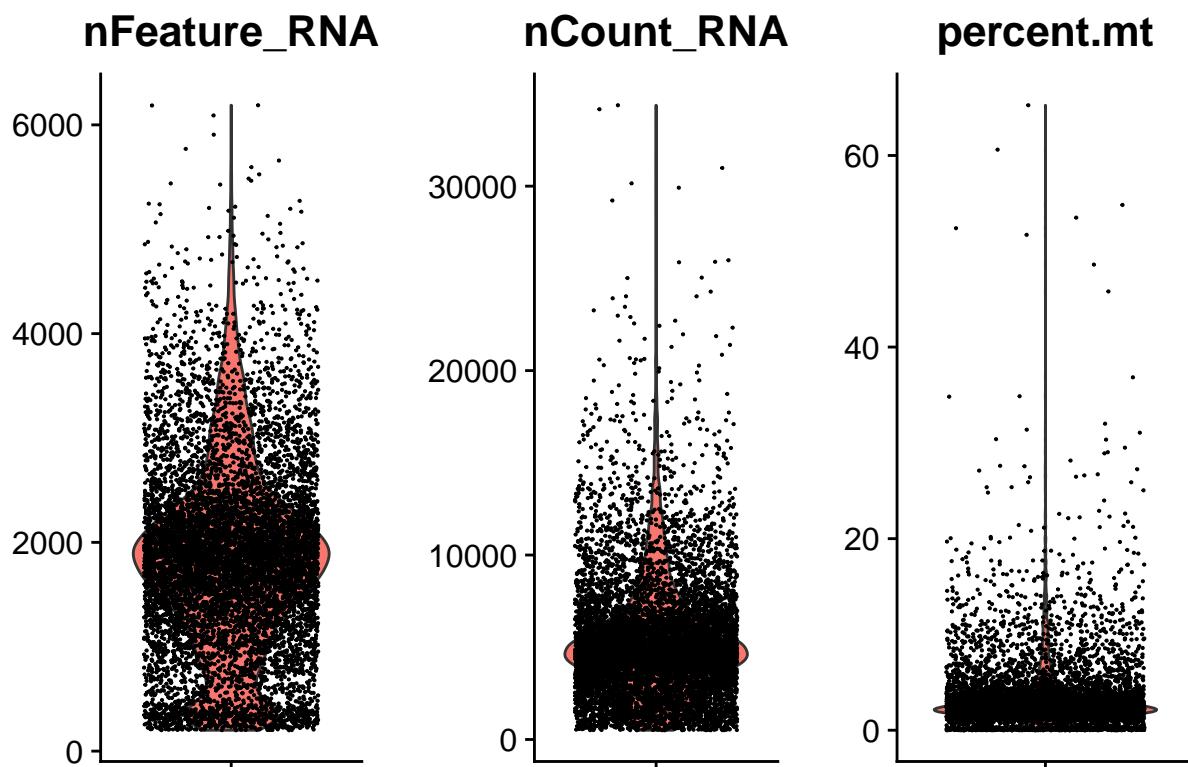
  pool <- str_split_i(library_id, "_", 2)
  # Even-numbered pools are the second replicate
  if (pool %in% c(4, 6, 8)) {
    pbmc[["batch"]] <- "rep 2"
  } else {pbmc[["batch"]] <- "rep 1"}

  pbmc[["library"]] <- library_id

  write_rds(pbmc, file = paste0(mywd, library_id, "/", library_id, ".rds"))
}

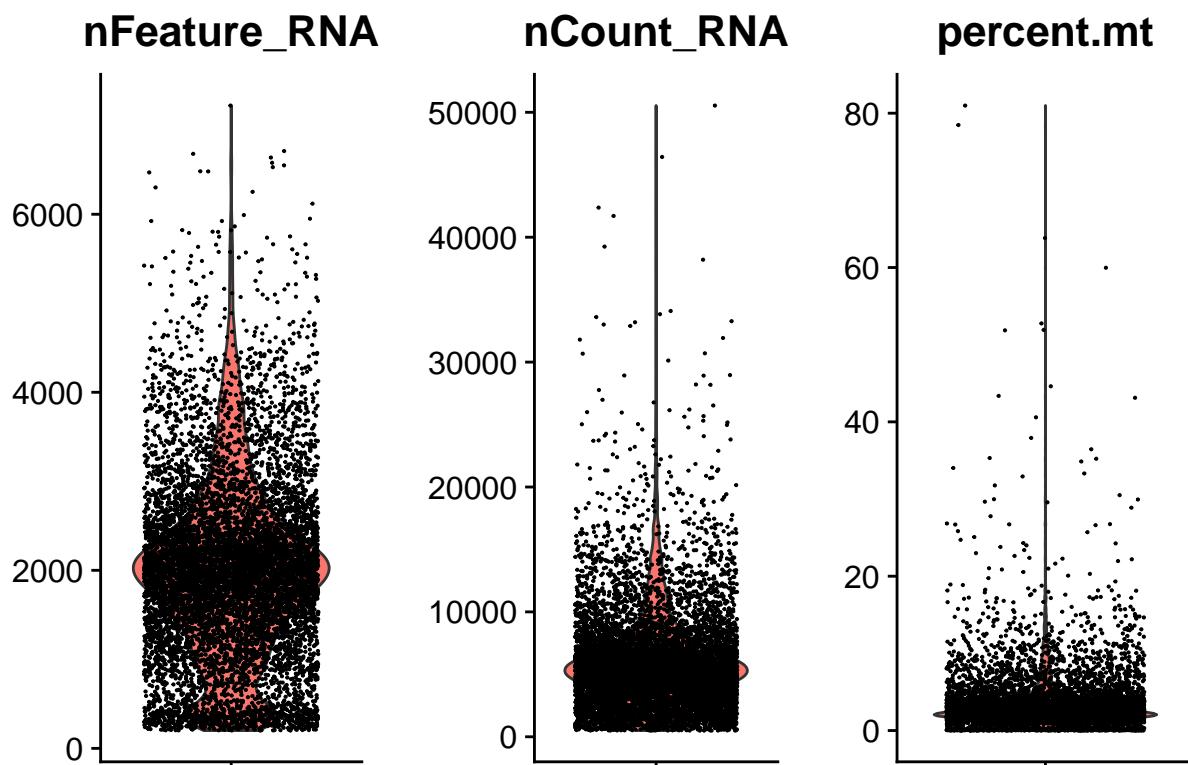
```

GSM8180644_3_4mths initial QC



Saving 6.5 x 4.5 in image

GSM8180645_4_4mths initial QC

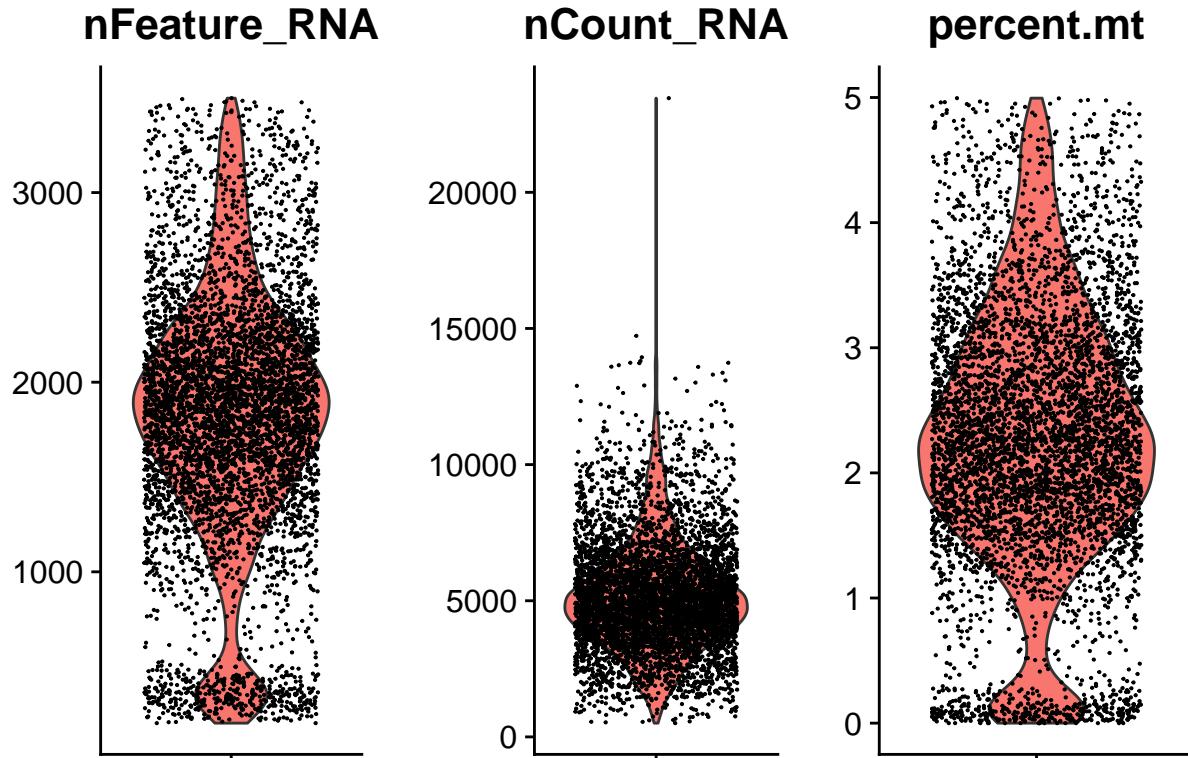


```

## Saving 6.5 x 4.5 in image
Filter and prepare individual data sets
ids <- c("GSM8180644_3_4mths",
        "GSM8180645_4_4mths")
# Empty list, for each batch to be added to
pbmc.list <- list()
# Load and process all data
for (library_id in ids) {
  pbmc <- readRDS(file = paste0(mywd, library_id, "/", library_id, "_EK.rds"))
  pbmc <- subset(pbmc,
                 subset = nFeature_RNA > 200 &
                   nFeature_RNA < 3500 &
                   percent.mt < 5 &
                   Demuxlet_DropletType == "singlet")
  p <- myVlnPlot(pbmc) %>%
    annotate_figure(top = text_grob(paste(library_id, "QC after filtering")))
  print(p)
  ggsave(filename = paste0(mywd, library_id, "/", library_id, "_after_filter_qc.png"), p)
  pbmc <- SCTtransform(pbmc, verbose = FALSE)
  pbmc <- RunPCA(pbmc, verbose = FALSE)
  pbmc <- RunUMAP(pbmc, dims = 1:30, verbose = FALSE)
  pbmc <- FindNeighbors(pbmc, dims = 1:30, verbose = FALSE)
  pbmc <- FindClusters(pbmc, verbose = FALSE)
  pbmc.list[[library_id]] = pbmc
}

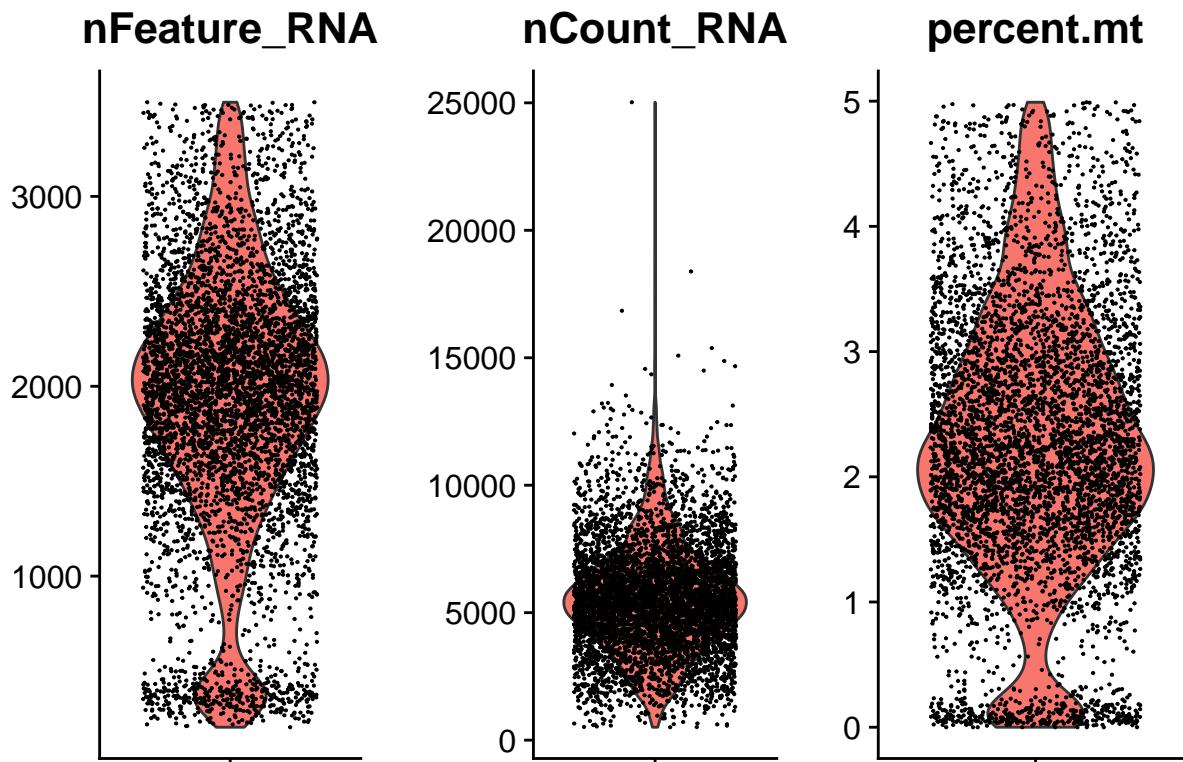
```

GSM8180644_3_4mths QC after filtering



```
## Saving 6.5 x 4.5 in image
```

GSM8180645_4_4mths QC after filtering



```

## Saving 6.5 x 4.5 in image
Integrate data
features <- SelectIntegrationFeatures(object.list = pbmc.list,
                                         nfeatures = 3000)
pbmc.list <- PrepSCTIntegration(object.list = pbmc.list,
                                   anchor.features = features)
anchors <- FindIntegrationAnchors(object.list = pbmc.list,
                                    normalization.method = "SCT",
                                    anchor.features = features)

## Finding all pairwise anchors
## Running CCA
## Merging objects
## Finding neighborhoods
## Finding anchors
## Found 13539 anchors
## Filtering anchors
## Retained 11936 anchors
combo <- IntegrateData(anchorset = anchors,
                        normalization.method = "SCT")

## [1] 1
## [1] 2

```

```
## Merging dataset 2 into 1
## Extracting anchors for merged samples
## Finding integration vectors
## Finding integration vector weights
## Integrating data
# Export data, to be used by future scripts
saveRDS(combo, "./data/covid/initial_combo_4mths.rds")
```