

COVID-19paed_PBMCanalyses_RL003

Rik G.H. Lindeboom

10/11/2021

Code to process and analyse the PBMC data in Yosida et al. Nature, 2021. by Rik Lindeboom. Please reach out if anything is unclear, missing or wrong. Some meta data processing of patient information is omitted to comply with ethics.

```
set.seed(1)
library(Seurat)
library(tidyverse)
library(ggplot2)
library(harmony)
library(ComplexHeatmap)
library(sceasy)
library(reticulate)
library(SoupX)
library(data.table)
library(DoubletFinder)
library(cardelino)
library(reticulate)
source("/mnt/scripts/RL003_function_collection_GitHub.R")
loompy <- import("loompy")
library(randomcoloR)
library(circlize)
library(readr)
library(patchwork)
library(cowplot)
library(lme4)
library(Matrix)
library(numDeriv)

manifest <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/CV001_KM_COVID Sample tracking table - Mani-
  stringsAsFactors = F, header = T, sep = "\t")
sampleTable <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/Pooled_pbmc_CITEseq_summary_kw_210422_pr-
  stringsAsFactors = F, header = T)
manis <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/RL003_manifest.txt",
  stringsAsFactors = F, header = T, sep = "\t")
manis$numberOfDonors <- sapply(gsub("^(.)*", "\\\1", manis$sample_id),
  function(x) sum(sampleTable$pool_group == x))
manis$sample_name <- manis$sample_id
manis$sample_id <- paste(manis$GEX, manis$CITE, sep = "-")
manis$donors <- as.character(sapply(gsub("^(.)*", "\\\1", manis$sample_name),
  function(x) unique(sampleTable$Sample.name[sampleTable$pool_group ==
    x])))
```

```

manis$location_multiplexed_bam <- paste0("/archive/HCA/10X/",
  manis$sample_id, "/outs")
manis$irods_or_farm <- "irods"
manis$bam_file <- "possorted_genome_bam.bam"
manis$barcodesLoc <- paste0("/archive/HCA/10X/", manis$sample_id,
  "/outs/filtered_feature_bc_matrix")
manis$barcode_file <- "barcodes.tsv.gz"
manis$baiFilePresent <- T
manis$bamReady <- NA
for (i in 1:nrow(manis)) {
  foo <- tryCatch(system(paste0("ils ", manis$location_multiplexed_bam[i],
    "/", manis$bam_file[i])))
  if (foo == 0) {
    manis$bamReady[i] <- T
  } else {
    manis$bamReady[i] <- F
  }
}
manis$alignment <- "cellranger"
manis$citeFile <- paste0(manis$GEX, "-", manis$CITE)

socTable <- manis[, c("sample_name", "sample_id", "donors", "numberOfDonors",
  "location_multiplexed_bam", "irods_or_farm", "bam_file",
  "barcodesLoc", "barcode_file", "baiFilePresent")]
write.table(socTable, file = "/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/sampleTable.txt",
  col.names = T, row.names = F, quote = F, sep = ",")

```

'socTable' is used as input for souporcell:

```

# sample_name,sample_id,donors,numberOfDonors,location_multiplexed_bam,irods_or_farm,bam_file,barcodesLoc
# K1-PBMC,CV001_KM9465380-CV001_KM9465395,K1-PBMC;K2-PBMC,5,/archive/HCA/10X/CV001_KM9465380-CV001_KM9465395

dos2unix ${multiplexedSampleTable}
while read -r samplePair; do
sample_id='echo ${samplePair} | cut -f2 -d\,'
selected_k='echo ${samplePair} | cut -f4 -d\,'
location_multiplexed_bam='echo ${samplePair} | cut -f5 -d\,'
irods_or_farm='echo ${samplePair} | cut -f6 -d\,'
bam_file='echo ${samplePair} | cut -f7 -d\,'
barcodesLoc='echo ${samplePair} | cut -f8 -d\,'
barcode_file='echo ${samplePair} | cut -f9 -d\,'
baiFilePresent='echo ${samplePair} | cut -f10 -d\,'

if ! [[ ${sample_id} == 'sample_id' ]]; then
#sample_id=${sample_id}_extraGenotype
cd ${outDir};
mkdir ${sample_id};
cd ${sample_id};
if [[ ${irods_or_farm} == 'irods' ]]; then
iget -Kr ${barcodesLoc}/${barcode_file};
iget -Kr ${location_multiplexed_bam}/${bam_file};
fi;
if [[ $baiFilePresent == FALSE ]]; then

```

```

samtools index -@ ${maxThreads} ${bam_file};
else
iget -Kr ${location_multiplexed_bam}/${bam_file}.bai;
fi;
cd ..;
singularity exec -B $PWD /home/ubuntu/bin/souporcell_latest.sif souporcell_pipeline.py -i ${sample_id}/;
rm ${sample_id}/${bam_file}; rm ${sample_id}/${bam_file}.bai; rm ${sample_id}/${barcode_file};
fi;
done < ${multiplexedSampleTable};

```

Downloading and processing the data from our internal storage systems to create a rds containing the GEX and ADT data using Seurat

```

for (i in 1:nrow(manis)) {
  downloadScData(cite = manis$citeFile[i], bcr = manis$BCR[i],
    tcr = manis$TCR[i], overwrite = F, alignment = manis$alignment[i],
    out_dir = "/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/")
}

for (i in 1:nrow(manis)) {
  currentSample <- try(processCiteSamples(sample = manis$citeFile[i],
    SoupX_rna = T, SoupX_adt = T, save_raw = F, doSct = F,
    data_dir = "/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cite",
    min_cells = 0, min_features = 200))
  write_rds(currentSample, file = paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/",
    manis$citeFile[i], ".rds"), compress = "gz")
}

sampleList <- list()
for (i in 1:nrow(manis)) {
  currentSample <- read_rds(paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/",
    manis$citeFile[i], ".rds"))
  currentSample <- RenameCells(currentSample, add.cell.id = manis$citeFile[i])
  sampleList[[manis$citeFile[i]]] <- currentSample
  rm(currentSample)
}
cov <- merge(sampleList[[1]], y = sampleList[2:length(sampleList)],
  merge.data = TRUE, project = "covidPbmcs_oldNew")
cv <- multiModal_processing(object = cov, gex = T, adt = T, sct = T,
  gexAdtWnn = F, sctAdtWnn = F, doHarmony = T, npca = 30, regress_cellcycle_gex = F,
  makeFinalWnnUmap = F, doFreshSct = T)

cv@meta.data$dataset <- ifelse(is.na(cv@meta.data$patient_id),
  "revision", "original")
cv@meta.data[, c("GEX", "CITE", "BCR", "TCR", "pool_name", "pool_patients")] <- NA
for (i in unique(cv@meta.data$orig.ident)) {
  sample_name <- manifest$sample_id[manifest$Sanger.Sample.ID ==
    gsub("(.*?)-.*", "\\\1", i)]
  cv@meta.data$pool_name[cv@meta.data$orig.ident == i] <- sample_name
  cv@meta.data$GEX[cv@meta.data$orig.ident == i] <- manifest$Sanger.Sample.ID[manifest$sample_id ==
    sample_name & manifest$modality == "GEX"]
  cv@meta.data$CITE[cv@meta.data$orig.ident == i] <- manifest$Sanger.Sample.ID[manifest$sample_id ==
    sample_name & manifest$modality == "CITE"]
}

```

```

try(cv@meta.data$BCR[cv@meta.data$orig.ident == i] <- manifest$Sanger.Sample.ID[manifest$sample_id ==
  sample_name & manifest$modality == "BCR"])
try(cv@meta.data$TCR[cv@meta.data$orig.ident == i] <- manifest$Sanger.Sample.ID[manifest$sample_id ==
  sample_name & manifest$modality == "TCR"])
}
for (i in unique(sampleTable$Sample.name)) {
  samples <- unlist(str_split(i, ";"))
  patientIds <- paste(sampleTable$Individual.Samples.ID[sampleTable$Sample.name ==
    i], collapse = ";")
  cv@meta.data$pool_patients[cv@meta.data$pool_name %in% samples] <- patientIds
}
for (i in unique(cv@meta.data$orig.ident[cv@meta.data$dataset ==
  "original"]))) {
  cv@meta.data$pool_patients[cv@meta.data$orig.ident == i] <- paste(unique(cv@meta.data$patient_id[cv@meta.
    data$orig.ident == i]), collapse = ";")
}

```

We use souporcell to demultiplex the pooled sequencing libraries based on their genotypes (see bash code above).

Manual inspection and iteration revealed that some failed to detect all genotypes because noise from one genotype is clustered into two, this is fixed by adding once more cluster for some samples where no match is found,

souporcell genotype doublets are used to ‘train’ DoubletFinder to find more doublets

```

for (i in unique(cv@meta.data$orig.ident)) {
  soc_out <- read.csv(paste0("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/",
    i, "/clusters.tsv"), header = T, stringsAsFactor = F,
    sep = "\t")
  soc_out$matched_barcode <- paste(i, soc_out$barcode, sep = "_")
  soc_out <- soc_out[soc_out$matched_barcode %in% rownames(cv@meta.data),
    ]
  cv@meta.data[soc_out$matched_barcode, colnames(soc_out)[colnames(soc_out) != "matched_barcode"]] <- soc_out[, colnames(soc_out)[colnames(soc_out) != "matched_barcode"]]
}

# Some failed to detect all genotypes because noise from
# one genotype is clustered into two, this is fixed by
# adding once more cluster for some samples where no match
# is found:
for (i in unique(cv@meta.data$orig.ident)) {
  if (file.exists(paste0("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/",
    i, "_extraGenotype/clusters.tsv"))) {
    soc_out <- read.csv(paste0("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/",
      i, "_extraGenotype/clusters.tsv"), header = T, stringsAsFactor = F,
      sep = "\t")
  } else {
    soc_out <- read.csv(paste0("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/",
      i, "/clusters.tsv"), header = T, stringsAsFactor = F,
      sep = "\t")
  }
  soc_out$matched_barcode <- paste(i, soc_out$barcode, sep = "_")
}

```

```

soc_out <- soc_out[soc_out$matched_barcode %in% rownames(cv@meta.data),
  ]
cv@meta.data[soc_out$matched_barcode, colnames(soc_out)[colnames(soc_out) != "matched_barcode"]] <- soc_out[, colnames(soc_out)[colnames(soc_out) != "matched_barcode"]]
}

cv@meta.data$df_classification_onSinglets <- NA
for (i in unique(cv@meta.data$orig.ident)) {
  cv_subset <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$orig.ident == i])
  cv_subset <- runDoubletFinderOnSouporcellOutput(object = cv_subset)
  cv@meta.data[rownames(cv_subset@meta.data), c("pANN", "DF.classifications",
    "doubletFinder_params", "df_classification_onSinglets")] <- cv_subset@meta.data[, c("pANN", "DF.classifications", "doubletFinder_params",
    "df_classification_onSinglets")]
  rm(cv_subset)
}

```

We've used the nasal GEX libraries (which were not multiplexed) to generate patient genotypes, which we match to souporcell clusters to assign patient ids to souporcell clusters. This required some minor manual adjustments for mismatches using the automated assignment.

```

ref_gt <- load_GT_vcf("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/newNasals.dedup.280421
  na.rm = F)
allCors <- as.data.frame(ref_gt$GT[0, ], stringsAsFactors = F)
cv@meta.data$matched_NB_sample <- NA
cv@meta.data$matched_NB_sample_overlap <- NA
for (i in unique(cv@meta.data$orig.ident)) {
  if (file.exists(paste0("/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision/",
    i, "_extraGenotype/clusters.tsv"))) {
    myCors <- compareGenotype(ref_gt = ref_gt, souporcell_output_dir = "/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision",
      sample_id = paste0(i, "_extraGenotype"))
  } else {
    myCors <- compareGenotype(ref_gt = ref_gt, souporcell_output_dir = "/mnt/projects/RL003_allCitePbmcsTheta/souporcell_revision",
      sample_id = i)
  }
  specificCors <- myCors[, sampleTable$sangerId_matchedSample[!is.na(sampleTable$sangerId_matchedSample),
    sampleTable$pool_group %in% gsub("^( [A-Z]*?) [0-9].*", "\\\1", cv@meta.data$pool_name[cv@meta.data$orig.ident == i])]]
  if (ncol(specificCors) == 0) {
    print("no matching genotypes present in ref")
  } else {
    specificCors <- specificCors[unique(cv@meta.data$assignment[cv@meta.data$status == "singlet" & cv@meta.data$orig.ident == i]), ]
    for (myCluster in rownames(specificCors)[order(decreasing = T,
      apply(specificCors, 1, max))]) {
      mostLikelyPatient <- names(specificCors[myCluster,
        ][order(specificCors[myCluster, ], decreasing = T)][1]
      if (mostLikelyPatient %in% cv@meta.data$matched_NB_sample[cv@meta.data$orig.ident == i & !is.na(cv@meta.data$matched_NB_sample)]) {

```

```

        cv@meta.data$matched_NB_sample[cv@meta.data$orig.ident ==
          i & cv@meta.data$assignment == myCluster] <- paste0(mostLikelyPatient,
          "_fail")
        rownames(myCors)[rownames(myCors) == myCluster] <- paste0(mostLikelyPatient,
          "_fail;cluster", myCluster)
    } else {
        cv@meta.data$matched_NB_sample[cv@meta.data$orig.ident ==
          i & cv@meta.data$assignment == myCluster] <- mostLikelyPatient
        rownames(myCors)[rownames(myCors) == myCluster] <- paste0(mostLikelyPatient,
          ";cluster", myCluster)
    }
    cv@meta.data$matched_NB_sample_overlap[cv@meta.data$orig.ident ==
      i & cv@meta.data$assignment == myCluster] <- as.numeric(specificCors[myCluster,
      ] [order(specificCors[myCluster, ], decreasing = T)][1])
}
rownames(myCors) <- paste0(unique(cv@meta.data$pool_name[cv@meta.data$orig.ident ==
  i]), ";", rownames(myCors))
if (any(colnames(myCors) != colnames(allCors))) {
  halt
}
allCors <- rbind(allCors, myCors)
}

allCors <- allCors[order(rownames(allCors), decreasing = F),
]
columnOrder <- c()
for (i in unique(gsub("[A-Z]*?)[0-9]*-.*", "\\\1", rownames(allCors)))) {
  columnOrder <- c(columnOrder, which(colnames(allCors) %in%
    sampleTable$sangerId_matchedSample[sampleTable$pool_group ==
      i & !is.na(sampleTable$sangerId_matchedSample)]))
}
allCors <- allCors[, c(columnOrder, (1:ncol(allCors))[!(1:ncol(allCors)) %in%
  columnOrder])]

# allCors_wNormalGenotypes <- allCors
Heatmap(allCors, cluster_rows = F, cluster_columns = F, row_names_gp = gpar(cex = 0.55))
# Heatmap(allCors_wNormalGenotypes, cluster_rows =
# F, cluster_columns = F, row_names_gp = gpar(cex=.55))

for (i in unique(cv@meta.data$matched_NB_sample[!is.na(cv@meta.data$matched_NB_sample) &
  !grepl("fail", cv@meta.data$matched_NB_sample)])) {
  cv@meta.data$patient_id[!is.na(cv@meta.data$matched_NB_sample) &
    cv@meta.data$matched_NB_sample == i] <- unique(sampleTable$Individual.Samples.ID[sampleTable$sangerId_matchedSample[sampleTable$sangerId_matchedSample == i & !is.na(sampleTable$sangerId_matchedSample)]])
  cv@meta.data$state[!is.na(cv@meta.data$matched_NB_sample) &
    cv@meta.data$matched_NB_sample == i] <- unique(sampleTable$state[sampleTable$sangerId_matchedSample[sampleTable$sangerId_matchedSample == i & !is.na(sampleTable$sangerId_matchedSample)]])
}

# Fix samples where no nasal data is available by assigning
# the only missing genotype
cv@meta.data$patient_id[cv@meta.data$pool_name == "H2-PBMC" &

```

```

cv@meta.data$matched_NB_sample == "CV001_KM9166445_fail" &
!is.na(cv@meta.data$matched_NB_sample)] <- "PC7"
cv@meta.data$patient_id[cv@meta.data$pool_name == "K1-PBMC" &
cv@meta.data$matched_NB_sample == "CV001_KM9465377_fail" &
!is.na(cv@meta.data$matched_NB_sample)] <- "PP11"
cv@meta.data$patient_id[cv@meta.data$pool_name == "K2-PBMC" &
cv@meta.data$matched_NB_sample == "CV001_KM9465377_fail" &
!is.na(cv@meta.data$matched_NB_sample)] <- "PP11"
cv@meta.data$patient_id[cv@meta.data$pool_name %in% c("Q3-PBMC",
"Q4-PBMC") & cv@meta.data$matched_NB_sample == "CV001_KM9166355_fail" &
!is.na(cv@meta.data$matched_NB_sample)] <- "not_ready"
cv@meta.data$patient_id[cv@meta.data$pool_name %in% c("01-PBMC",
"02-PBMC", "P1-PBMC", "P2-PBMC")] <- "not_ready"
cv@meta.data$state[cv@meta.data$pool_name %in% c("01-PBMC", "02-PBMC",
"P1-PBMC", "P2-PBMC")] <- "Post-COVID"

```

```

cv[["percent.mt"]] <- PercentageFeatureSet(cv, pattern = "^MT-")
cv <- subset(cv, percent.mt < 10)
cv <- subset(cv, cells = rownames(cv@meta.data[(cv@meta.data$df_classification_onSinglets ==
"singlet" & cv@meta.data$status == "singlet"), ]))
cv <- subset(cv, cells = rownames(cv@meta.data[!grepl("CV001_KM9465",
cv@meta.data$orig.ident), ])) # Remove bad samples
cv <- FindNeighbors(cv, dims = 1:30, reduction = "harmony_RNA",
graph.name = "rna_snn")
cv <- FindClusters(cv, graph.name = "rna_snn", resolution = c(0.5,
4, 32), algorithm = 4, method = "igraph", verbose = FALSE)
cv <- FindNeighbors(cv, dims = 1:30, reduction = "harmony_ADT",
graph.name = "adt_snn")
cv <- FindClusters(cv, graph.name = "adt_snn", resolution = c(0.5,
4, 32), algorithm = 4, method = "igraph", verbose = FALSE)
cv <- FindClusters(cv, graph.name = "wsnn_rnaAdt", resolution = c(0.5,
4, 32), algorithm = 4, method = "igraph", verbose = FALSE)
write_rds(cv, "/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem.rda",
compress = "gz")

```

We annotate Leiden clusters using cell type markers by subsetting large cell type compartments and reclusing to increase the resolution

```

cv@meta.data$subset <- NA
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("1",
"6", "7", "8")] <- "T"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("2",
"5", "10")] <- "TNK"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("11")] <- "cycling"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("3",
"9")] <- "MonoDCs"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("12")] <- "platelets"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("4")] <- "B"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("14")] <- "HSPC"
cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in% c("13",
"15")] <- "separateCelltypes"

resolutions <- c(0.5, 4)

```

```

for (i in unique(cv@meta.data$subset)) {
  if (!file.exists(paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered",
    i, ".rds"))) {
    try(rm(cv_subset))
    gc()
    cv <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_ba
  cv@meta.data$subset <- NA
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("1", "6", "7", "8")] <- "T"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("2", "5", "10")] <- "TNK"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("11")] <- "cycling"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("3", "9")] <- "MonoDCs"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("12")] <- "platelets"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("4")] <- "B"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("14")] <- "HSPC"
  cv@meta.data$subset[cv@meta.data$rna_snn_res.0.5 %in%
    c("13", "15")] <- "separateCelltypes"
  cv_subset <- subset(cv, cells = rownames(cv@meta.data[cv@meta.data$subset ==
    i, ]))
  rm(cv)
  gc()
  cv_subset <- multiModal_processing(object = cv_subset,
    gex = T, adt = T, sct = T, gexAdtWnn = T, sctAdtWnn = T,
    doHarmony = T, npca = 30, regress_cellcycle_gex = F,
    makeFinalWnnUmap = T, doFreshSct = T)
  cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_RNA",
    graph.name = "rna_snn")
  cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_ADT",
    graph.name = "adt_snn")
  cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_SCT",
    graph.name = "sct_snn")
  cv_subset <- FindClusters(cv_subset, graph.name = "rna_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
  cv_subset <- FindClusters(cv_subset, graph.name = "adt_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
  cv_subset <- FindClusters(cv_subset, graph.name = "sct_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
  cv_subset <- FindClusters(cv_subset, graph.name = "wsnn_rnaAdt",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
  cv_subset <- FindClusters(cv_subset, graph.name = "wsnn_sctAdt",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
  cv_subset <- combineSmallWnnClusters(object = cv_subset,

```

```

    resolutions = resolutions, graphNames = c("wsnn_rnaAdt",
                                              "wsnn_sctAdt"), minClusterSize = 100)

  write_rds(cv_subset, file = paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldN
  i, ".rds"), compress = "gz")
}

}

```

In the chunks below we go over each compartment that is subsetted above to annotate cell types manually.

```

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad

FeaturePlot(cv_subset, reduction = "umapAfterHarmony_SCT", features = c("pANN",
  "log_prob_doublet", "log_prob_singleton"))

gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
  "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
  "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
  "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
  "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
  "MS4A1", "TNFRSF13B", "ENTPD1", "KIT", "CD34", "PPBP", "PF4",
  "HBB")

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = T, label = T, group.by = "dataset")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = T, label = T, group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()

FeaturePlot(cv_subset, dims = c(1, 2), reduction = "umapAfterHarmony_RNA",
  features = c("HBB", "IL3RA", "CLEC4C", "MKI67"))
foo <- FetchData(cv_subset, c("HBB", "IL3RA", "CLEC4C", "CLEC10A"))
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(HBB +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(IL3RA +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(CLEC4C +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(CLEC10A +
  1))) + geom_point() + theme_classic()
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$HBB)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$IL3RA)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$CLEC4C)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$CLEC10A)

Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust7Markers <- FindMarkers(object = cv_subset, ident.1 = "7",
  assay = "RNA", ident.2 = "1", logfc.threshold = 0.5, min.pct = 0.25,
  only.pos = T) # Seems to be so called AS-DC, markers are AXL SIGLEC6

cv_subset@meta.data$cell_annotation_revision <- NA

```

```

cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("2")] <- "Red Blood Cells"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("1", "3", "4", "5")] <- "pDC"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("7")] <- "AS-DC"
cv_subset@meta.data$cell_annot_revision[is.na(cv_subset@meta.data$cell_annot_revision)] <- "Doublets"

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "cell_annot_revision")
DotPlot(cv_subset, features = gexOnlyList, group.by = "cell_annot_revision",
  cluster.idents = T) + RotatedAxis()

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "cell_annot_revision",
  cols = randomcoloR::distinctColorPalette(length(unique(cv_subset_nk@meta.data$cell_annot_revision))-
  2))
# write_rds(cv_subset@meta.data,file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_separa

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad
DefaultAssay(object = cv_subset) <- "RNA"

DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
  group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
  group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
  group.by = "sct_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
  group.by = "sct_snn_res.0.5")

gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
  "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
  "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
  "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
  "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
  "MS4A1", "TNFRSF13B", "ENTPD1", "KIT", "CD34", "PPBP", "PF4",
  "HBB")

DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
FeaturePlot(cv_subset, dims = c(1, 2), reduction = "umapAfterHarmony_RNA",
  features = c("HBB", "IL3RA", "CLEC4C", "MKI67"))
foo <- FetchData(cv_subset, c("TPSAB1", "CPA3", "TPSB2", "C1QC",
  "EPX", "PRG2"))
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(TPSAB1 +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(CPA3 +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(TPSB2 +
  1))) + geom_point() + theme_classic()

```

```

ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(C1QC +
  1))) + geom_point() + theme_classic()
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$TPSAB1)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$CPA3)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$TPSB2)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$C1QC)

Idents(cv_subset) <- paste(cv_subset@meta.data$rna_snn_res.0.5)
clust10Markers <- FindMarkers(object = cv_subset, ident.1 = "10",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust1Markers <- FindMarkers(object = cv_subset, ident.1 = "1",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("TPSAB1", "CPA3", "KIT", "PRSS57",
  "HPGDS", "GATA2", "TNFSF10", "TRIM63", "IGHA1", "IGHA2",
  "FCER1A"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
  RotatedAxis()
DotPlot(cv_subset, features = c("SEPP1", "AMICA1", "GNLY", "KIAA1598",
  "IL8", "FTL", "ALAS2", "PTPLAD2", "MS4A7", "APOE"), group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("SPINK2", "MTND1P23", "AL450405.1",
  "BEX3", "HBG2", "GUCY1A1", "AL513365.1", "SMIM24", "LAPTM4B",
  "CRHBP"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
  RotatedAxis()
DotPlot(cv_subset, features = c("HDC", "MS4A2", "PRG2", "MS4A3",
  "TPSAB1", "TPSB2", "EPX", "CLC"), cluster.idents = T) + RotatedAxis()
# DotPlot(cv, features =
# c('HDC', 'MS4A2', 'PRG2', 'MS4A3', 'TPSAB1'), group.by =
# 'rna_snn_res.32', cluster.idents = T) + RotatedAxis()

cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision <- "HPSCs"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("11")] <- "HPSCs IFN induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("10")] <- "Mast & Eosinophils"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("8")] <- "Doublets"

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "cell_annot_revision",
  cols = randomcolorR::distinctColorPalette(length(unique(cv_subset@meta.data$cell_annot_revision))-
  2))
# write_rds(cv_subset@meta.data, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_HSPC_a

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad
DefaultAssay(object = cv_subset) <- "RNA"
DimPlot(cv_subset, reduction = "wnn.umap_sctAdt", shuffle = T,
  raster = F, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "wnn.umap_rnaAdt", shuffle = T,
  raster = F, group.by = "projected_annot_rik")

```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_ADT", shuffle = T,
        raster = F, group.by = "projected_annot_rik")

DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "Immune_All_High")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "sct_snn_res.0.5")

DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "sct_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "sct_snn_res.0.5")

gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
                 "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
                 "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
                 "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
                 "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
                 "MS4A1", "TNFRSF13B", "ENTPD1", "adt_AB-KIT", "adt_AB-CD34",
                 "KIT", "CD34", "PPBP", "PF4", "HBB")
DotPlot(cv_subset, features = gexOnlyList, group.by = "sct_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "dataset")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Immune_All_High")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")

Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clustMarkers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                 logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clustMarkers[order(clustMarkers$pct.2 - clustMarkers$pct.1),
            ]

FeaturePlot(cv_subset, dims = c(1, 2), reduction = "umapAfterHarmony_RNA",
            features = c("HBB", "IL3RA", "CLEC4C", "MKI67"))
foo <- FetchData(cv_subset, c("TPSAB1", "CPA3", "TPSB2", "C1QC",
                             "EPX", "PRG2"))

```

```

foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(TPSAB1 +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(CPA3 +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(TPSB2 +
  1))) + geom_point() + theme_classic()
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(C1QC +
  1))) + geom_point() + theme_classic()
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$TPSAB1)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$CPA3)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$TPSB2)
dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo$C1QC)

Idents(cv_subset) <- paste(cv_subset@meta.data$rna_snn_res.0.5)
# Idents(cv_subset) <-
# paste(cv_subset@meta.data$sct_snn_res.0.5)
clust10Markers <- FindMarkers(object = cv_subset, ident.1 = "10",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust1Markers <- FindMarkers(object = cv_subset, ident.1 = "1",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust6Markers <- FindMarkers(object = cv_subset, ident.1 = "6",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust4Markers <- FindMarkers(object = cv_subset, ident.1 = "4",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("TPSAB1", "CPA3", "KIT", "PRSS57",
  "HPGDS", "GATA2", "TNFSF10", "TRIM63", "IGHA1", "IGHA2",
  "FCER1A"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
  RotatedAxis()
DotPlot(cv_subset, features = c("SEPP1", "AMICA1", "GNLY", "KIAA1598",
  "IL8", "FTL", "ALAS2", "PTPLAD2", "MS4A7", "APOE"), group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("SPINK2", "MTND1P23", "AL450405.1",
  "BEX3", "HBG2", "GUCY1A1", "AL513365.1", "SMIM24", "LAPTM4B",
  "CRHBP"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
  RotatedAxis()
DotPlot(cv_subset, features = c("HDC", "MS4A2", "PRG2", "MS4A3",
  "TPSAB1", "TPSB2", "EPX", "CLC"), cluster.idents = T) + RotatedAxis()
# DotPlot(cv, features =
# c('HDC', 'MS4A2', 'PRG2', 'MS4A3', 'TPSAB1'), group.by =
# 'rna_snn_res.32', cluster.idents = T) + RotatedAxis()

cv_subset@meta.data$cell_annotation_revision <- NA
cv_subset@meta.data$cell_annotation_revision <- "Cycling"
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("7", "12", "13")] <- "Doublets"
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("9", "11")] <- "Plasma cells"
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("8")] <- "Plasmablasts"

```

```

DotPlot(cv_subset, features = gexOnlyList, group.by = "cell_annot_revision",
        cluster.idents = T) + RotatedAxis()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision",
        cols = randomcoloR::distinctColorPalette(length(unique(cv_subset@meta.data$cell_annot_revision)) +
        2))
# write_rds(cv_subset@meta.data,file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_cyclin

# Subset plasmablasts to separate isotypes
resolutions <- c(0.5, 4)
cv_subset <- subset(cv_subset, cells = rownames(cv_subset@meta.data[cv_subset@meta.data$cell_annot_revision,
  c("Plasma cells", "Plasmablasts"), ]))
cv_subset@meta.data$type_sample <- gsub("(..) .*", "\\\1", cv_subset@meta.data$patient_id)
gc()
cv_subset <- multiModal_processing(object = cv_subset, gex = T,
  adt = T, sct = T, gexAdtWnn = T, sctAdtWnn = T, doHarmony = T,
  npca = 30, regress_cellcycle_gex = F, makeFinalWnnUmap = T,
  doFreshSct = T)
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_RNA",
  graph.name = "rna_snn")
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_ADT",
  graph.name = "adt_snn")
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_SCT",
  graph.name = "sct_snn")
cv_subset <- FindClusters(cv_subset, graph.name = "rna_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "adt_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "sct_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")

cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("6")] <- "Doublets"

# write_rds(cv_subset@meta.data,file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_plasma

cv_subset <- subset(cv_subset, cells = rownames(cv_subset@meta.data[cv_subset@meta.data$rna_snn_res.0.5 +
  "6", ])) # These are doublets
cv_subset <- multiModal_processing(object = cv_subset, gex = T,
  adt = T, sct = T, gexAdtWnn = T, sctAdtWnn = T, doHarmony = T,
  npca = 30, regress_cellcycle_gex = F, makeFinalWnnUmap = T,

```

```

doFreshSct = T)
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_RNA",
  graph.name = "rna_snn")
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_ADT",
  graph.name = "adt_snn")
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_SCT",
  graph.name = "sct_snn")
cv_subset <- FindClusters(cv_subset, graph.name = "rna_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "adt_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "sct_snn",
  resolution = resolutions, algorithm = 4, method = "igraph",
  verbose = FALSE)

# write_rds(cv_subset,file='farm/cov_oldNewMerged_filtered_badAdtRem_subset_plasmas.rds',compress='gz')
DefaultAssay(cv_subset) <- "RNA"
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust05Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
  logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.4
clust4Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
  logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust05Markers <- clust05Markers[order(clust05Markers$pct.2 -
  clust05Markers$pct.1), ]
clust4Markers <- clust4Markers[order(clust4Markers$pct.2 - clust4Markers$pct.1),
  ]

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, group.by = "projected_annot_rik", cols = myColsForCellTypes)
DimPlot(cv_subset, reduction = "umapAfterHarmony_ADT", shuffle = T,
  raster = F, group.by = "projected_annot_rik", cols = myColsForCellTypes)
DimPlot(cv_subset, reduction = "wnn.umap_rnaAdt", shuffle = T,
  raster = F, group.by = "projected_annot_rik", cols = myColsForCellTypes)

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "dataset")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "projected_annot_rik",
  cols = myColsForCellTypes)
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "Immune_All_Low")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "Immune_Blood_Low")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,

```

```

raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcolorR::distinctColorPalette(length(un
        2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
               "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
               "GZMK", "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1",
               "FCGR3A", "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A",
               "CD8A", "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "PPBP",
               "PF4", "KLRB1", "TRDV2", "adt_AB-TRDV2", "TRGV9", "adt_AB-TRGV9",
               "IL7R", "KLRC3", "GNLY", "CD27", "adt_AB-CD27", "SELL", "TIGIT",
               "CXCR4", "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-2",
               "adt_AB-PTPRC-3")
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = gexOnlyList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = gexOnlyList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("adt_AB-PTPRC-2",
               "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
               "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
               "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
               "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
               "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
               "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
               "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
               "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("IFNG",
               "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
               "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
               cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("IFNG",
               "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
               "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
               cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")

# Not really any clustering of isotypes

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad"
DefaultAssay(object = cv_subset) <- "RNA"
DimPlot(cv_subset, reduction = "wnn.umap_sctAdt", shuffle = T,
        raster = F, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "umapAfterHarmony_ATD", shuffle = T,
        raster = F, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "wnn.umap_rnaAdt", shuffle = T,
        raster = F, group.by = "projected_annot_rik")

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "projected_annot_rik")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Immune_All_High")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")

DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "adt_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "adt_snn_res.0.5")

gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
                 "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
                 "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
                 "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
                 "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
                 "MS4A1", "TNFRSF13B", "ENTPD1", "KIT", "CD34", "PPBP", "PF4",
                 "HBB")

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "dataset") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "projected_annot_rik",
        cols = myColsForCellTypes) + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Immune_All_Low") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.32") + NoLegend()

```

```

# markerList <-
# c('CD8A', 'CD8B', 'CCR7', 'GZMB', paste0('adt_AB-', c('PTPRC-2', 'PTPRC-3', 'CD4', 'CD8A')))
markerList <- c("NCR1", "NCAM1", "FCGR3A", "FCER1G", "adt_AB-CD8A")
markerList <- c("NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D")
markerList <- c("KLRB1", "CD3G", "FGFBP2")
markerList <- c("FOXP3", "IL2RA", "CTLA4")
markerList <- c("GNLY", "NKG7", "GZMK")
# markerList <- c('GNLY', 'NKG7', 'CD3D')

# g/d t cells and mait
markerList <- c("SLC4A10", "TRAV1-2", "TRBV6-2", "adt_AB-TRAV7")
markerList <- c("TRDV1", "TRDV2", "TRGV9", "TRDC", "TRGC1", "TRGC2",
  paste0("adt_AB-", c("TRAV24", "TRAV7", "TRBV13", "TRGV9",
    "TRDV2")))
DotPlot(cv_subset, features = markerList, group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5") + NoLegend()
DotPlot(cv_subset, features = markerList, group.by = "rna_snn_res.4",
  cluster.idents = T) + RotatedAxis()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4") + NoLegend()
foo <- FetchData(cv_subset, markerList)
foo$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[grep("^TR.*V[0-9].*", 
  rownames(cv_subset)[["RNA"]])])
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
for (i in markerList) {
  print(ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2,
    col = log2(get(i) + 1))) + geom_point(cex = 0.1) + theme_classic() +
    ggtitle(i))
}
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(rnaTr +
  1))) + geom_point(cex = 0.1) + theme_classic() + ggtitle("TR RNA")

# CTLs
markerList <- c("CD8A", "CD8B", "CCR7", "GZMB", "GZMH", paste0("adt_AB-",
  c("PTPRC-2", "PTPRC-3", "CD4", "CD8A")))
# DotPlot(cv_subset, features = markerList, group.by =
#   'rna_snn_res.0.5', cluster.idents = T) + RotatedAxis()
# DimPlot(cv_subset, reduction = 'umapAfterHarmony_RNA', shuffle
#   = T, raster = F, label = T, group.by = 'rna_snn_res.0.5') +
#   NoLegend()
DotPlot(cv_subset, features = markerList, group.by = "rna_snn_res.4",
  cluster.idents = T) + RotatedAxis()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4") + NoLegend()
foo <- FetchData(cv_subset, c(markerList, "nCount_RNA"))
foo$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[grep("^TR.*V[0-9].*", 
  rownames(cv_subset)[["RNA"]])])
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
for (i in markerList) {
  print(ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2,
    col = log2(get(i) + 1))) + geom_point(cex = 0.1) + theme_classic() +
    ggtitle(i))
}

```

```

        ggttitle(i))
    }
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(rnaTr +
  1))) + geom_point(cex = 0.1) + theme_classic() + ggttitle("TR RNA")
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = nCount_RNA)) +
  geom_point(cex = 0.1) + theme_classic() + ggttitle("nCount")
for (i in markerList) {
  print(dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo[, i]))
  title(i)
}

Idents(cv_subset) <- paste(cv_subset@meta.data$rna_snn_res.0.5)
clust12Markers <- FindMarkers(object = cv_subset, ident.1 = "1",
  ident.2 = "4", assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25,
  only.pos = F)
clust12MarkersAdt <- FindMarkers(object = cv_subset, ident.1 = "12",
  ident.2 = "3", assay = "ADT", logfc.threshold = 0.5, min.pct = 0.25,
  only.pos = F)
clust1Markers <- FindMarkers(object = cv_subset, ident.1 = "1",
  assay = "RNA", logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)

cv_subset@meta.data$cell_annotation_revision <- NA
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("6")] <- "g/d T"
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("1")] <- "CD4 CTL"
cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("8")] <- "MAIT"
# cv_subset@meta.data$cell_annotation_revision[cv_subset@meta.data$rna_snn_res.0.5 %in% c('1')]
# <- 'AS-DC'
# cv_subset@meta.data$cell_annotation_revision[is.na(cv_subset@meta.data$cell_annotation_revision)]
# <- 'Doublets'

cv_subset@meta.data$subsubcluster <- "subT"
cv_subset@meta.data$subsubcluster[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("4")] <- "subTNK"
cv_subset@meta.data$subsubcluster[cv_subset@meta.data$rna_snn_res.0.5 %in%
  c("1", "6")] <- "subNK"

cv <- cv_subset

for (i in unique(cv@meta.data$subsubcluster)) {
  if (!file.exists(paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filter",
    i, ".rds"))) {
    cv_subset <- subset(cv, cells = rownames(cv@meta.data[cv@meta.data$subsubcluster ==
      i, ]))
    gc()
    cv_subset <- multiModal_processing(object = cv_subset,
      gex = T, adt = T, sct = T, gexAdtWnn = T, sctAdtWnn = T,
      doHarmony = T, npca = 30, regress_cellcycle_gex = F,
      makeFinalWnnUmap = T, doFreshSct = T)
    cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_RNA",
      graph.name = "rna_snn")
  }
}

```

```

cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_ADT",
    graph.name = "adt_snn")
cv_subset <- FindNeighbors(cv_subset, dims = 1:30, reduction = "harmony_SCT",
    graph.name = "sct_snn")
cv_subset <- FindClusters(cv_subset, graph.name = "rna_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "adt_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
cv_subset <- FindClusters(cv_subset, graph.name = "sct_snn",
    resolution = resolutions, algorithm = 4, method = "igraph",
    verbose = FALSE)
# cv_subset <- FindClusters(cv_subset, graph.name =
#   'wsnn_rnaAdt', resolution = resolutions, algorithm
#   = 4, method = 'igraph', verbose = FALSE)
# cv_subset <- FindClusters(cv_subset, graph.name =
#   'wsnn_sctAdt', resolution = resolutions, algorithm
#   = 4, method = 'igraph', verbose = FALSE)

# cv_subset <-
# combineSmallWnnClusters(object=cv_subset, resolutions=resolutions, graphNames=c('wsnn_rnaAdt', 'wsnn_sctAdt'))

write_rds(cv_subset, file = paste0("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad",
    i, ".rds"), compress = "gz")
}

}

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad.rds")
cv_subset@meta.data$type_sample <- gsub("(..)\\.\\*", "\\\\1", cv_subset@meta.data$patient_id)
cv_subset@meta.data$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset[["RNA"]])[grep("^\^TR.*V[0-9]", rownames(cv_subset[["RNA"]]))]]))

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "sct_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "type_sample")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "Phase")

# markerList <-
# c('NCAM1', 'adt_AB-NCAM1', 'CD3D', 'adt_AB-CD3D') foo <-
# FetchData(cv_subset, c(markerList, 'nCount_RNA')) foo$rnaTr
# <-

```

```

# rowSums(FetchData(cv_subset,rownames(cv_subset)[['RNA']] )[grep('^TR.*V[0-9].*',rownames(cv_subset)[['RNA']] )])
# foo[,c('UMAP_1','UMAP_2')] <-
# as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
# for (i in markerList)
# {print(ggplot(foo[sample(1:nrow(foo)),],aes(UMAP_1,UMAP_2,col=log2(get(i)+1)))
# + geom_point(cex=.1) + theme_classic() + ggtitle(i))}
# ggplot(foo[sample(1:nrow(foo)),],aes(UMAP_1,UMAP_2,col=log2(rnaTr+1)))
# + geom_point(cex=.1) + theme_classic() + ggtitle('TR
# RNA')
# ggplot(foo[sample(1:nrow(foo)),],aes(UMAP_1,UMAP_2,col=nCount_RNA))
# + geom_point(cex=.1) + theme_classic() +
# ggtitle('nCount') for (i in markerList)
# {print(dplot(x=foo$UMAP_1,y=foo$UMAP_2,z=foo[,i]);title(i)}

# Idents(cv_subset) <-
# paste(cv_subset@meta.data$rna_snn_res.32) clust385Markers
# <- FindMarkers(object = cv_subset,ident.1 = '385',assay =
# 'RNA',logfc.threshold = .5,min.pct = .25,only.pos = F)
# Idents(cv_subset) <-
# paste(cv_subset@meta.data$rna_snn_res.0.5) clust2Markers
# <- FindMarkers(object = cv_subset,ident.1 = '2',ident.2 =
# '1',assay = 'RNA',logfc.threshold = .5,min.pct =
# .25,only.pos = F) # S100A4/6 high FCGR3A high CCL4/5 high
# GZMH/B high Idents(cv_subset) <-
# paste(cv_subset@meta.data$sct_snn_res.4) clust20Markers
# <- FindMarkers(object = cv_subset,ident.1 = '20',assay =
# 'RNA',logfc.threshold = .5,min.pct = .25,only.pos = F) #
# IFN activated clust45Markers <- FindMarkers(object =
# cv_subset,ident.1 = '45',assay = 'RNA',logfc.threshold =
# .5,min.pct = .25,only.pos = F) # cycling clust29Markers
# <- FindMarkers(object = cv_subset,ident.1 = '29',assay =
# 'RNA',logfc.threshold = .5,min.pct = .25,only.pos = F) #
# AP12 specific AP1-complex up clust41Markers <-
# FindMarkers(object = cv_subset,ident.1 = '41',assay =
# 'RNA',logfc.threshold = .5,min.pct = .25,only.pos = F) #
# IL7R expressing clust38Markers <- FindMarkers(object =
# cv_subset,ident.1 = '38',assay = 'RNA',logfc.threshold =
# .5,min.pct = .25,only.pos = F) # HLA-DR expressing
# clust9Markers <- FindMarkers(object = cv_subset,ident.1 =
# '9',assay = 'RNA',logfc.threshold = .5,min.pct =
# .25,only.pos = F) # clust21Markers <- FindMarkers(object
# = cv_subset,ident.1 = '21',assay = 'RNA',logfc.threshold
# = .5,min.pct = .25,only.pos = F) #

DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(unique(
        2)))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(unique(

```

```

    2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "sct_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "sct_snn_res.4")

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
    "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
    "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1", "FCGR3A",
    "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A", "CD8A",
    "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "KLRD1", "adt_AB-KLRD1",
    "TRAV24", "adt_AB-TRAV24")
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
    cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, group.by = "sct_snn_res.0.5", features = markerList,
    cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
    cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, group.by = "sct_snn_res.4", features = markerList,
    cluster.idents = T) + RotatedAxis()

foo <- FetchData(cv_subset, c(markerList, "nCount_RNA"))
foo$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[grep("^TR.*V[0-9].*", rownames(cv_subset)[["RNA"]])])
# foo[,c('UMAP_1', 'UMAP_2')] <-
# as.data.frame(cv_subset@reductions$umapAfterHarmony_SCT@cell.embeddings)
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
for (i in markerList) {
    print(ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2,
        col = log2(get(i) + 1))) + geom_point(cex = 0.1) + theme_classic() +
        ggtitle(i))
}
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(rnaTr +
    1))) + geom_point(cex = 0.1) + theme_classic() + ggtitle("TR RNA")
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = nCount_RNA)) +
    geom_point(cex = 0.1) + theme_classic() + ggtitle("nCount")
for (i in markerList) {
    print(dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo[, i]))
    title(i)
}

cv_subset_nk <- cv_subset
cv_subset_nk@meta.data$cell_annotation_revision <- NA
cv_subset_nk@meta.data$cell_annotation_revision <- "NK FCER1G+"
cv_subset_nk@meta.data$cell_annotation_revision[cv_subset_nk@meta.data$rna_snn_res.0.5 %in%
    c("2")] <- "NK"
cv_subset_nk@meta.data$cell_annotation_revision[cv_subset_nk@meta.data$sct_snn_res.0.5 %in%
    c("4")] <- "NK CD56 bright"
cv_subset_nk@meta.data$cell_annotation_revision[cv_subset_nk@meta.data$sct_snn_res.4 %in%

```

```

  c("33")] <- "NKT"
cv_subset_nk@meta.data$cell_annot_revision[cv_subset_nk@meta.data$rna_snn_res.4 %in%
  c("36")] <- "cycling"
cv_subset_nk@meta.data$cell_annot_revision[cv_subset_nk@meta.data$sct_snn_res.4 %in%
  c("38")] <- "NK HLA-DR+"
cv_subset_nk@meta.data$cell_annot_revision[cv_subset_nk@meta.data$sct_snn_res.4 %in%
  c("20")] <- "NK IFN induced"
cv_subset_nk@meta.data$cell_annot_revision[cv_subset_nk@meta.data$sct_snn_res.4 %in%
  c("41")] <- "ILC"

# write_rds(cv_subset@meta.data, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRemsubset_subNK_'

DimPlot(cv_subset_nk, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision",
        cols = randomcoloR::distinctColorPalette(length(unique(cv_subset_nk@meta.data$cell_annot_revision))-
          2))
DimPlot(cv_subset_nk, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision",
        cols = randomcoloR::distinctColorPalette(length(unique(cv_subset_nk@meta.data$cell_annot_revision))-
          2))

DotPlot(cv_subset_nk, group.by = "cell_annot_revision", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad"
cv_subset@meta.data$type_sample <- gsub("(..).*", "\\\1", cv_subset@meta.data$patient_id)
cv_subset@meta.data$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset[["RNA"]])[grep("^\^TR.*V[0-9]"
rownames(cv_subset[["RNA"]])])))

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "sct_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(un-
          2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,

```

```

raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3G", "CD3D",
               "adt_AB-CD3D", "HLA-DRA", "CCL4", "CCL5", "GZMH", "GZMB",
               "IL32", "IFNG", "IFI6", "IRF7", "IFI44L", "PRF1", "FCGR3A",
               "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A", "CD8A",
               "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "KLRD1", "adt_AB-KLRD1",
               "TRAV24", "adt_AB-TRAV24", "TRDV2", "TRGV9", "IL7R", "SELL",
               "MX1")

DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
# DotPlot(cv_subset, group.by = 'sct_snn_res.0.5', features =
# markerList, cluster.idents = T,) + RotatedAxis() +
# theme(axis.text=element_text(size=7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
# DotPlot(cv_subset, group.by = 'sct_snn_res.4', features =
# markerList, cluster.idents = T) + RotatedAxis() +
# theme(axis.text=element_text(size=7))

DotPlot(subset(cv_subset, cells = rownames(cv_subset@meta.data[!cv_subset@meta.data$rna_snn_res.4 %in%
    c("3", "6", "8", "9", "19", "1", "30", "7", "27", "20", "22",
      "26", "34"), ])), group.by = "rna_snn_res.4", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DimPlot(subset(cv_subset, cells = rownames(cv_subset@meta.data[!cv_subset@meta.data$rna_snn_res.4 %in%
    c("3", "6", "8", "9", "19", "1", "30", "7", "27", "20", "22",
      "26", "34"), ])), reduction = "umapAfterHarmony_RNA",
        shuffle = T, raster = F, label = T, group.by = "rna_snn_res.4")

foo <- FetchData(cv_subset, c(markerList, "nCount_RNA"))
foo$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset[["RNA"]])[grep("^\w+.*V[0-9].*", rownames(cv_subset[["RNA"]]))]))
# foo[,c('UMAP_1', 'UMAP_2')] <-
# as.data.frame(cv_subset@reductions$umapAfterHarmony_SCT@cell.embeddings)
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings)
for (i in markerList) {
  print(ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2,
    col = log2(get(i) + 1))) + geom_point(cex = 0.1) + theme_classic() +
    ggtitle(i))
}
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(rnaTr +
  1))) + geom_point(cex = 0.1) + theme_classic() + ggtitle("TR RNA")
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log10(nCount_RNA))) +
  geom_point(cex = 0.1) + theme_classic() + ggtitle("nCount")
for (i in markerList) {
  print(dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo[, i]))
  title(i)
}

```

```

cv_subset_tnk <- cv_subset

DimPlot(cv_subset_tnk, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, group.by = "type_sample")
Idents(cv_subset_tnk) <- cv_subset_tnk@meta.data$rna_snn_res.4
clustMarkers <- FindAllMarkers(object = cv_subset_tnk, assay = "RNA",
                                logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clustMarkers[order(clustMarkers$pct.2 - clustMarkers$pct.1),
            ]

cv_subset_tnk@meta.data$cell_annot_revision <- NA
cv_subset_tnk@meta.data$cell_annot_revision <- as.character(cv_subset_tnk@meta.data$rna_snn_res.4)
cv_subset_tnk@meta.data$cell_annot_revision <- "T CD8 CTL"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("25", "11", "10", "33")] <- "T CD8 EM"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("13")] <- "NK CD56 bright"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("23")] <- "NK IFN induced"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("32")] <- "T CD8 CTL IFN induced"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("5")] <- "T g/d"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("31")] <- "T CD4 CTL"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("1", "9", "3")] <- "NK FCER1G+"
cv_subset_tnk@meta.data$cell_annot_revision[cv_subset_tnk@meta.data$rna_snn_res.4 %in%
                                             c("6", "8", "19", "30", "7", "27", "20", "22", "34")] <- "NK"

# write_rds(cv_subset@meta.data, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem.subset_subTNK')

DimPlot(cv_subset_tnk, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision",
        cols = randomcoloR::distinctColorPalette(length(unique(cv_subset_tnk@meta.data$cell_annot_revision)) - 2))
DimPlot(cv_subset_tnk, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision",
        cols = randomcoloR::distinctColorPalette(length(unique(cv_subset_tnk@meta.data$cell_annot_revision)) - 2))

DotPlot(cv_subset_tnk, group.by = "cell_annot_revision", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad")
cv_subset@meta.data$type_sample <- gsub("(..)\\.\\*", "\\\1", cv_subset@meta.data$patient_id)
cv_subset@meta.data$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[[grep("^.TR.*V[0-9]", rownames(cv_subset)[["RNA"]])]])
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust05Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)

```

```

Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.4
clust4Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
  logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust05Markers[order(clust05Markers$pct.2 - clust05Markers$pct.1),
  ]
clust4Markers[order(clust4Markers$pct.2 - clust4Markers$pct.1),
  ]

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(unlist(
    2)))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
  "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
  "GZMK", "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1",
  "FCGR3A", "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A",
  "CD8A", "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "PPBP",
  "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9", "adt_AB-TRGV9",
  "IL7R", "KLRC3", "GNLY", "CD27", "adt_AB-CD27", "SELL", "TIGIT",
  "CXCR4", "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-2",
  "adt_AB-PTPRC-3", "SLC4A10", "NCR3", "KLRB1")
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = gexOnlyList,
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = gexOnlyList,
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
  "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
  "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
  "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
  "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

foo <- FetchData(cv_subset, c(markerList, "nCount_RNA"))
foo$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[grep("^TR.*V[0-9].*", rownames(cv_subset)[["RNA"]]]]))
# foo[,c('UMAP_1', 'UMAP_2')] <-

```

```

# as.data.frame(cv_subset@reductions$umapAfterHarmony_SCT@cell.embeddings)
foo[, c("UMAP_1", "UMAP_2")] <- as.data.frame(cv_subset@reductions$umapAfterHarmony_RNA@cell.embeddings
for (i in markerList) {
  print(ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2,
    col = log2(get(i) + 1))) + geom_point(cex = 0.1) + theme_classic() +
    ggtitle(i))
}
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = log2(rnaTr +
  1))) + geom_point(cex = 0.1) + theme_classic() + ggtitle("TR RNA")
ggplot(foo[sample(1:nrow(foo)), ], aes(UMAP_1, UMAP_2, col = nCount_RNA)) +
  geom_point(cex = 0.1) + theme_classic() + ggtitle("nCount")
for (i in markerList) {
  print(dplot(x = foo$UMAP_1, y = foo$UMAP_2, z = foo[, i]))
  title(i)
}
cv_subset_t <- cv_subset

# tem: CCR7lo SELLlo CX3CR1hi CD27lo IL7Rhi CD27- CD45RA-
# '30', '1', tcm: CCR7hi SELLhi CX3CR1lo CD27hi IL7Rhi CD27+
# CD45RA- '35', '36', '17', '7', '15', '25', '19' temra: CCR7-
# IL7Rlo CD27- CD45RA+ '5', '3', '24',

```

```

cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision <- "T CD8 CTL"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("30")] <- "T CD8 em"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("1")] <- "MAIT"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("35", "36", "17", "7", "15", "25", "19", "6")] <- "T CD8 cm"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("5", "3", "24")] <- "T CD8 emra"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("4", "11")] <- "T g/d"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("34", "8", "9")] <- "T CD4 CTL"

cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("33")] <- "Doublets"

# write_rds(cv_subset@meta.data, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAvtRem.subset_subT_a

```

```

DotPlot(cv_subset, group.by = "cell_annot_revision", features = c("adt_AB-PTPRC-2",
  "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
  "CX3CR1", "GZMH", "PRF1", "SLC4A10", "NCR3", "KLRB1"), cluster.idents = T) +
  RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
  "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
  "CX3CR1", "GZMH", "PRF1", "SLC4A10", "NCR3", "KLRB1"), cluster.idents = T) +
  RotatedAxis() + theme(axis.text = element_text(size = 7))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,

```

```

raster = F, label = T, group.by = "cell_annot_revision")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

```

```

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad")
cv_subset@meta.data$type_sample <- gsub("(..)\\.\\*", "\\\\1", cv_subset@meta.data$patient_id)
cv_subset@meta.data$rnaTr <- rowSums(FetchData(cv_subset, rownames(cv_subset)[["RNA"]])[grep("^\u03c4R.*V[0-9]", rownames(cv_subset)[["RNA"]])]))
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust05Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.4
clust4Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.32
clust32Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust05Markers[order(clust05Markers$pct.2 - clust05Markers$pct.1),
               ]
clust4Markers[order(clust4Markers$pct.2 - clust4Markers$pct.1),
               ]
clust32Markers[order(clust32Markers$pct.2 - clust32Markers$pct.1),
               ]

```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcolor::distinctColorPalette(length(unique(type_sample)) + 2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

```

```

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
                "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
                "GZMK", "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1",
                "FCGR3A", "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A",
                "CD8A", "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "PPBP",
                "PF4", "KLRB1", "TRDV2", "adt_AB-TRDV2", "TRGV9", "adt_AB-TRGV9",
                "IL7R", "KLRC3", "GNLY", "CD27", "adt_AB-CD27", "SELL", "TIGIT",
                "CXCR4", "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-2",
                "adt_AB-PTPRC-3")

```

```

DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
```

```

    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = gexOnlyList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = gexOnlyList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("adt_AB-PTPRC-2",
    "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
    "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
    "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
    "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
    "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
    "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
    "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
    "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("IFNG",
    "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
    "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("IFNG",
    "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
    "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

# Fairly shallow
cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("4")] <- "T low quality"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("5")] <- "T regulatory"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("6")] <- "T CD4 IFN induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("1")] <- "T CD4 Naive"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("2")] <- "T CD4 Helper"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
    c("3")] <- "T CD8 Naive"

# write_rds(cv_subset@meta.data, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_T_anno

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
    raster = F, label = T, group.by = "cell_annot_revision")

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad
cv_subset@meta.data$type_sample <- gsub("(..)\\.\\*", "\\\\1", cv_subset@meta.data$patient_id)
Ident(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust05Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
Ident(cv_subset) <- cv_subset@meta.data$rna_snn_res.4

```

```

clust4Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
  logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust05Markers <- clust05Markers[order(clust05Markers$pct.2 -
  clust05Markers$pct.1), ]
clust4Markers <- clust4Markers[order(clust4Markers$pct.2 - clust4Markers$pct.1),
  ]

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(un
  2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")

DotPlot(cv_subset, features = c("CD14", "FCGR3A", "C1QA", "CLEC9A",
  "CLEC10A", "CD1C", "PTPRC", "PPBP", "PF4", "MS4A1", "NEAT1",
  "IFI6", "IRF7", "IFI44L"), group.by = "rna_snn_res.0.5",
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, features = c("CD14", "FCGR3A", "C1QA", "CLEC9A",
  "CLEC10A", "CD1C", "PTPRC", "PPBP", "PF4", "MS4A1", "NEAT1",
  "IFI6", "IRF7", "IFI44L"), group.by = "rna_snn_res.4", cluster.idents = T) +
  RotatedAxis() + theme(axis.text = element_text(size = 7))

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
  "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
  "GZMK", "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1",
  "FCGR3A", "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A",
  "CD8A", "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "PPBP",
  "PF4", "KLRB1", "TRDV2", "adt_AB-TRDV2", "TRGV9", "adt_AB-TRGV9",
  "IL7R", "KLRC3", "GNLY", "CD27", "adt_AB-CD27", "SELL", "TIGIT",
  "CXCR4", "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-2",
  "adt_AB-PTPRC-3")
gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
  "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
  "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
  "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
  "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
  "MS4A1", "TNFRSF13B", "ENTPD1", "KIT", "CD34", "PPBP", "PF4",
  "HBB", "HDC", "MS4A2", "PRG2", "MS4A3", "TPSAB1", "TPSB2",

```

```

    "EPX", "CLC", "IFI6", "IRF7", "IFI44L")
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = gexOnlyList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = gexOnlyList,
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("adt_AB-PTPRC-2",
    "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
    "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
    "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
    "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
    "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
    "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
    "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
    "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("IL6",
    "GPBAR1", "CXCL10", "IFNG", "TBX21", "TNF", "GATA3", "IL4",
    "IL5", "RORC", "IL17A", "IL17F", "IL21", "CCL5", "PHLDA1",
    "LYAR", "ODF2L", "IL7R", "PDE4D"), cluster.idents = T) +
    RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("IL6",
    "GPBAR1", "CXCL10", "IFNG", "TBX21", "TNF", "GATA3", "IL4",
    "IL5", "RORC", "IL17A", "IL17F", "IL21", "CCL5", "PHLDA1",
    "LYAR", "ODF2L", "IL7R", "PDE4D"), cluster.idents = T) +
    RotatedAxis() + theme(axis.text = element_text(size = 7))

cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision <- "Monocyte CD14"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("36", "24", "16", "5")] <- "Monocyte CD14 IFN-induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("39")] <- "Monocyte CD14 IL6+"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("41", "1", "47", "45", "6", "26")] <- "Monocyte CD16"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("32")] <- "Monocyte CD16 IFN-induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("30")] <- "Monocyte CD16+C1"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("48", "22", "43", "38")] <- "doublets"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("49")] <- "cDC1"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
    c("3")] <- "cDC2"

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")
DotPlot(cv_subset, features = c("IL6", "GPBAR1", "CXCL10", "CD14",
                                "FCGR3A", "C1QA", "CLEC9A", "CLEC10A", "CD1C", "PTPRC", "PPBP",
                                "PF4", "MS4A1", "NEAT1", "IFI6", "IRF7", "IFI44L"), group.by = "rna_snn_res.4",
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, features = c("IL6", "GPBAR1", "CXCL10", "CD14",
                                "FCGR3A", "C1QA", "CLEC9A", "CLEC10A", "CD1C", "PTPRC", "PPBP",
                                "PF4", "MS4A1", "NEAT1", "IFI6", "IRF7", "IFI44L"), group.by = "cell_annot_revision",
        cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

# write_rds(cv_subset@meta.data,file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_MonoDC

cv_subset <- subset(cv_subset, cells = rownames(cv_subset@meta.data)[grep("Monocyte",
                                                                      cv_subset@meta.data$cell_annot_revision)])
```

cv_subset <- multiModal_processing(object = cv_subset, gex = T,
 adt = F, sct = F, gexAdtWnn = F, sctAdtWnn = F, doHarmony = T,
 npca = 30, regress_cellcycle_gex = F, makeFinalWnnUmap = F,
 doFreshSct = F)

cv_subset <- multiModal_processing(object = cv_subset, gex = F,
 adt = F, sct = T, gexAdtWnn = F, sctAdtWnn = F, doHarmony = T,
 npca = 30, regress_cellcycle_gex = F, makeFinalWnnUmap = F,
 doFreshSct = T)

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")
DimPlot(cv_subset, reduction = "umapAfterHarmony_SCT", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")
```

```

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad

DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "rna_snn_res.0.5")
DotPlot(cv_subset, features = c("log_prob_doublet", "log_prob_singleton"),
        group.by = "sct_snn_res.0.5")
DotPlot(cv_subset, features = c("pANN", "log_prob_doublet", "log_prob_singleton"),
        group.by = "sct_snn_res.0.5")

gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
                  "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
                  "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
                  "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
                  "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
                  "MS4A1", "TNFRSF13B", "ENTPD1", "adt_AB-KIT", "adt_AB-CD34",
                  "KIT", "CD34", "PPBP", "PF4", "HBB")
DotPlot(cv_subset, features = gexOnlyList, group.by = "sct_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()
```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "dataset")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")

DotPlot(cv_subset, features = gexOnlyList, group.by = "rna_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("TPSAB1", "CPA3", "KIT", "PRSS57",
                                "HPGDS", "GATA2", "TNFSF10", "TRIM63", "IGHA1", "IGHA2",
                                "FCER1A"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
    RotatedAxis()
DotPlot(cv_subset, features = c("SEPP1", "AMICA1", "GNLY", "KIAA1598",
                                "IL8", "FTL", "ALAS2", "PTPLAD2", "MS4A7", "APOE"), group.by = "rna_snn_res.0.5",
        cluster.idents = T) + RotatedAxis()
DotPlot(cv_subset, features = c("SPINK2", "MTND1P23", "AL450405.1",
                                "BEX3", "HBG2", "GUCY1A1", "AL513365.1", "SMIM24", "LAPTM4B",
                                "CRHBP"), group.by = "rna_snn_res.0.5", cluster.idents = T) +
    RotatedAxis()
DotPlot(cv_subset, features = c("HDC", "MS4A2", "PRG2", "MS4A3",
                                "TPSAB1", "TPSB2", "EPX", "CLC"), cluster.idents = T) + RotatedAxis()
# DotPlot(cv,features =
# # c('HDC','MS4A2','PRG2','MS4A3','TPSAB1'),group.by =
# # 'rna_snn_res.32',cluster.idents = T) + RotatedAxis()

cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision <- "Doublets"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.0.5 %in%
                                         c("7", "1", "3")] <- "Platelets"

DotPlot(cv_subset, features = gexOnlyList, group.by = "cell_annot_revision",
        cluster.idents = T) + RotatedAxis()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "cell_annot_revision")

# write_rds(cv_subset@meta.data,file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_platelet'

cv_subset <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_bad")
cv_subset@meta.data$type_sample <- gsub("(..)*", "\\\\"1", cv_subset@meta.data$patient_id)
cv_subset@meta.data$rnaBr <- rowSums(FetchData(cv_subset, rownames(cv_subset[["RNA"]]) [grep("^\^IG[KHL].*",
                                         rownames(cv_subset[["RNA"]])]])))
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.0.5
clust05Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
Idents(cv_subset) <- cv_subset@meta.data$rna_snn_res.4
clust4Markers <- FindAllMarkers(object = cv_subset, assay = "RNA",
                                    logfc.threshold = 0.5, min.pct = 0.25, only.pos = T)
clust05Markers[order(clust05Markers$pct.2 - clust05Markers$pct.1),
               ]
clust4Markers[order(clust4Markers$pct.2 - clust4Markers$pct.1),
               ]

```

```

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "orig.ident") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "patient_id") + NoLegend()
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "Phase")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "type_sample", cols = randomcoloR::distinctColorPalette(length(un
        2))
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.0.5")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
        raster = F, label = T, group.by = "rna_snn_res.4")

DotPlot(cv_subset, features = c("adt_AB-PTPRC-2", "IGHD", "FCER2",
    "CD19", "CD24", "CCR7", "TCL1A", "IGHM", "CD79A", "MS4A1",
    "TNFRSF13B", "CR2", "BANK1", "CD27"), group.by = "rna_snn_res.0.5",
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, features = c("adt_AB-PTPRC-2", "IGHD", "FCER2",
    "CD19", "CD24", "CCR7", "TCL1A", "IGHM", "CD79A", "MS4A1",
    "TNFRSF13B", "CR2", "BANK1", "CD27"), group.by = "rna_snn_res.4",
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

markerList <- c("NKG7", "NCAM1", "adt_AB-NCAM1", "CD3D", "adt_AB-CD3D",
    "HLA-DRA", "S100A4", "S100A6", "CCL4", "CCL5", "GZMH", "GZMB",
    "GZMK", "IL32", "IFNG", "IFI6", "IRF7", "IFIT3", "PRF1",
    "FCGR3A", "adt_AB-FCGR3A", "FCER1G", "rnaTr", "adt_AB-CD8A",
    "CD8A", "CD8B", "adt_AB-CD4", "NCR1", "adt_AB-NCR1", "PPBP",
    "PF4", "KLRB1", "TRDV2", "adt_AB-TRDV2", "TRGV9", "adt_AB-TRGV9",
    "IL7R", "KLRC3", "GNLY", "CD27", "adt_AB-CD27", "SELL", "TIGIT",
    "CXCR4", "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-2",
    "adt_AB-PTPRC-3")
gexOnlyList <- c("CD3D", "CCR7", "SELL", "CD27", "CD4", "CD40LG",
    "CD8A", "GZMH", "IL2RA", "FOXP3", "IKZF2", "TRGV9", "TRDV2",
    "TRAV1-2", "SLC4A10", "MKI67", "NCR1", "NCAM1", "FXYD7",
    "FCGR3A", "CD14", "C1QA", "CLEC4C", "IL3RA", "AXL", "SIGLEC6",
    "CLEC9A", "FCER1A", "FCER2", "CXCR5", "CD19", "CCR6", "IGHD",
    "MS4A1", "TNFRSF13B", "ENTPD1", "KIT", "CD34", "PPBP", "PF4",
    "HBB", "HDC", "MS4A2", "PRG2", "MS4A3", "TPSAB1", "TPSB2",
    "EPX", "CLC", "IFI6", "IRF7", "IFI44L")
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = markerList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = markerList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = gexOnlyList,
    cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = gexOnlyList,

```

```

cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("adt_AB-PTPRC-2",
  "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
  "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
  "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
  "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("adt_AB-PTPRC-2",
  "adt_AB-CD4", "adt_AB-CD8A", "IL7R", "CD27", "CCR7", "SELL",
  "CX3CR1", "adt_AB-CX3CR1", "adt_AB-PTPRC-1", "adt_AB-PTPRC-3",
  "GZMH", "PRF1", "PPBP", "PF4", "TRDV2", "adt_AB-TRDV2", "TRGV9",
  "adt_AB-TRGV9"), cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.0.5", features = c("IFNG",
  "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
  "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, group.by = "rna_snn_res.4", features = c("IFNG",
  "TBX21", "TNF", "GATA3", "IL4", "IL5", "RORC", "IL17A", "IL17F",
  "IL21", "CCL5", "PHLDA1", "LYAR", "ODF2L", "IL7R", "PDE4D"),
  cluster.idents = T) + RotatedAxis() + theme(axis.text = element_text(size = 7))

cv_subset@meta.data$cell_annot_revision <- NA
cv_subset@meta.data$cell_annot_revision <- "B naive"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("43", "28", "29", "12")] <- "B switched mem"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("15", "39", "17", "2")] <- "B non-switched mem"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("39")] <- "B non-switched mem IFN induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("3")] <- "B naive IFN induced"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("4")] <- "B exhausted"
cv_subset@meta.data$cell_annot_revision[cv_subset@meta.data$rna_snn_res.4 %in%
  c("38", "41", "42", "32")] <- "doublets"

DotPlot(cv_subset, features = c("adt_AB-PTPRC-2", "IGHD", "FCER2",
  "CD19", "CD24", "CCR7", "TCL1A", "IGHM", "CD79A", "MS4A1",
  "TNFRSF13B", "CR2", "BANK1", "CD27", "IFI44L", "TBX21"),
  group.by = "rna_snn_res.4", cluster.idents = T) + RotatedAxis() +
  theme(axis.text = element_text(size = 7))
DotPlot(cv_subset, features = c("adt_AB-PTPRC-2", "IGHD", "FCER2",
  "CD19", "CD24", "CCR7", "TCL1A", "IGHM", "CD79A", "MS4A1",
  "TNFRSF13B", "CR2", "BANK1", "CD27", "IFI44L", "TBX21"),
  group.by = "cell_annot_revision", cluster.idents = T) + RotatedAxis() +
  theme(axis.text = element_text(size = 7))

DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "rna_snn_res.4")
DimPlot(cv_subset, reduction = "umapAfterHarmony_RNA", shuffle = T,
  raster = F, label = T, group.by = "cell_annot_revision")

# write_rds(cv_subset@meta.data, file =

```

```

# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_subset_B_anno

cv_fil <- cv
# Fix MAIT annotation
cv_subset_meta <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered")
cv_subset_meta <- cv_subset_meta[rownames(cv_subset_meta) %in%
  rownames(cv_fil@meta.data), ]
cv_fil@meta.data[rownames(cv_subset_meta), "cell.annot_revision"] <- cv_subset_meta$cell.annot_revision

# Add IL6 monos
cv_subset_meta <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered")
cv_subset_meta <- cv_subset_meta[rownames(cv_subset_meta) %in%
  rownames(cv_fil@meta.data), ]
cv_fil@meta.data[rownames(cv_subset_meta), "cell.annot_revision"] <- cv_subset_meta$cell.annot_revision

# Make cell labels prettier
cv_fil@meta.data$cell.annot_revision_fullNames <- cv_fil@meta.data$cell.annot_revision
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T CD8 cm"] <- "T CD8 Central Mem"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T CD8 em"] <- "T CD8 Effector Mem"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T CD8 emra"] <- "T CD8 Effector Mem CD45RA+"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T g/d"] <- "T Gamma/Delta"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T regulatory"] <- "T Regulatory"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T CD8 CTL IFN induced"] <- "T CD8 CTL IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "T CD4 IFN induced"] <- "T CD4 IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "NK IFN induced"] <- "NK IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Red Blood Cells"] <- "Red Blood cells"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD14"] <- "Classical Monocyte"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD14 IFN-induced"] <- "Classical Monocyte IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD14 IL6+"] <- "Classical Monocyte IL6+"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD16 IFN-induced"] <- "Non-classical Monocyte IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD16"] <- "Non-classical Monocyte"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "Monocyte CD16+C1"] <- "Non-classical Monocyte Complement+"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "HPSCs IFN induced"] <- "Hematopoietic progenitors IFN-induced"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "HPSCs"] <- "Hematopoietic progenitors"
cv_fil@meta.data$cell.annot_revision_fullNames[cv_fil@meta.data$cell.annot_revision_fullNames == "cycling"] <- "Cycling"

```

```

cv_file@meta.data$cell_annot_revision_fullNames[cv_file@meta.data$cell_annot_revision_fullNames ==
  "B non-switched mem IFN induced"] <- "B non-switched mem IFN-induced"
cv_file@meta.data$cell_annot_revision_fullNames[cv_file@meta.data$cell_annot_revision_fullNames ==
  "B naive IFN induced"] <- "B naive IFN-induced"

# Also make short names
cv_file@meta.data$cell_annot_revision_short <- cv_file@meta.data$cell_annot_revision
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T CD8 cm"] <- "T CD8 CM"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T CD8 em"] <- "T CD8 EM"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T CD8 emra"] <- "T CD8 EMRA"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T regulatory"] <- "T reg"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T CD8 CTL IFN induced"] <- "T CD8 CTL IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "T CD4 IFN induced"] <- "T CD4 IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "NK IFN induced"] <- "NK IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "Red Blood Cells"] <- "RBC"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "Monocyte CD14 IFN-induced"] <- "Monocyte CD14 IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "Monocyte CD14 IL6+"] <- "Monocyte CD14 IL6"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "Monocyte CD16 IFN-induced"] <- "Monocyte CD16 IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "HPSCs IFN induced"] <- "HPC IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "HPSCs"] <- "HPC"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "cycling"] <- "Cycling"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "B non-switched mem IFN induced"] <- "B n-sw mem IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "B non-switched mem"] <- "B n-sw mem"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "B switched mem"] <- "B sw mem"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "B naive IFN induced"] <- "B naive IFNi"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "B exhausted"] <- "B exh"
cv_file@meta.data$cell_annot_revision_short[cv_file@meta.data$cell_annot_revision_short ==
  "Mast & Eosinophils"] <- "Mast/Eos"

# broad labels
cv_file@meta.data$cell_annot_revision_broad <- cv_file@meta.data$cell_annot_revision
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "T CD8 cm"] <- "T CD8+"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==

```

```

    "T CD8 em"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD8 emra"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD8 CTL IFN induced"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "NKT"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD8 CTL"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD8 EM"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T g/d"] <- "T g/d"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T regulatory"] <- "T Reg"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD4 IFN induced"] <- "T CD4+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD4 Helper"] <- "T CD4+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD4 CTL"] <- "T CD4+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD4 CTL"] <- "T CD4+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD4 Naive"] <- "T CD4+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "T CD8 Naive"] <- "T CD8+"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "NK IFN induced"] <- "NK"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "NK FCR1G+"] <- "NK"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "NK CD56 bright"] <- "NK"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "NK HLA-DR+"] <- "NK"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Red Blood Cells"] <- "RBC"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD14"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD14 IFN-induced"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD14 IL6+"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD16 IFN-induced"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD16"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "Monocyte CD16+C1"] <- "Monocyte"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "HPSCs IFN induced"] <- "HPC"
cv_fil@meta.data$cell_annot_revision_broad[cv_fil@meta.data$cell_annot_revision_broad ==
    "HPSCs"] <- "HPC"

```

```

cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "cycling"] <- "Cycling"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B non-switched mem IFN induced"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B naive IFN induced"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B naive"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B non-switched mem"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B switched mem"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "B exhausted"] <- "B"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "AS-DC"] <- "DC"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "cDC1"] <- "DC"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "cDC2"] <- "DC"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "pDC"] <- "DC"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "pDC"] <- "DC"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "Plasma cells"] <- "Plasma"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "Plasmablasts"] <- "Plasma"
cv_file@meta.data$cell_annot_revision_broad[cv_file@meta.data$cell_annot_revision_broad ==
  "Mast & Eosinophils"] <- "Mast/Eos"

# write_rds(cv_file, file =
# '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_fil2.rds', compress = TRUE)
cv <- cv_file

```

Visualize marker genes

```

# Ignore the IFN pops for now TNK first
cv_subset <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$cell_annot_revision_broad %in%
  c("T CD4+", "T CD8+", "T g/d", "T Reg", "MAIT", "NK", "ILC") &
  !grepl("IFNi", cv@meta.data$cell_annot_revision_short)])}

markersGex <- unique(c("CD3D", "CD4", "CD8A", "CCR7", "CD27",
  "SELL", "CX3CR1", "IL7R", "PTPRC-2", "PTPRC-3", "GZMH", "PRF1",
  "TRGV9", "TRDV2", "FOXP3", "IL2RA", "TRAV1-2", "SLC4A10",
  "NCR1", "NCAM1", "GNLY", "FCER1G", "HLA-DRA", "TNFRSF4",
  "CD14", "FCGR3A", "IL6", "C1QA", "CLEC4C", "IL3RA", "AXL",
  "SIGLEC6", "CLEC9A", "FCER1A", "FCER2", "IGHD", "CD19", "CD24",
  "TCL1A", "IGHM", "CD79A", "MS4A1", "TNFRSF13B", "CR2", "BANK1",
  "JCHAIN", "IGHG1", "TNFRSF13B", "ENTPD1", "KIT", "CD34",
  "SPINK2", "TPSAB1", "TPSB2", "PRG2", "EPX", "MKI67", "PPBP",
  "PF4", "HBB"))

markersAdt <- paste0("AB-", markersGex)

```

```

TmarkersGex <- c("CD3D", "CD4", "CD8A", "CCR7", "CD27", "SELL",
  "CX3CR1", "IL7R", "PTPRC-2", "PTPRC-3", "GZMH", "PRF1", "TRGV9",
  "TRDV2", "FOXP3", "IL2RA", "TRAV1-2", "TRAV7", "SLC4A10",
  "NCR1", "NCAM1", "GNLY", "FCER1G", "HLA-DRA", "AREG", "TNFRSF18",
  "TNFRSF4")

TmarkersAdt <- paste0("AB-", TmarkersGex)

TmarkersGex <- unique(TmarkersGex[TmarkersGex %in% rownames(cv[["RNA"]])])
TmarkersAdt <- unique(TmarkersAdt[TmarkersAdt %in% rownames(cv[["ADT"]])])

# tem: CCR7lo SELLlo CX3CR1hi CD27lo IL7Rhi CD27- CD45RA-
# '30', '1', tcm: CCR7hi SELLhi CX3CR1lo CD27hi IL7Rhi CD27+
# CD45RA- '35', '36', '17', '7', '15', '25', '19' temra: CCR7-
# IL7Rlo CD27- CD45RA+ '5', '3', '24',

(DotPlot(cv_subset, features = TmarkersGex, assay = "RNA", cluster.idents = F,
  group.by = "cell_annot_revision_short", col.min = 0, cols = c("lightgrey",
    "blue")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9))) +
  (DotPlot(cv_subset, features = TmarkersAdt, assay = "ADT",
    cluster.idents = F, group.by = "cell_annot_revision_short",
    col.min = 0, cols = c("lightgrey", "red")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9)))

# Mono DC
cv_subset <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$cell_annot_revision_broad %in%
  c("Monocyte", "DC") & !grepl("IFNi", cv@meta.data$cell_annot_revision_short)])

MmarkersGex <- c("CD14", "FCGR3A", "IL6", "C1QA", "CLEC4C", "IL3RA",
  "AXL", "SIGLEC6", "CLEC9A", "FCER1A")
MmarkersAdt <- paste0("AB-", MmarkersGex)

MmarkersGex <- unique(MmarkersGex[MmarkersGex %in% rownames(cv[["RNA"]])])
MmarkersAdt <- unique(MmarkersAdt[MmarkersAdt %in% rownames(cv[["ADT"]])])

(DotPlot(cv_subset, features = MmarkersGex, assay = "RNA", cluster.idents = F,
  group.by = "cell_annot_revision_short", col.min = 0, cols = c("lightgrey",
    "blue")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9))) +
  (DotPlot(cv_subset, features = MmarkersAdt, assay = "ADT",
    cluster.idents = F, group.by = "cell_annot_revision_short",
    col.min = 0, cols = c("lightgrey", "red")) + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9)))

# B plasma
cv_subset <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$cell_annot_revision_broad %in%
  c("B", "Plasma") & !grepl("IFNi", cv@meta.data$cell_annot_revision_short)])

# 'adt_AB-PTPRC-2', 'IGHD', 'FCER2', 'CD19', 'CD24', 'CCR7', 'TCL1A', 'IGHM', 'CD79A', 'MS4A1', 'TNFRSF13B', 'CR2'

```

```

BmarkersGex <- c("CCR7", "TCL1A", "FCER2", "CD19", "CD22", "CD79A",
  "MS4A1", "BANK1", "IGHM", "IGHD", "TNFRSF13B", "CR2", "BANK1",
  "CD27", "JCHAIN", "TNFRSF13B", "ENTPD1", "CD38", "MKI67")
BmarkersAdt <- paste0("AB-", BmarkersGex)

BmarkersGex <- unique(BmarkersGex[BmarkersGex %in% rownames(cv[["RNA"]])])
BmarkersAdt <- unique(BmarkersAdt[BmarkersAdt %in% rownames(cv[["ADT"]])])

(DotPlot(cv_subset, features = BmarkersGex, assay = "RNA", cluster.idents = F,
  group.by = "cell_annot_revision_short", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9))) +
(DotPlot(cv_subset, features = BmarkersAdt, assay = "ADT",
  cluster.idents = F, group.by = "cell_annot_revision_short",
  col.min = 0, cols = c("lightgrey", "red")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9)))

# Other celltypes
cv_subset <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$cell_annot_revision_broad %in%
  c("RBC", "Platelets", "Cycling", "Mast/Eos", "HPC") & !grepl("IFNi",
  cv@meta.data$cell_annot_revision_short)])

```



```

OmarkersGex <- c("KIT", "CD34", "SPINK2", "HDC", "MS4A2", "PRG2",
  "MS4A3", "TPSAB1", "TPSB2", "EPX", "CLC", "MKI67", "PPBP",
  "PF4", "HBB")
OmarkersAdt <- paste0("AB-", OmarkersGex)
#'HDC', 'MS4A2', 'PRG2', 'MS4A3', 'TPSAB1', 'TPSB2', 'EPX', 'CLC'

OmarkersGex <- unique(OmarkersGex[OmarkersGex %in% rownames(cv[["RNA"]])])
OmarkersAdt <- unique(OmarkersAdt[OmarkersAdt %in% rownames(cv[["ADT"]])])

(DotPlot(cv_subset, features = OmarkersGex, assay = "RNA", cluster.idents = F,
  group.by = "cell_annot_revision_short", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9))) +
(DotPlot(cv_subset, features = OmarkersAdt, assay = "ADT",
  cluster.idents = F, group.by = "cell_annot_revision_short",
  col.min = 0, cols = c("lightgrey", "red")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9)))

```

Further discussions with immunologists made us make small changes to our initial annotation

```

cv <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_"

# B exh becomes invariant B cells (based on
# 'TBX21', 'FCRL5', 'FCRL3') NK HLA-DRA+ becomes NK T CD4 IFN
# induced becomes T CD4 Naive IFN induced IFN induced
# becomes IFN stim NK FCER1G+ becomes NK Remove PIMS
# patients PP12 and PP18
cv <- subset(cv, cells = rownames(cv@meta.data)[!cv@meta.data$patient_id %in%
  c("PP12", "PP18")])

```

```

cv@meta.data$cell_annot_revision_fullNames <- cv@meta.data$cell_annot_revision
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD8 cm"] <- "T CD8 central mem"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD8 Naive"] <- "T CD8 naive"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD4 Naive"] <- "T CD4 naive"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD4 Helper"] <- "T CD4 helper"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "NK HLA-DR+"] <- "NK"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "NK FCER1G+"] <- "NK"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "B exhausted"] <- "B invariant"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames %in% c("T CD8 EM", "T CD8 em")] <- "T CD8 effector mem"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD8 emra"] <- "T CD8 effector mem CD45RA+"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T g/d"] <- "T gamma/delta"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T regulatory"] <- "T regulatory"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD8 CTL IFN induced"] <- "T CD8 CTL IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "T CD4 IFN induced"] <- "T CD4 naive IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "NK IFN induced"] <- "NK IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Red Blood Cells"] <- "Red blood cells"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD14"] <- "Classical monocyte"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD14 IFN-induced"] <- "Classical monocyte IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD14 IL6+"] <- "Classical monocyte IL6+"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD16 IFN-induced"] <- "Non-classical monocyte IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD16"] <- "Non-classical monocyte"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "Monocyte CD16+C1"] <- "Non-classical monocyte complement+"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "HPSCs IFN induced"] <- "Hematopoietic progenitors IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "HPSCs"] <- "Hematopoietic progenitors"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "cycling"] <- "Cycling"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "B non-switched mem IFN induced"] <- "B non-switched mem IFN stim"
cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames == "B naive IFN induced"] <- "B naive IFN stim"

```

```

cv@meta.data$cell_annot_revision_fullNames[cv@meta.data$cell_annot_revision_fullNames ==  

  "Mast & Eosinophils"] <- "Basophils & Eosinophils"  
  

# Also make short names  

cv@meta.data$cell_annot_revision_short <- cv@meta.data$cell_annot_revision  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 cm"] <- "T CD8 CM"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 Naive"] <- "T CD8 naive"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 Naive"] <- "T CD4 naive"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 Helper"] <- "T CD4 helper"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "NK HLA-DR+"] <- "NK"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "NK FCER1G+"] <- "NK"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short %in%  

  c("T CD8 EM", "T CD8 em")] <- "T CD8 EM"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 emra"] <- "T CD8 EMRA"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T regulatory"] <- "T reg"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 CTL IFN induced"] <- "T CD8 CTL IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 IFN induced"] <- "T CD4 naive IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "NK IFN induced"] <- "NK IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "Red Blood Cells"] <- "RBC"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD14 IFN-induced"] <- "Monocyte CD14 IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD14 IL6+"] <- "Monocyte CD14 IL6"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD16 IFN-induced"] <- "Monocyte CD16 IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "HPSCs IFN induced"] <- "HPC IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "HPSCs"] <- "HPC"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "cycling"] <- "Cycling"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "B non-switched mem IFN induced"] <- "B n-sw mem IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "B non-switched mem"] <- "B n-sw mem"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "B switched mem"] <- "B sw mem"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==  

  "B naive IFN induced"] <- "B naive IFN stim"  

cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==

```

```

    "B exhausted"] <- "B invar"
cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==
  "Mast & Eosinophils"] <- "Baso/Eos"
cv@meta.data$cell_annot_revision_short[cv@meta.data$cell_annot_revision_short ==
  "NK CD56 bright"] <- "NK CD56"

cv@meta.data$cell_annot_revision_short_wolfnStim <- gsub(" IFN stim",
  "", cv@meta.data$cell_annot_revision_short)

# broad labels
cv@meta.data$cell_annot_revision_broad <- cv@meta.data$cell_annot_revision
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 cm"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad %in%
  c("T CD8 EM", "T CD8 em")] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 emra"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 CTL IFN induced"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "NKT"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 CTL"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 EM"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T g/d"] <- "T g/d"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T regulatory"] <- "T reg"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD4 IFN induced"] <- "T CD4+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD4 Helper"] <- "T CD4+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD4 CTL"] <- "T CD4+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD4 CTL"] <- "T CD4+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD4 Naive"] <- "T CD4+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "T CD8 Naive"] <- "T CD8+"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "NK IFN induced"] <- "NK"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "NK FCER1G+"] <- "NK"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "NK CD56 bright"] <- "NK"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "NK HLA-DR+"] <- "NK"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "Red Blood Cells"] <- "RBC"
cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==
  "Monocyte CD14"] <- "Monocyte"

```

```

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Monocyte CD14 IFN-induced"] <- "Monocyte"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Monocyte CD14 IL6+"] <- "Monocyte"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Monocyte CD16 IFN-induced"] <- "Monocyte"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Monocyte CD16"] <- "Monocyte"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Monocyte CD16+C1"] <- "Monocyte"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "HPSCs IFN induced"] <- "HPC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "HPSCs"] <- "HPC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "cycling"] <- "Cycling"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B non-switched mem IFN induced"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B naive IFN induced"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B naive"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B non-switched mem"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B switched mem"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "B exhausted"] <- "B"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "AS-DC"] <- "DC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "cDC1"] <- "DC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "cDC2"] <- "DC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "pDC"] <- "DC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "pDC"] <- "DC"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Plasma cells"] <- "Plasma"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Plasmablasts"] <- "Plasma"  

cv@meta.data$cell_annot_revision_broad[cv@meta.data$cell_annot_revision_broad ==  

  "Mast & Eosinophils"] <- "Baso/Eos"

cv@meta.data$labelOrder <- NA  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 naive"] <- 1  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 helper"] <- 3  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 naive IFN stim"] <- 45

```

```

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD4 CTL"] <- 4  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 naive"] <- 5  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 CM"] <- 6  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 EM"] <- 7  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 EMRA"] <- 8  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 CTL IFN stim"] <- 46  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T CD8 CTL"] <- 10  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T g/d"] <- 11  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "T reg"] <- 12  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "MAIT"] <- 13  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "NKT"] <- 14  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "NK"] <- 15  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "NK CD56"] <- 16  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "NK IFN stim"] <- 47  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "ILC"] <- 20  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD14"] <- 21  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD14 IFN stim"] <- 48  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD14 IL6"] <- 23  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD16"] <- 24  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD16 IFN stim"] <- 49  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "Monocyte CD16+C1"] <- 26  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "pDC"] <- 27  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "cDC1"] <- 29  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "cDC2"] <- 30  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "AS-DC"] <- 28  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==  

  "B naive"] <- 31  

cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short ==

```

```

    "B naive IFN stim"] <- 50
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "B n-sw mem"] <- 33
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "B n-sw mem IFN stim"] <- 51
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "B sw mem"] <- 35
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "B invar"] <- 36
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "Plasma cells"] <- 37
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "Plasmablasts"] <- 38
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "HPC"] <- 39
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "HPC IFN stim"] <- 52
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "Baso/Eos"] <- 41
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "Cycling"] <- 42
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "Platelets"] <- 43
cv@meta.data$labelOrder[cv@meta.data$cell_annot_revision_short == "RBC"] <- 44

cv@meta.data$cell_annot_revision_short <- factor(cv@meta.data$cell_annot_revision_short,
  levels = cv@meta.data$cell_annot_revision_short[!duplicated(cv@meta.data$cell_annot_revision_short)],
  decreasing = T))
cv@meta.data$cell_annot_revision_fullNames <- factor(cv@meta.data$cell_annot_revision_fullNames,
  levels = cv@meta.data$cell_annot_revision_fullNames[!duplicated(cv@meta.data$cell_annot_revision_fullNames)],
  decreasing = T))
cv@meta.data$cell_annot_revision_broad <- factor(cv@meta.data$cell_annot_revision_broad,
  levels = cv@meta.data$cell_annot_revision_broad[!duplicated(cv@meta.data$cell_annot_revision_broad)],
  decreasing = T))
cv@meta.data$cell_annot_revision_short_woIfnStim <- factor(cv@meta.data$cell_annot_revision_short_woIfnStim,
  levels = cv@meta.data$cell_annot_revision_short_woIfnStim[!duplicated(cv@meta.data$cell_annot_revision_short_woIfnStim)],
  decreasing = T))

markersGex <- unique(c("CD3D", "CD4", "CD8A", "CCR7", "CD27",
  "SELL", "CX3CR1", "IL7R", "PTPRC-2", "PTPRC-3", "GZMH", "PRF1",
  "TRGV9", "TRDV2", "FOXP3", "IL2RA", "TRAV1-2", "SLC4A10",
  "NCR1", "NCAM1", "GNLY", "TNFRSF18", "TNFRSF4", "FCER1G",
  "CD14", "FCGR3A", "IL6", "C1QA", "CLEC4C", "IL3RA", "AXL",
  "SIGLEC6", "CLEC9A", "FCER1A", "FCER2", "IGHD", "CD19", "CD24",
  "TCL1A", "IGHM", "CD79A", "MS4A1", "TNFRSF13B", "CR2", "BANK1",
  "JCHAIN", "IGHG1", "TNFRSF13B", "TBX21", "FCRL5", "FCRL3",
  "ENTPD1", "KIT", "CD34", "SPINK2", "TPSAB1", "TPSB2", "PRG2",
  "EPX", "MKI67", "PPBP", "PF4", "HBB", "IFI44L", "MX2", "IFI6"))
markersAdt <- paste0("AB-", markersGex)

markersGex <- markersGex[markersGex %in% rownames(cv[["RNA"]])]
markersAdt <- markersAdt[markersAdt %in% rownames(cv[["ADT"]])]
```

```

(DotPlot(cv, group.by = "cell_annot_revision_short", features = markersGex,
  cluster.idents = F, assay = "RNA", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("RNA"))
(DotPlot(cv, group.by = "cell_annot_revision_fullNames", features = markersGex,
  cluster.idents = F, assay = "RNA", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("RNA"))
(DotPlot(cv, group.by = "cell_annot_revision_broad", features = markersGex,
  cluster.idents = F, assay = "RNA", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("RNA"))
(DotPlot(cv, group.by = "cell_annot_revision_short", features = markersAdt,
  cluster.idents = F, assay = "ADT", col.min = 0, cols = c("lightgrey",
  "red")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("ADT"))
(DotPlot(cv, group.by = "cell_annot_revision_fullNames", features = markersAdt,
  cluster.idents = F, assay = "ADT", col.min = 0, cols = c("lightgrey",
  "red")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("ADT"))
(DotPlot(cv, group.by = "cell_annot_revision_broad", features = markersAdt,
  cluster.idents = F, assay = "ADT", col.min = 0, cols = c("lightgrey",
  "red")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("ADT"))

# write_rds(cv,file='/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdt'
#           # = 'gz')

```

We use scirpy to transform the vdj data in a cell / row format

```

bcrTable <- manis[!is.na(manis$BCR), c("sample_id", "TCR", "BCR")]
tcrTable <- manis[!is.na(manis$TCR), c("sample_id", "TCR", "BCR")]

```

```

# py_install(pip = T, packages = "scirpy")
import sys
import warnings

import numpy as np
import pandas as pd
import pandas

import scanpy as sc
import scirpy as ir
from matplotlib import pyplot as plt
import seaborn as sns
import matplotlib.pyplot as plt

```

```

import scipy.stats
import scipy as sp
import anndata
import os
from glob import glob

meta_GEX_VDJ = r.bcrTable.set_index('BCR')
meta_GEX_VDJ.head(3)

holder = []

for sample_vdj in meta_GEX_VDJ.index:

    holder.append(ir.io.read_10x_vdj('/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/bcr/' + sample_vdj + '/'))

    sample_gex = meta_GEX_VDJ.loc[sample_vdj, 'sample_id']
    holder[-1].obs_names = [sample_gex + '_' + i.split('-')[0] for i in holder[-1].obs_names]

    adata_bcr = pd.concat([i.obs for i in holder])
    adata_bcr.to_csv("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/fileNames_vdj_bcr_210712_toScirpy.csv")

#Do the same but for TCR
meta_GEX_VDJ = r.tcrTable.set_index('TCR')
meta_GEX_VDJ.head(3)

holder = []

for sample_vdj in meta_GEX_VDJ.index:

    holder.append(ir.io.read_10x_vdj('/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/tcr/' + sample_vdj + '/'))

    sample_gex = meta_GEX_VDJ.loc[sample_vdj, 'sample_id']
    holder[-1].obs_names = [sample_gex + '_' + i.split('-')[0] for i in holder[-1].obs_names]

    adata_tcr = pd.concat([i.obs for i in holder])
    adata_tcr.to_csv("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/fileNames_vdj_tcr_210712_toScirpy.csv")

# Somehow r.adata gives a malformed factor..

allTcr <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/fileNames_vdj_tcr_210712_toScirpy.csv")
  header = T, stringsAsFactors = F, sep = ","
allBcr <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/fileNames_vdj_bcr_210712_toScirpy.csv")
  header = T, stringsAsFactors = F, sep = ","
allBcr$gex_barcode <- paste0(allBcr$X, "-1")
allTcr$gex_barcode <- paste0(allTcr$X, "-1")

cv@meta.data[allTcr$gex_barcode, colnames(allTcr)[!colnames(allTcr) %in%
  colnames(cv@meta.data)]] <- allTcr[, !colnames(allTcr) %in%
  colnames(cv@meta.data)]
cv@meta.data[allBcr$gex_barcode, colnames(allBcr)[!colnames(allBcr) %in%
  colnames(cv@meta.data)]] <- allBcr[, !colnames(allBcr) %in%
  colnames(cv@meta.data)]

```

```

# write_rds(cv, '/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_
# = 'gz')

cv <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/cov_oldNewMerged_filtered_badAdtRem_"

# library(sceasy) library(reticulate) loompy <-
# reticulate::import('loompy')
# sceasy::convertFormat('/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/nasal/covid_airway_20210501
# from='anndata', to='seurat',
# outFile='/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/nasal/covid_airway_20210501.soupx.bbknn_p

nasal <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/nasal/covid_airway_20210501.soupx
nasal@meta.data$COVID19_infected_cell <- ifelse(nasal[["RNA"]]\@data["VIRAL-SARS-CoV2",
] > 0, "COVID19_infected_cell", "not_infected_cell")

```

Airway data was annotated and analysed separately by Ni Huang, and integrated here. First we calculate relative cell type proportions, then we calculate blood to airway correlations

```

nasalProps <- as.data.frame(table(paste0(nasal@meta.data$donor,
"_", nasal@meta.data$Sample_location), nasal@meta.data$v6_annot2),
stringsAsFactors = F)
nasalProps_broad <- as.data.frame(table(paste0(nasal@meta.data$donor,
"_", nasal@meta.data$Sample_location), nasal@meta.data$v6_broad_annot2),
stringsAsFactors = F)
nasalProps_epi <- as.data.frame(table(paste0(nasal@meta.data$donor[nasal@meta.data$v6_broad_annot2 ==
"Epi"], "_", nasal@meta.data$Sample_location[nasal@meta.data$v6_broad_annot2 ==
"Epi"]), nasal@meta.data$v6_annot2[nasal@meta.data$v6_broad_annot2 ==
"Epi"]), stringsAsFactors = F)
nasalProps_immune <- as.data.frame(table(paste0(nasal@meta.data$donor[nasal@meta.data$v6_broad_annot2 ==
"Immune"], "_", nasal@meta.data$Sample_location[nasal@meta.data$v6_broad_annot2 ==
"Immune"]), nasal@meta.data$v6_annot2[nasal@meta.data$v6_broad_annot2 ==
"Immune"]), stringsAsFactors = F)

nasalProps_epi$prop <- NA
for (i in unique(nasalProps_epi$Var1)) {
  nasalProps_epi[nasalProps_epi$Var1 == i, "prop"] <- nasalProps_epi[nasalProps_epi$Var1 ==
    i, "Freq"]/sum(nasalProps_epi[nasalProps_epi$Var1 ==
    i, "Freq"]))
}
nasalProps_immune$prop <- NA
for (i in unique(nasalProps_immune$Var1)) {
  nasalProps_immune[nasalProps_immune$Var1 == i, "prop"] <- nasalProps_immune[nasalProps_immune$Var1 ==
    i, "Freq"]/sum(nasalProps_immune[nasalProps_immune$Var1 ==
    i, "Freq"]))
}

nasalProps_epi_covidInfected <- as.data.frame(table(paste0(nasal@meta.data$donor[nasal@meta.data$v6_broad_annot2 ==
"Epi"], "_", nasal@meta.data$Sample_location[nasal@meta.data$v6_broad_annot2 ==
"Epi"]), nasal@meta.data$COVID19_infected_cell[nasal@meta.data$v6_broad_annot2 ==
"Epi"]), stringsAsFactors = F)
nasalProps_epi_covidInfected$prop <- NA
nasalProps_epi <- rbind(nasalProps_epi, nasalProps_epi_covidInfected)

```

```

nasalProps_immune_covidInfected <- as.data.frame(table(paste0(nasal@meta.data$donor[nasal@meta.data$v6_Immune"], "_", nasal@meta.data$Sample_location[nasal@meta.data$v6_broad_annot2 == "Immune"]), nasal@meta.data$COVID19_infected_cell[nasal@meta.data$v6_broad_annot2 == "Immune"])), stringsAsFactors = F)
nasalProps_immune_covidInfected$prop <- NA
nasalProps_immune <- rbind(nasalProps_immune, nasalProps_immune_covidInfected)

for (i in unique(nasalProps_epi$Var1)) {
  nasalProps_epi[nasalProps_epi$Var1 == i & nasalProps_epi$Var2 ==
    "COVID19_infected_cell", "prop"] <- nasalProps_epi[nasalProps_epi$Var1 ==
      i & nasalProps_epi$Var2 == "COVID19_infected_cell", "Freq"]/sum(nasalProps_epi[nasalProps_epi$Var1 ==
        i & nasalProps_epi$Var2 %in% c("not_infected_cell", "COVID19_infected_cell"),
        "Freq"])
}
for (i in unique(nasalProps_immune$Var1)) {
  nasalProps_immune[nasalProps_immune$Var1 == i & nasalProps_immune$Var2 ==
    "COVID19_infected_cell", "prop"] <- nasalProps_immune[nasalProps_immune$Var1 ==
      i & nasalProps_immune$Var2 == "COVID19_infected_cell",
      "Freq"]/sum(nasalProps_immune[nasalProps_immune$Var1 ==
        i & nasalProps_immune$Var2 %in% c("not_infected_cell",
        "COVID19_infected_cell"), "Freq"])
}

nasalProps_epi <- nasalProps_epi[nasalProps_epi$Var2 != "not_infected_cell",
  ]
nasalProps_immune <- nasalProps_immune[nasalProps_immune$Var2 != "not_infected_cell",
  ]
for (i in unique(nasalProps_epi$Var2)) {
  if (sum(nasalProps_epi$Freq[nasalProps_epi$Var2 == i]) ==
    0) {
    nasalProps_epi <- nasalProps_epi[nasalProps_epi$Var2 !=
      i, ]
  }
  if (sum(nasalProps_immune$Freq[nasalProps_immune$Var2 ==
    i]) == 0) {
    nasalProps_immune <- nasalProps_immune[nasalProps_immune$Var2 !=
      i, ]
  }
}

bloodProps <- as.data.frame(table(paste0(cv@meta.data$patient_id[!grepl("post",
  ignore.case = T, cv@meta.data$patient_id)], "_Blood"), cv@meta.data$cell_annotation_revision_short[!grep(
  ignore.case = T, cv@meta.data$patient_id)]), stringsAsFactors = F)
bloodProps$prop <- NA
for (i in unique(bloodProps$Var1)) {
  bloodProps[bloodProps$Var1 == i, "prop"] <- bloodProps[bloodProps$Var1 ==
    i, "Freq"]/sum(bloodProps[bloodProps$Var1 == i, "Freq"])
}
bloodProps$id <- gsub("(.*?)_(.*)", "\\\1", bloodProps$Var1)
bloodProps$tissue <- gsub("(.*?)_(.*)", "\\\2", bloodProps$Var1)
bloodProps$celltype <- "Immune"
nasalProps_immune$id <- gsub("(.*?)_(.*)", "\\\1", nasalProps_immune$Var1)
nasalProps_immune$tissue <- gsub("(.*?)_(.*)", "\\\2", nasalProps_immune$Var1)

```

```

nasalProps_immune$celltype <- "Immune"
nasalProps_epi$id <- gsub("(.*?)_(.*)", "\\1", nasalProps_epi$Var1)
nasalProps_epi$tissue <- gsub("(.*?)_(.*)", "\\2", nasalProps_epi$Var1)
nasalProps_epi$celltype <- "Epi"

myProbs <- rbind(bloodProps, nasalProps_immune, nasalProps_epi)
myProbs <- myProbs[myProbs$id %in% cv@meta.data$ID & myProbs$id %in%
  nasal@meta.data$donor, ]
myProbs$days_since_symptoms <- NA
myProbs$prob_inflamed_epi1 <- NA
myProbs$prob_inflamed_epi2 <- NA
myProbs$prob_IFN_stim <- NA
for (i in unique(myProbs$id)) {
  myProbs$days_since_symptoms[myProbs$id == i] <- unique(cv@meta.data$If.COVID.19...Interval.between.::
    i])
  myProbs$prob_inflamed_epi1[myProbs$id == i] <- max(unique(myProbs$prop[myProbs$Var2 ==
    "Transit epi 1" & myProbs$id == i]))
  myProbs$prob_inflamed_epi2[myProbs$id == i] <- max(unique(myProbs$prop[myProbs$Var2 ==
    "Transit epi 2" & myProbs$id == i]))
  myProbs$prob_IFN_stim[myProbs$id == i] <- mean(myProbs$prop[grep("IFN stim",
    myProbs$Var2) & myProbs$id == i])
}
myProbs$days_since_symptoms <- as.numeric(myProbs$days_since_symptoms)

# Do a cor test
myProbs$spread_Var2 <- paste(myProbs$Var2, myProbs$tissue, myProbs$celltype,
  sep = "_")
spreadProbs <- pivot_wider(myProbs, names_from = "spread_Var2",
  values_from = "prop", -c("Freq", "Var2", "celltype", "tissue",
  "Var1"))
spreadProbs <- spreadProbs[, colSums(apply(spreadProbs, 2, is.na)) != 55]

```

```

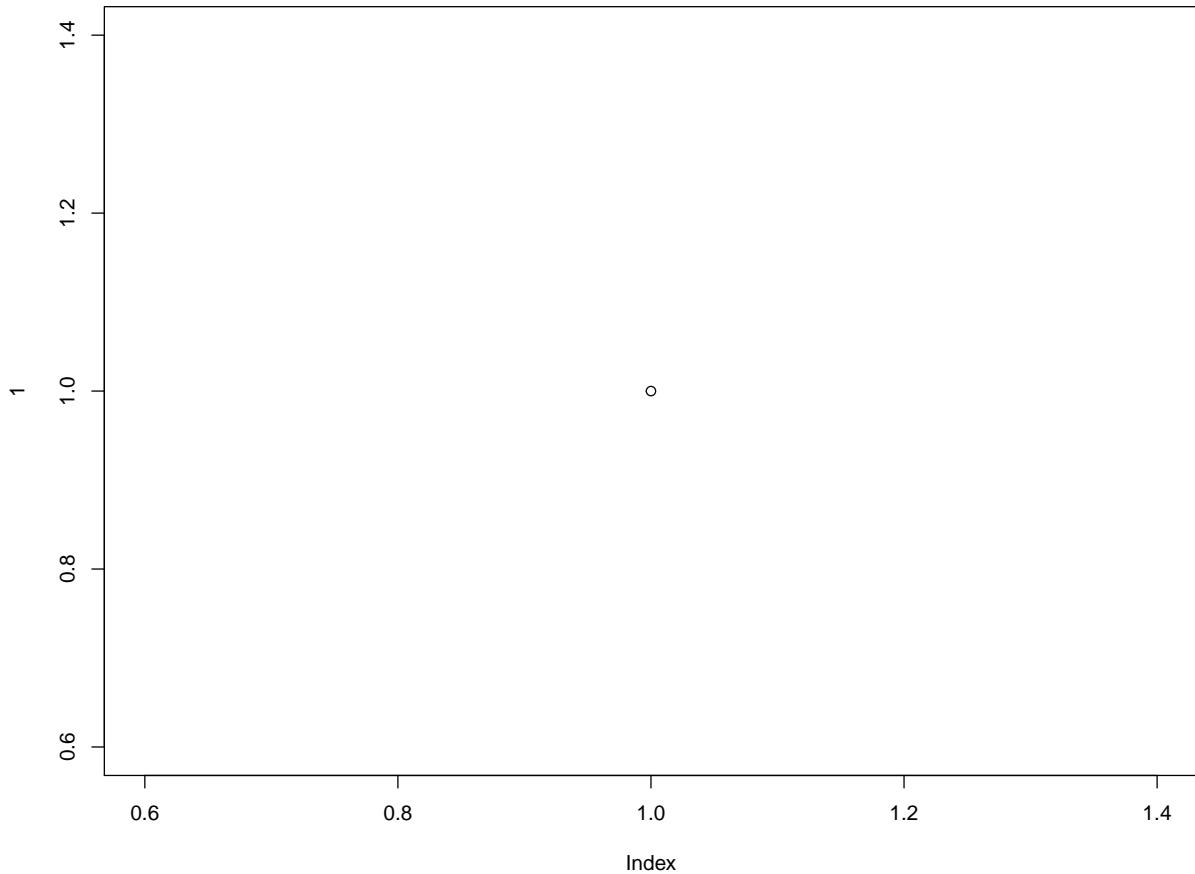
myCorVector_fil_list <- list()
myCorTestVector_fil_list <- list()
for (i in c("^PP", "^AP")) {
  # PP are paediatric covid samples and AP adult covid
  # samples
  spreadProbs_fil <- spreadProbs[grep(i, spreadProbs$id),
    !grep("Trachea", colnames(spreadProbs))]
  myCorTestVector <- suppressWarnings(psych::corr.test(spreadProbs_fil[,
    6:ncol(spreadProbs_fil)], adjust = "none", method = "spearman"))
  myCorVector_fil <- suppressWarnings(as.data.frame(cor(spreadProbs_fil[,
    6:ncol(spreadProbs_fil)], use = "pairwise.complete.obs",
    method = "spearman"), stringsAsFactors = F))
  myCorTestVector_fil <- as.data.frame(myCorTestVector$p)
  myCorTestVector_fil <- myCorTestVector_fil[grep("Blood_Immune",
    rownames(myCorTestVector_fil)), !grep("Blood", colnames(myCorTestVector_fil))]
  myCorVector_fil <- myCorVector_fil[grep("Blood_Immune",
    rownames(myCorVector_fil)), !grep("Blood", colnames(myCorVector_fil))]
  myCorVector_fil[is.na(myCorVector_fil)] <- 0
  myCorTestVector_fil[is.na(myCorTestVector_fil)] <- 1
  myCorVector_fil_list[[i]] <- myCorVector_fil

```

```

myCorTestVector_fil_list[[i]] <- myCorTestVector_fil
myCorVector_fil <- myCorVector_fil[, order(!grep("Immune",
  colnames(myCorVector_fil)))]
myCorTestVector_fil <- myCorTestVector_fil[, order(!grep("Immune",
  colnames(myCorTestVector_fil)))]
rownames(myCorVector_fil) <- gsub("_Blood_Immune", "", rownames(myCorVector_fil))
colnames(myCorVector_fil) <- gsub("_Nose_(Epi|Immune)", "",
  colnames(myCorVector_fil))
# par(mar=c(6,10,2,16),family='Liberation Sans')
# Dotplot_forCorHeatmap(myCorVector_fil,
#   SORT=c(F,F),zlim=c(-1,1),ltsr=1-myCorTestVector_fil,cex=0.8,measure=paste('r(s)',i),cex.axis=.5,s
}
myClust_row <- rownames(myCorVector_fil_list[[1]])[hclust(dist(cbind(myCorVector_fil_list[[1]],
  myCorVector_fil_list[[2]])), method = "complete")$order]
myClust_col <- colnames(myCorVector_fil_list[[1]])[hclust(dist(t(rbind(myCorVector_fil_list[[1]],
  myCorVector_fil_list[[2]]))), method = "complete")$order]
# Run twice to determine a shared row and column clustering
# first
plot(1)

```

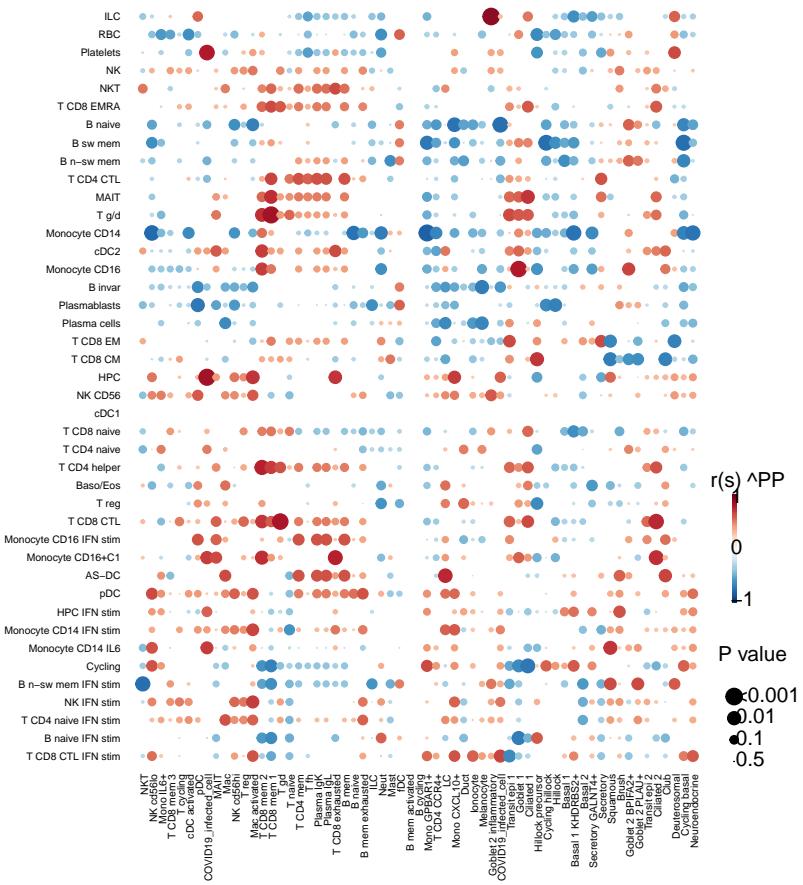


```

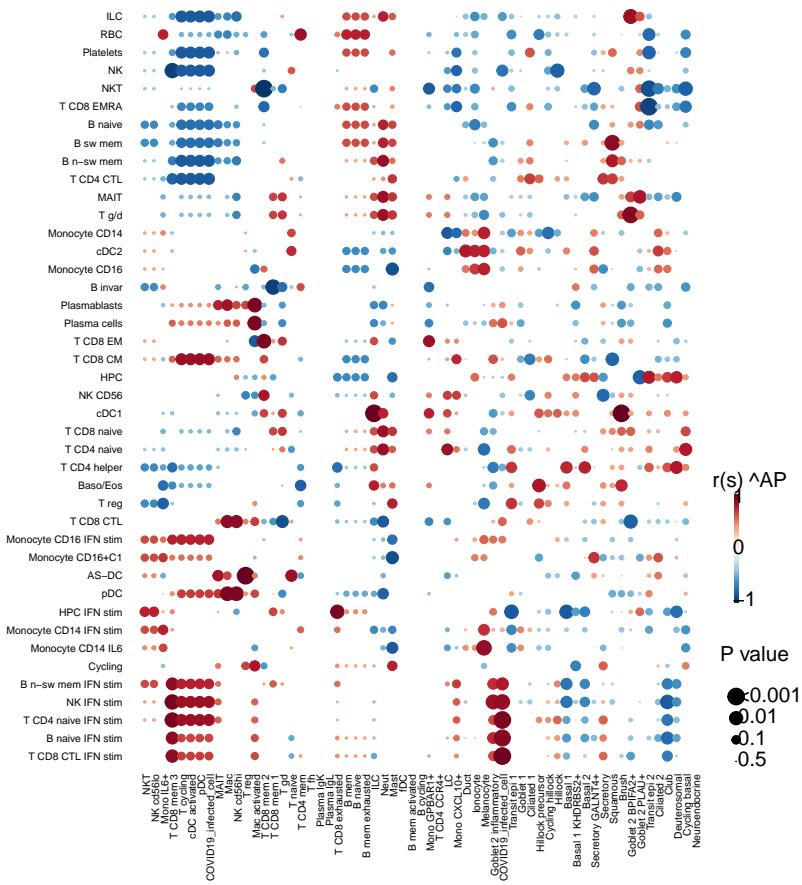
for (i in c("^PP", "^AP")) {
  # PP are paediatric covid samples and AP adult covid
  # samples
  spreadProbs_fil <- spreadProbs[grep(i, spreadProbs$id),
    !grep("Trachea", colnames(spreadProbs))]
  myCorTestVector <- suppressWarnings(psych::corr.test(spreadProbs_fil[,
    6:ncol(spreadProbs_fil)], adjust = "none", method = "spearman"))
  myCorVector_fil <- suppressWarnings(as.data.frame(cor(spreadProbs_fil[,
    6:ncol(spreadProbs_fil)], use = "pairwise.complete.obs",
    method = "spearman"), stringsAsFactors = F))
  myCorTestVector_fil <- as.data.frame(myCorTestVector$p)
  myCorTestVector_fil <- myCorTestVector_fil[grep("Blood_Immune",
    rownames(myCorTestVector_fil)), !grep("Blood", colnames(myCorTestVector_fil))]
  myCorVector_fil <- myCorVector_fil[grep("Blood_Immune",
    rownames(myCorVector_fil)), !grep("Blood", colnames(myCorVector_fil))]
  myCorVector_fil[is.na(myCorVector_fil)] <- 0
  myCorTestVector_fil[is.na(myCorTestVector_fil)] <- 1
  myCorVector_fil_list[[i]] <- myCorVector_fil
  myCorTestVector_fil_list[[i]] <- myCorTestVector_fil
  myCorVector_fil <- myCorVector_fil[myClust_row, myClust_col]
  myCorTestVector_fil <- myCorTestVector_fil[myClust_row, myClust_col]
  myCorVector_fil <- myCorVector_fil[, order(!grep("Immune",
    colnames(myCorVector_fil)))]
  myCorTestVector_fil <- myCorTestVector_fil[, order(!grep("Immune",
    colnames(myCorTestVector_fil)))]
  rownames(myCorVector_fil) <- gsub("_Blood_Immune", "", rownames(myCorVector_fil))
  colnames(myCorVector_fil) <- gsub("_Nose_(Epi|Immune)", "",
    colnames(myCorVector_fil))
  par(mar = c(6, 10, 2, 16))
  Dotplot_forCorHeatmap(myCorVector_fil, SORT = c(F, F), zlim = c(-1,
    1), ltsr = 1 - myCorTestVector_fil, cex = 0.8, measure = paste("r(s)",
    i), cex.axis = 0.5, srt = 90)
}

## Warning in sqrt(ltsr): NaNs produced

```



```
## Warning in sqrt(ltsr): NaNs produced
```



Visualize clonal diversity over age

```

vdjPresent <- cv$orig.ident %in% unique(cv$orig.ident[!is.na(cv$IR_VJ_1_cdr3_tcr)]) &
  cv$orig.ident %in% unique(cv$orig.ident[!is.na(cv$IR_VJ_1_cdr3_bcr)])
```

```

cv@meta.data$IR_cdr3_bcr <- paste(cv@meta.data$IR_VJ_1_cdr3_bcr,
  cv@meta.data$IR_VDJ_1_cdr3_bcr)
cv@meta.data$IR_cdr3_bcr[cv@meta.data$IR_cdr3_bcr == "NA NA"] <- NA
cv@meta.data$IR_cdr3_tcr <- paste(cv@meta.data$IR_VJ_1_cdr3_tcr,
  cv@meta.data$IR_VDJ_1_cdr3_tcr)
cv@meta.data$IR_cdr3_tcr[cv@meta.data$IR_cdr3_tcr == "NA NA"] <- NA
```

```

atLeast100TcrCells <- names(table(cv@meta.data$patient_id, !is.na(cv@meta.data$IR_cdr3_tcr))[, 
  "TRUE"])[table(cv@meta.data$patient_id, !is.na(cv@meta.data$IR_cdr3_tcr))[, 
  "TRUE"] >= 100]
atLeast100BcrCells <- names(table(cv@meta.data$patient_id, !is.na(cv@meta.data$IR_cdr3_bcr))[, 
  "TRUE"])[table(cv@meta.data$patient_id, !is.na(cv@meta.data$IR_cdr3_bcr))[, 
  "TRUE"] >= 100]
```

```

irCounts <- data.frame(stringsAsFactors = F, patient_id = unique(cv@meta.data$patient_id),
  ageGroup = sapply(unique(cv@meta.data$patient_id), function(x) unique(cv@meta.data$ageGroup[cv@meta
  x])), covidStatus = sapply(unique(cv@meta.data$patient_id),

```

```

function(x) unique(cv@meta.data$covid_status[cv@meta.data$patient_id == x]), nCells = sapply(unique(cv@meta.data$patient_id),
function(x) length(cv@meta.data$patient_id[cv@meta.data$patient_id == x])), nBcrExprCells = sapply(unique(cv@meta.data$patient_id),
function(x) length((cv@meta.data$IR_cdr3_bcr[cv@meta.data$patient_id == x] & !is.na(cv@meta.data$IR_cdr3_bcr)))), nTcrExprCells = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_tcr))), nTcrExprCells = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_tcr))), naiveUniqueBcrs = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_bcr) & grepl("naive", cv@meta.data$cell_annotation_revision_short))), naiveUniqueTcrs = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_tcr) & grepl("naive", cv@meta.data$cell_annotation_revision_short))), uniqueBcrs = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_bcr))), uniqueTcrs = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_tcr))), uniqueTcrs = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_tcr))))))

irCounts$patient_id_factor <- factor(irCounts$patient_id, levels = unique(cv$patient_id[order(vdjPresent, cv$age_year)]))
irCounts$ageGroup_factor <- factor(irCounts$ageGroup, levels = unique(cv$ageGroup[order(cv$age_year)]))

uniqueBcrs <- unique(cv@meta.data$IR_cdr3_bcr[!is.na(cv@meta.data$IR_cdr3_bcr)])
# clonalityBcrs <- sapply(uniqueBcrs, function(x)
# sum(cv@meta.data$IR_cdr3_bcr[!is.na(cv@meta.data$IR_cdr3_bcr)]==x))

cv@meta.data$IR_cdr3_dist_bcr <- NA
for (i in unique(cv@meta.data$patient_id[vdjPresent])) {
  tempBcrs <- uniqueBcrs[uniqueBcrs %in% cv@meta.data$IR_cdr3_bcr[cv@meta.data$patient_id == i]]
  tempBcrs_stripped <- gsub(" ", "", gsub("nan", "", tempBcrs))
  hammingDists <- stringdist::stringdistmatrix(tempBcrs_stripped,
    tempBcrs_stripped, method = "h", nthread = 4)
  hammingDists_norm <- sapply(1:nrow(hammingDists), function(x) hammingDists[x,
    ]/nchar(tempBcrs_stripped[x]))
  matchedBcrs <- apply(hammingDists_norm, 1, function(x) tempBcrs[x <=
    0.1] [1])
  names(matchedBcrs) <- tempBcrs
  cv@meta.data$IR_cdr3_dist_bcr[cv@meta.data$patient_id == i & !is.na(cv@meta.data$IR_cdr3_bcr)] <- sapply(cv@meta.data$IR_cdr3_bcr[cv@meta.data$patient_id == i & !is.na(cv@meta.data$IR_cdr3_bcr)], function(x) matchedBcrs[x])
}

irCounts <- data.frame(stringsAsFactors = F, patient_id = unique(cv@meta.data$patient_id),
  ageGroup = sapply(unique(cv@meta.data$patient_id), function(x) unique(cv@meta.data$ageGroup[cv@meta.data$patient_id == x])),
  covidStatus = sapply(unique(cv@meta.data$patient_id),
  function(x) unique(cv@meta.data$covid_status[cv@meta.data$patient_id == x])), nCells = sapply(unique(cv@meta.data$patient_id),
  function(x) length(cv@meta.data$patient_id[cv@meta.data$patient_id == x])), nBcrExprCells = sapply(unique(cv@meta.data$patient_id),
  function(x) length((cv@meta.data$IR_cdr3_dist_bcr[cv@meta.data$patient_id == x] & !is.na(cv@meta.data$IR_cdr3_dist_bcr)))), nTcrExprCells = sapply(unique(cv@meta.data$patient_id == x & !is.na(cv@meta.data$IR_cdr3_dist_bcr))))

```

```

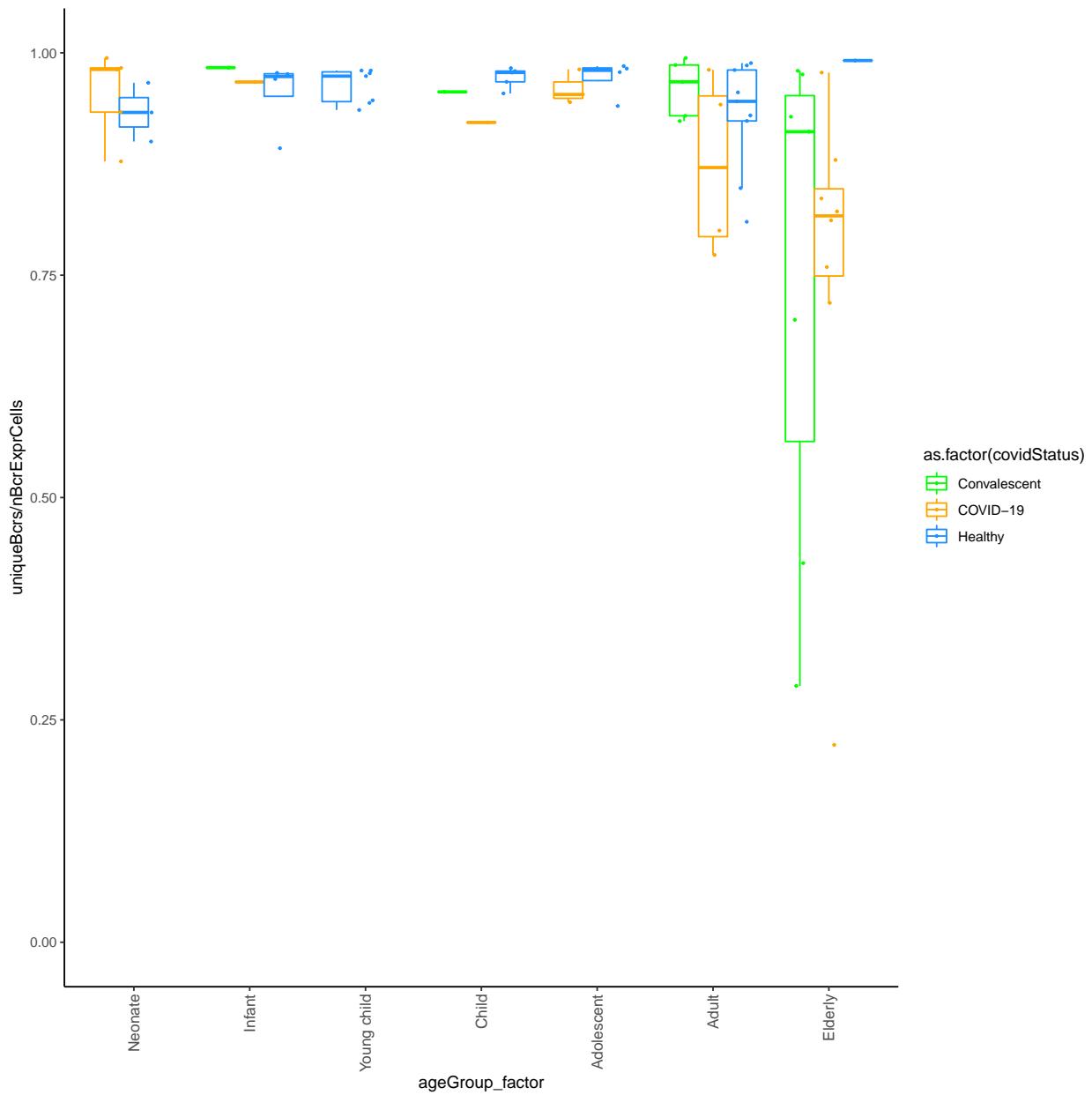
function(x) length((cv@meta.data$IR_cdr3_tcr[cv@meta.data$patient_id ==
x & !is.na(cv@meta.data$IR_cdr3_tcr)])), naiveUniqueBcrs = sapply(unique(cv@meta.data$patient_id ==
function(x) length(unique(cv@meta.data$IR_cdr3_dist_bcr[cv@meta.data$patient_id ==
x & !is.na(cv@meta.data$IR_cdr3_dist_bcr) & grepl("naive",
cv@meta.data$cell_annotation_revision_short)])), naiveUniqueTcrs = sapply(unique(cv@meta.data$patient_id ==
function(x) length(unique(cv@meta.data$IR_cdr3_tcr[cv@meta.data$patient_id ==
x & !is.na(cv@meta.data$IR_cdr3_tcr) & grepl("naive",
cv@meta.data$cell_annotation_revision_short)])), uniqueBcrs = sapply(unique(cv@meta.data$patient_id ==
function(x) length(unique(cv@meta.data$IR_cdr3_dist_bcr[cv@meta.data$patient_id ==
x & !is.na(cv@meta.data$IR_cdr3_dist_bcr)])), uniqueTcrs = sapply(unique(cv@meta.data$patient_id ==
function(x) length(unique(cv@meta.data$IR_cdr3_tcr[cv@meta.data$patient_id ==
x & !is.na(cv@meta.data$IR_cdr3_tcr)]))

irCounts$patient_id_factor <- factor(irCounts$patient_id, levels = unique(cv$patient_id[order(vdjPresent ==
cv$age_year)]))

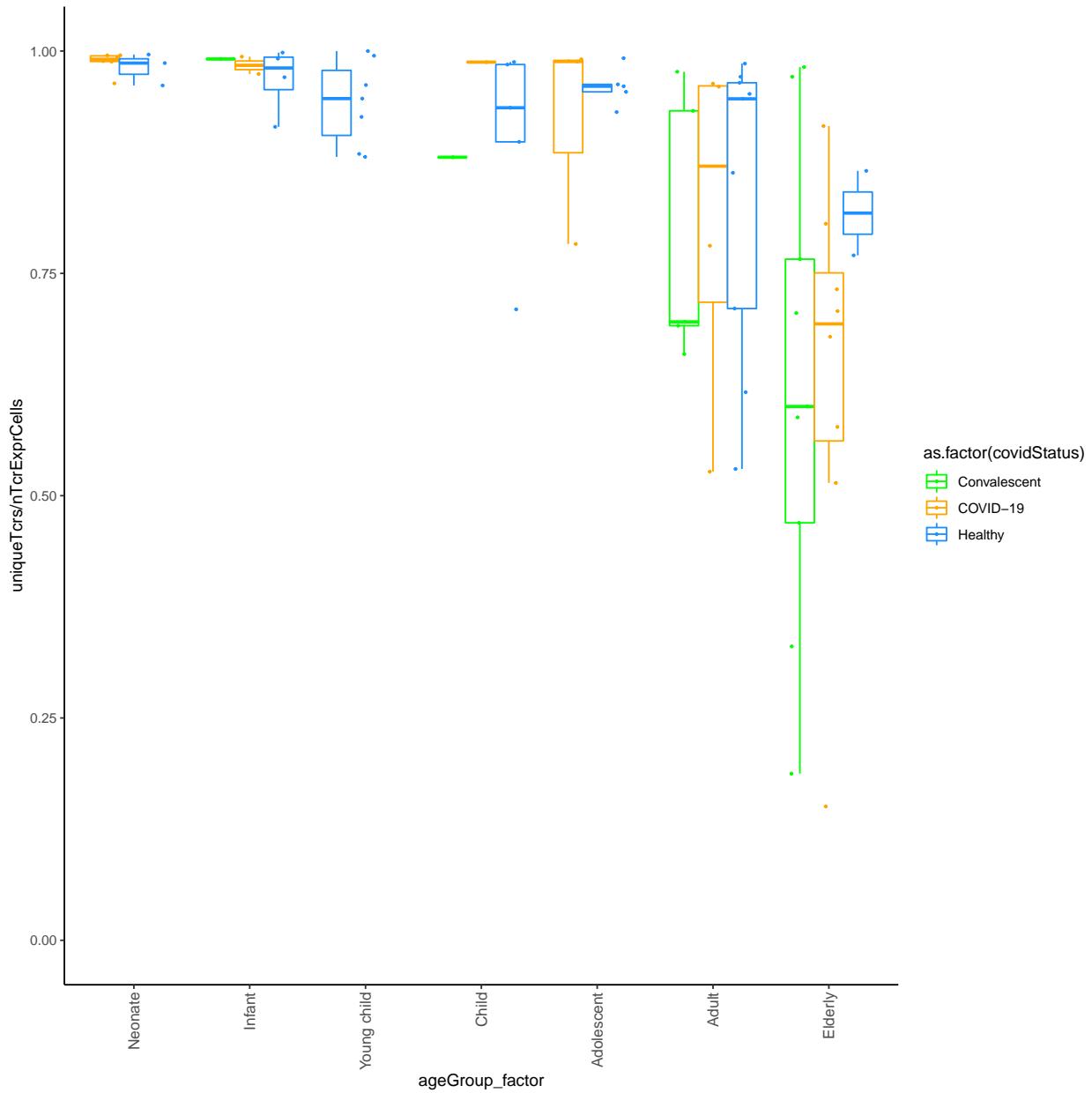
irCounts$ageGroup_factor <- factor(irCounts$ageGroup, levels = unique(cv$ageGroup[order(cv$age_year)]))

ggplot(irCounts[irCounts$patient_id %in% unique(cv$patient_id[vdjPresent]) &
!is.na(irCounts$ageGroup) & irCounts$patient_id %in% atLeast100BcrCells,
], aes(ageGroup_factor, uniqueBcrs/nBcrExprCells)) + geom_boxplot(aes(col = as.factor(covidStatus)),
outlier.shape = NA, position = position_dodge(preserve = "single")) +
coord_cartesian(ylim = c(0, 1)) + geom_point(aes(col = as.factor(covidStatus)),
position = position_jitterdodge(), size = 0.5) + scale_color_manual(values = c("green",
"orange", "dodgerblue")) + theme_classic() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust = 1, size = 10))

```



```
ggplot(irCounts[irCounts$patient_id %in% unique(cv$patient_id[vdjPresent]) &
  !is.na(irCounts$ageGroup) & irCounts$patient_id %in% atLeast100TcrCells,
  ], aes(ageGroup_factor, uniqueTcrs/nTcrExprCells)) + geom_boxplot(aes(col = as.factor(covidStatus)),
outlier.shape = NA, position = position_dodge(preserve = "single")) +
coord_cartesian(ylim = c(0, 1)) + geom_point(aes(col = as.factor(covidStatus)),
position = position_jitterdodge(), size = 0.5) + scale_color_manual(values = c("green",
"orange", "dodgerblue")) + theme_classic() + theme(axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust = 1, size = 10))
```



UMAPs using RNA only or RNA+ADT reductions to show annotation and patient info in umap space

```
# Define good color scheme for annots
colsForNewAnnot <- randomcoloR::distinctColorPalette(length(unique(cv@meta.data$cell_annot_revision_short)) + 1)
names(colsForNewAnnot) <- levels(cv@meta.data$cell_annot_revision_short)
colsForNewAnnot <- colsForNewAnnot[!is.na(names(colsForNewAnnot))]

# colsForBroadAnnot <-
# randomcoloR::distinctColorPalette(length(unique(cv@meta.data$cell_annot_revision_broad))), runTsne = T)
# names(colsForBroadAnnot) <- levels(cv@meta.data$cell_annot_revision_broad)
# write_rds(colsForNewAnnot, file='/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot3.rds')
```

```

# write_rds(colsForBroadAnnot,file='/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForBroadAnno

# 1 Broad annot
colsForNewAnnot <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot2.rds")
colsForNewAnnot["B"] <- "#B7E9F7"
colsForNewAnnot["Plasma"] <- "#80471C"
colsForNewAnnot["T CD8+"] <- "#efb261"
colsForNewAnnot["T CD4+"] <- "#f699cd"
colsForNewAnnot["DC"] <- "#A6BEB2"
colsForNewAnnot["Monocyte"] <- "#99fadc"

cv$current_annot <- as.character(cv$cell_annot_revision_broad)

colsForNewAnnot <- colsForNewAnnot[names(colsForNewAnnot) %in%
  cv$current_annot]

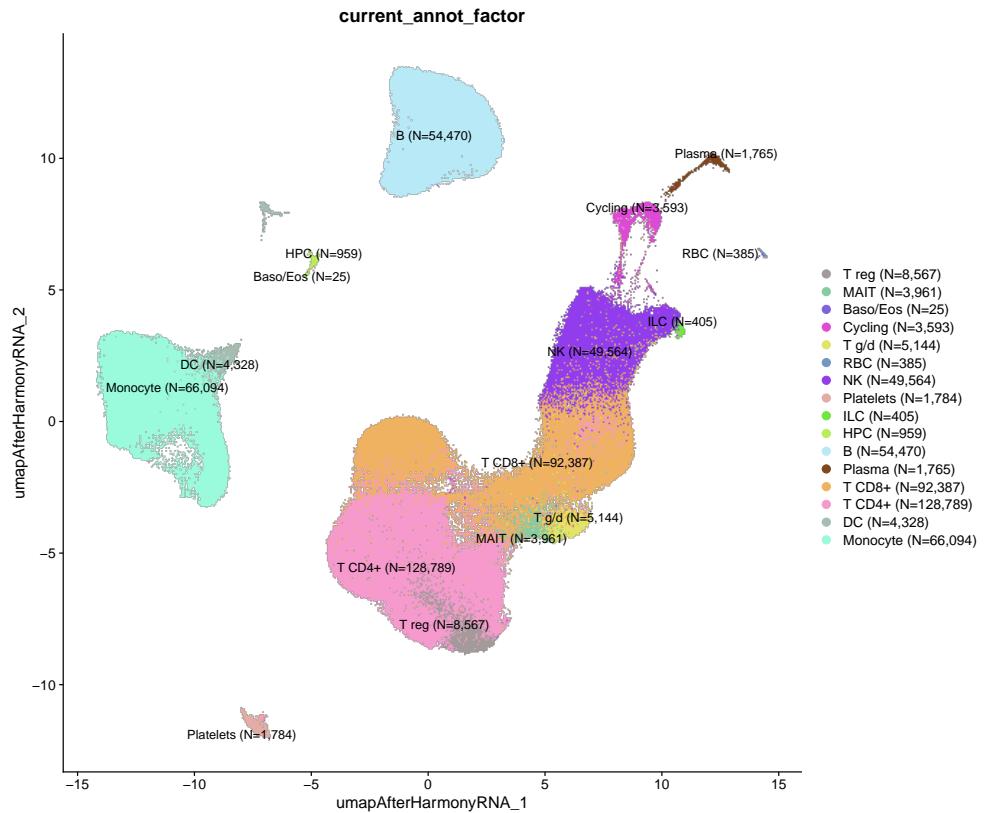
for (i in unique(cv$current_annot)) {
  cv$current_annot[cv$current_annot == i] <- paste0(i, " (N=",
    scales::label_comma()(length(cv$current_annot[cv$current_annot ==
      i])), ")")
}
cv$current_annot_factor <- factor(cv$current_annot, levels = cv$current_annot[!duplicated(cv$current_annot))

for (i in 1:length(names(colsForNewAnnot))) {
  names(colsForNewAnnot)[i] <- unique(cv$current_annot[cv$cell_annot_revision_broad ==
    names(colsForNewAnnot)[i]])
}

(DimPlot(cv, reduction = "umapAfterHarmony_RNA", group.by = "current_annot_factor",
  shuffle = T, label = T, raster = T, repel = T, pt.size = 0.1,
  cols = colsForNewAnnot) + theme(aspect.ratio = 1))

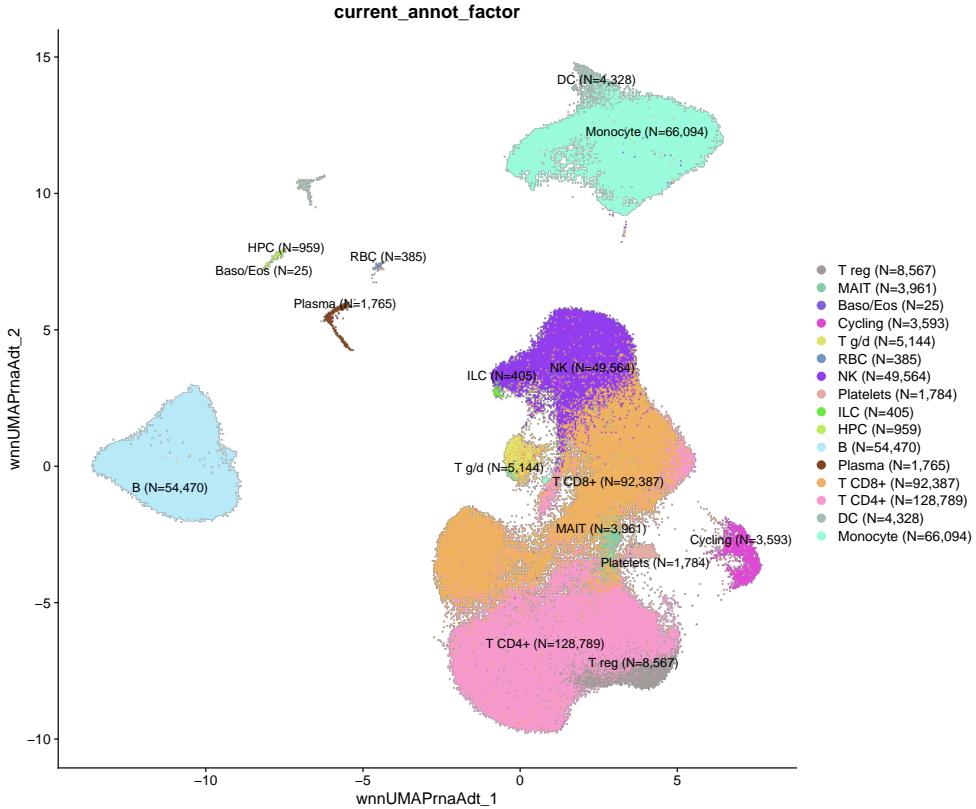
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

```



```
(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "current_annot_factor",
        shuffle = T, label = T, raster = T, repel = T, pt.size = 0.1,
        cols = colsForNewAnnot) + theme(aspect.ratio = 1))
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



```

# 3 Refined annot
myCounter <- 1
for (i in unique(cv$labelOrder[order(cv$labelOrder)])) {
  cv$labelOrder[cv$labelOrder == i] <- myCounter
  myCounter <- myCounter + 1
}
cv$labelOrder_factor <- factor(cv$labelOrder, levels = unique(cv$labelOrder)[order(unique(cv$labelOrder))])

colsForNewAnnot <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot2.rds")

cv$current_annot <- as.character(cv$cell_annot_revision_short)

cv$current_annot[grep("Monocyte", cv$current_annot)] <- gsub("Monocyte",
  "Mono", cv$current_annot[grep("Monocyte", cv$current_annot)])
for (i in unique(cv$current_annot)) {
  cv$current_annot[cv$current_annot == i] <- paste0(i, " (N=",
    scales::label_comma()(length(cv$current_annot[cv$current_annot ==
      i])), ")")
}
cv$current_annot_factor <- factor(cv$current_annot, levels = cv$current_annot[!duplicated(cv$labelOrder)])
colsForNewAnnot_clusterNumber <- colsForNewAnnot
for (i in 1:length(names(colsForNewAnnot))) {
  names(colsForNewAnnot_clusterNumber)[i] <- unique(cv$labelOrder[cv$cell_annot_revision_short ==
    names(colsForNewAnnot)[i]])
  names(colsForNewAnnot)[i] <- unique(cv$current_annot[cv$cell_annot_revision_short ==
    names(colsForNewAnnot)[i]])
}

```

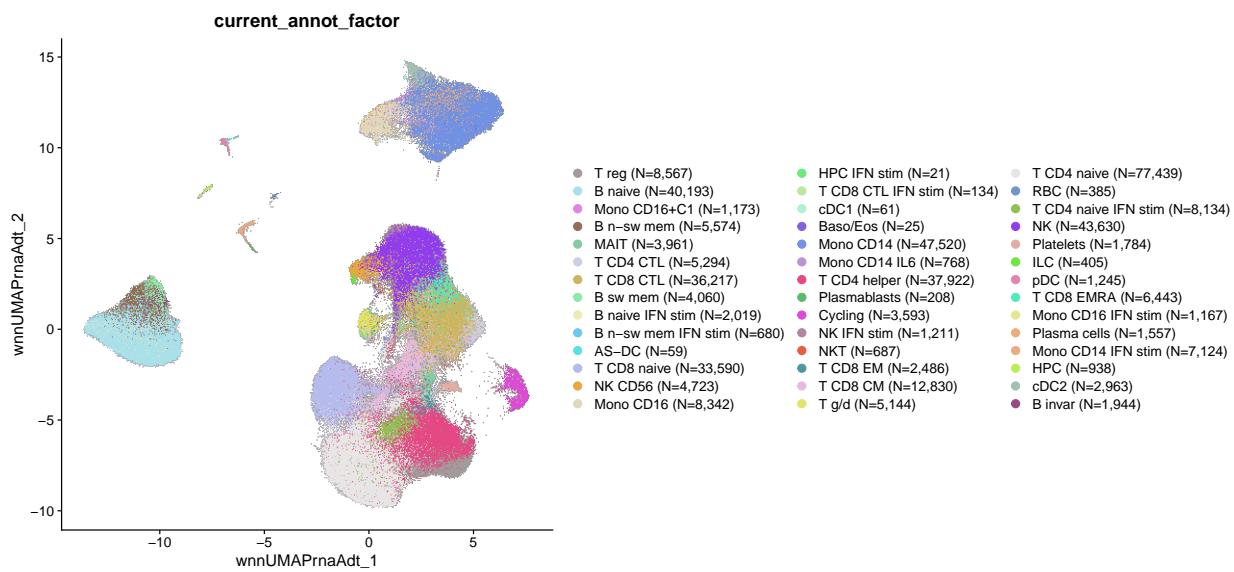
```

# (DimPlot(cv, reduction='umapAfterHarmony_RNA', group.by='current_annot_factor', shuffle
# = T, label=F, raster=T, repel=T, pt.size =
# .1, cols=colsForNewAnnot) + theme(aspect.ratio = 1))
# (DimPlot(cv, reduction='umapAfterHarmony_RNA', group.by='labelOrder_factor', shuffle
# = T, label=T, raster=F, repel=F, pt.size =
# NA, cols=colsForNewAnnot_clusterNumber) +
# theme(aspect.ratio = 1))

(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "current_annot_factor",
  shuffle = T, label = F, raster = T, repel = T, pt.size = 0.1,
  cols = colsForNewAnnot) + theme(aspect.ratio = 1))

```

Rasterizing points since number of points exceeds 100,000.
 ## To disable this behavior set 'raster=FALSE'



```

# (DimPlot(cv, reduction='wnn.umap_rnaAdt', group.by='labelOrder_factor', shuffle
# = T, label=T, raster=F, repel=F, pt.size =
# NA, cols=colsForNewAnnot_clusterNumber) +
# theme(aspect.ratio = 1))

# 3 Refined annot + age/covid colours
covidCols <- c("dodgerblue", "orange", "green")
names(covidCols) <- c("Healthy", "COVID-19", "Convalescent")

```

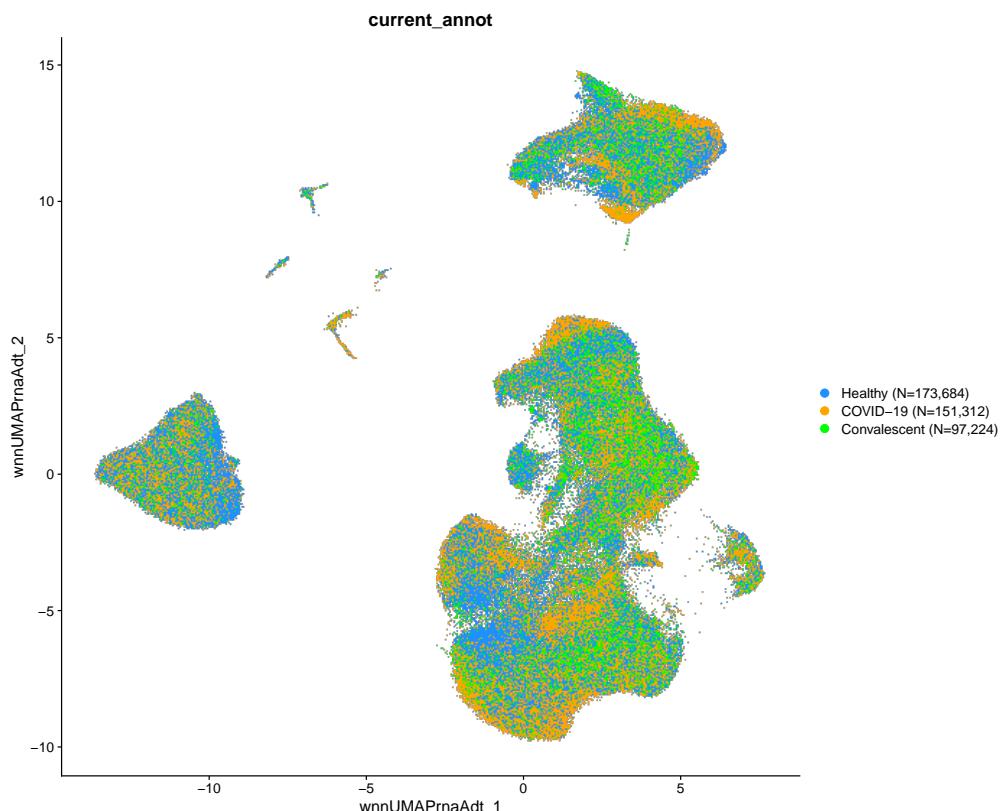
```

cv$current_annot <- cv$covid_status
for (i in unique(cv$current_annot)) {
  cv$current_annot[cv$current_annot == i] <- paste0(i, " (N=",
    scales::label_comma()(length(cv$current_annot[cv$current_annot ==
      i])), ")")
}
myCols_wN <- covidCols
for (i in names(myCols_wN)) {
  names(myCols_wN)[names(myCols_wN) == i] <- unique(cv$current_annot[grep1(paste0(i,
    " \\\\"(N="), cv$current_annot)])
}
cv$current_annot <- factor(cv$current_annot, levels = names(myCols_wN))

(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "current_annot",
  cols = myCols_wN, shuffle = T, label = F, repel = T, raster = T,
  pt.size = 0.1) + theme(aspect.ratio = 1))

```

Rasterizing points since number of points exceeds 100,000.
 ## To disable this behavior set 'raster=FALSE'

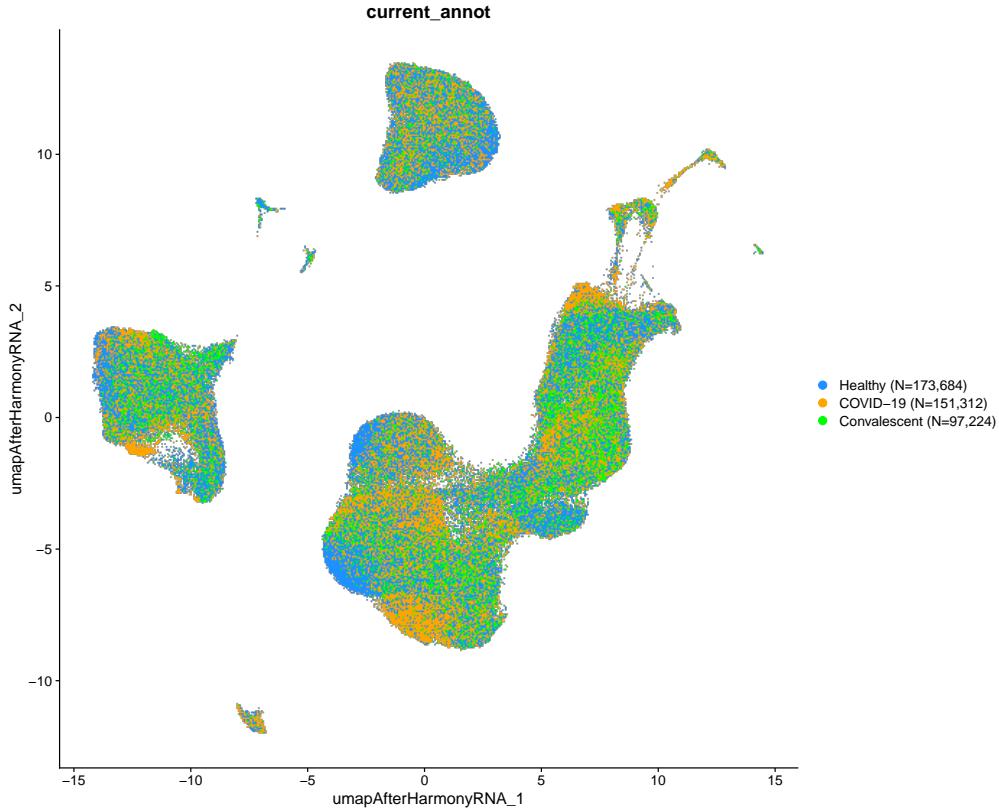


```

(DimPlot(cv, reduction = "umapAfterHarmony_RNA", group.by = "current_annot",
  cols = myCols_wN, shuffle = T, label = F, repel = T, raster = T,
  pt.size = 0.1) + theme(aspect.ratio = 1))

```

Rasterizing points since number of points exceeds 100,000.
 ## To disable this behavior set 'raster=FALSE'



```

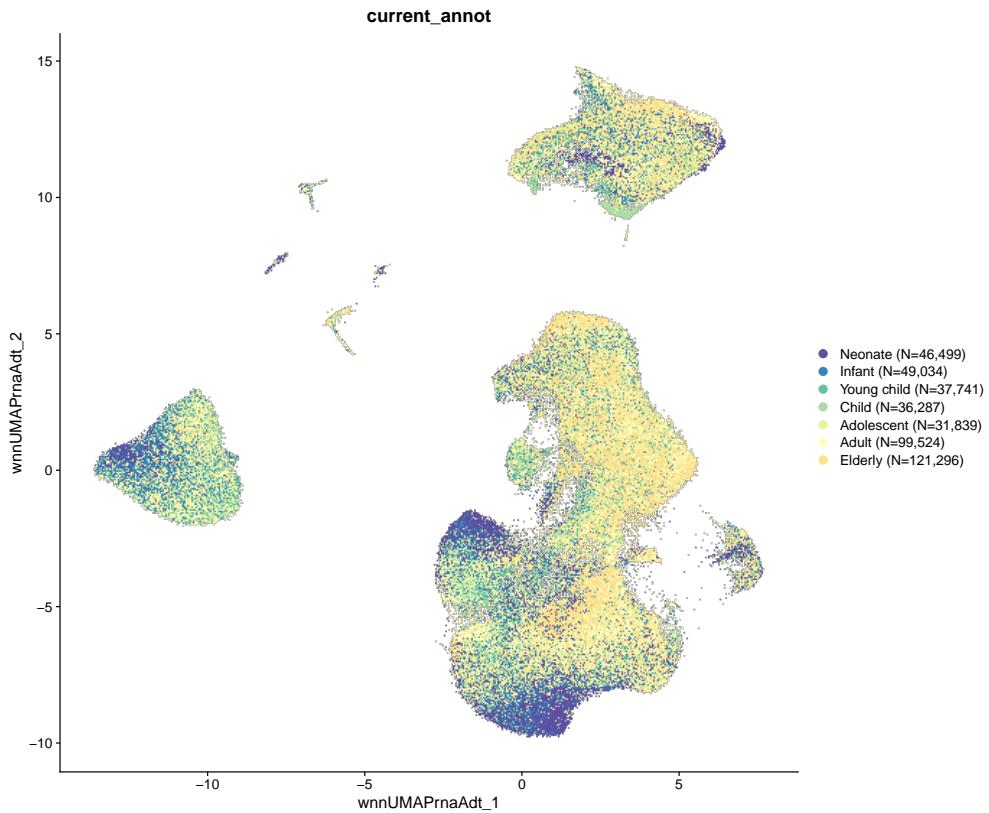
ageCols <- rev(RColorBrewer::brewer.pal(11, "Spectral") [c(5:11)])
names(ageCols) <- c("Neonate", "Infant", "Young child", "Child",
    "Adolescent", "Adult", "Elderly")

cv$current_annot <- cv$ageGroup
for (i in unique(cv$current_annot)) {
    cv$current_annot[cv$current_annot == i] <- paste0(i, " (N=",
        scales::label_comma()(length(cv$current_annot[cv$current_annot ==
            i])), ")")
}
myCols_wN <- ageCols
for (i in names(myCols_wN)) {
    names(myCols_wN)[names(myCols_wN) == i] <- unique(cv$current_annot[grep1(paste0(i,
        " \\\\" (N="), cv$current_annot))]
}
cv$current_annot <- factor(cv$current_annot, levels = names(myCols_wN))

(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "current_annot",
    cols = myCols_wN, shuffle = T, label = F, repel = T, raster = T,
    pt.size = 0.1) + theme(aspect.ratio = 1))

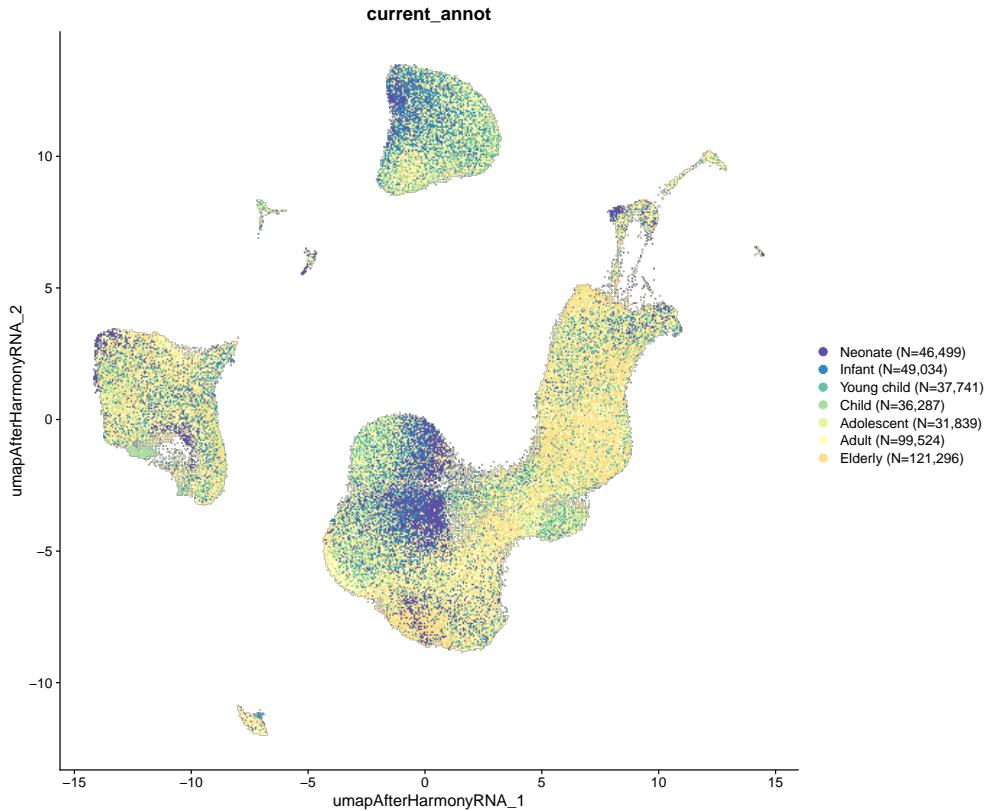
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

```



```
(DimPlot(cv, reduction = "umapAfterHarmony_RNA", group.by = "current_annot",
  cols = myCols_wN, shuffle = T, label = F, repel = T, raster = T,
  pt.size = 0.1) + theme(aspect.ratio = 1))
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



```
# 4 IFN stim annot

colsForNewAnnot <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot2.rds")

cv$current_annot <- as.character(cv$cell_annot_revision_short_woIfnStim)
colsForNewAnnot <- colsForNewAnnot[names(colsForNewAnnot) %in%
  cv$current_annot]

cv$current_annot[grep("Monocyte", cv$current_annot)] <- gsub("Monocyte",
  "Mono", cv$current_annot[grep("Monocyte", cv$current_annot)])
for (i in unique(cv$current_annot)) {
  cv$current_annot[cv$current_annot == i] <- paste0(i, " (N=",
    scales::label_comma()(length(cv$current_annot[cv$current_annot ==
      i])), ")")
}
cv$current_annot_factor <- factor(cv$current_annot, levels = cv$current_annot[!duplicated(cv$current_annot)])
colsForNewAnnot_clusterNumber <- colsForNewAnnot
for (i in 1:length(names(colsForNewAnnot))) {
  names(colsForNewAnnot_clusterNumber)[i] <- min(cv$labelOrder[cv$cell_annot_revision_short_woIfnStim
    names(colsForNewAnnot)[i]])
  names(colsForNewAnnot)[i] <- unique(cv$current_annot[cv$cell_annot_revision_short_woIfnStim ==
    names(colsForNewAnnot)[i]])
}

# (DimPlot(cv,reduction='umapAfterHarmony_RNA',group.by='current_annot_factor',shuffle
# = T,label=F,raster=T,repel=T, pt.size =
# .1,cols=colsForNewAnnot) + theme(aspect.ratio = 1))
```

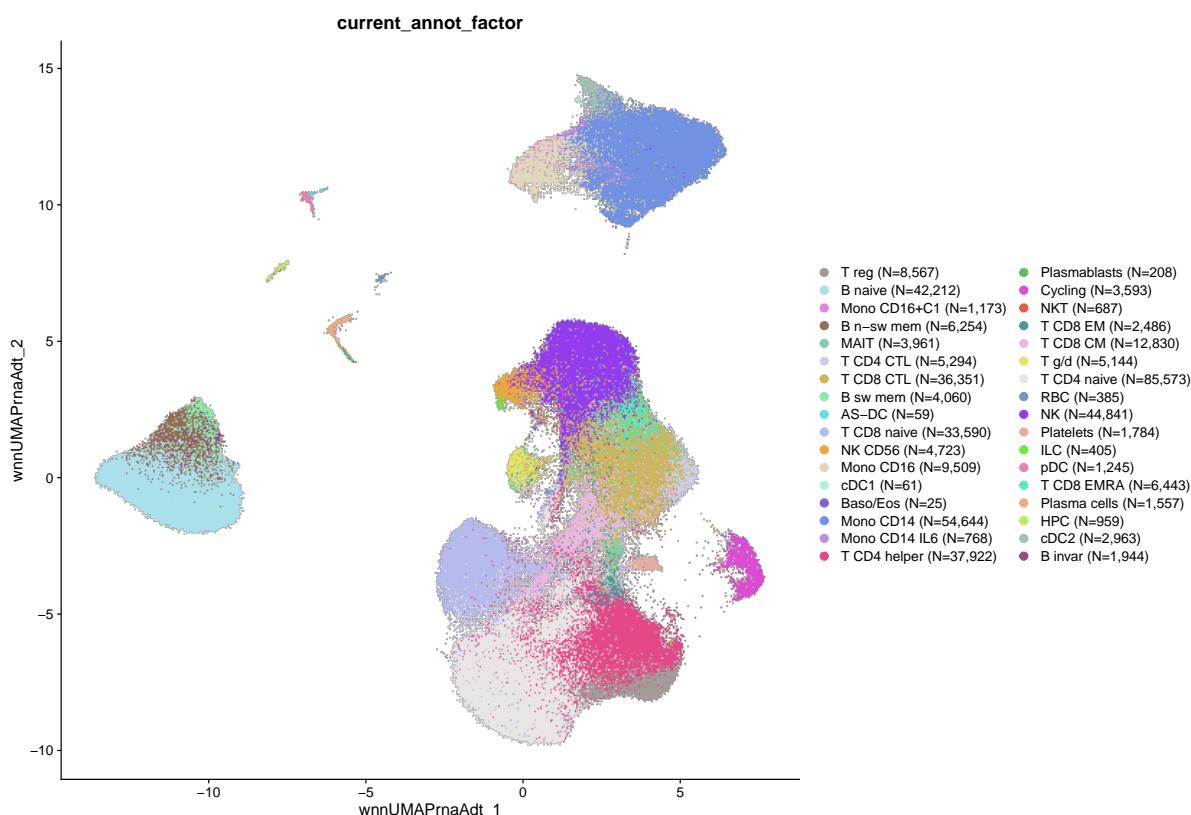
```

# (DimPlot(subset(cv, cells=rownames(cv@meta.data)[!grepl('IFN', cv$cell_annot_revision_short)]), reduction =
# = T, label=T, raster=F, repel=F, pt.size =
# NA, cols=colsForNewAnnot_clusterNumber) +
# theme(aspect.ratio = 1))

(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "current_annot_factor",
    shuffle = T, label = F, raster = T, repel = T, pt.size = 0.1,
    cols = colsForNewAnnot) + theme(aspect.ratio = 1))

```

Rasterizing points since number of points exceeds 100,000.
To disable this behavior set 'raster=FALSE'



```

# (DimPlot(subset(cv, cells=rownames(cv@meta.data)[!grepl('IFN', cv$cell_annot_revision_short)]), reduction =
# = T, label=T, raster=F, repel=F, pt.size =
# NA, cols=colsForNewAnnot_clusterNumber) +
# theme(aspect.ratio = 1))

```

Show cell type marker expression

```

# 5 GEX only plots
stimGenes <- rev(c("IRF7", "XAF1", "UBE2L6", "TRIM22", "STAT1",
    "SP110", "SAMD9L", "SAMD9", "PLSCR1", "PARP9", "OAS2", "OAS1",
    "MX2", "MX1", "LY6E", "ISG15", "IFIT3", "IFI6", "IFI44L",
    "IFI35", "HERC5", "EPSTI1", "EIF2AK2", "CMPK2", "BST2"))
# markersGex <-

```

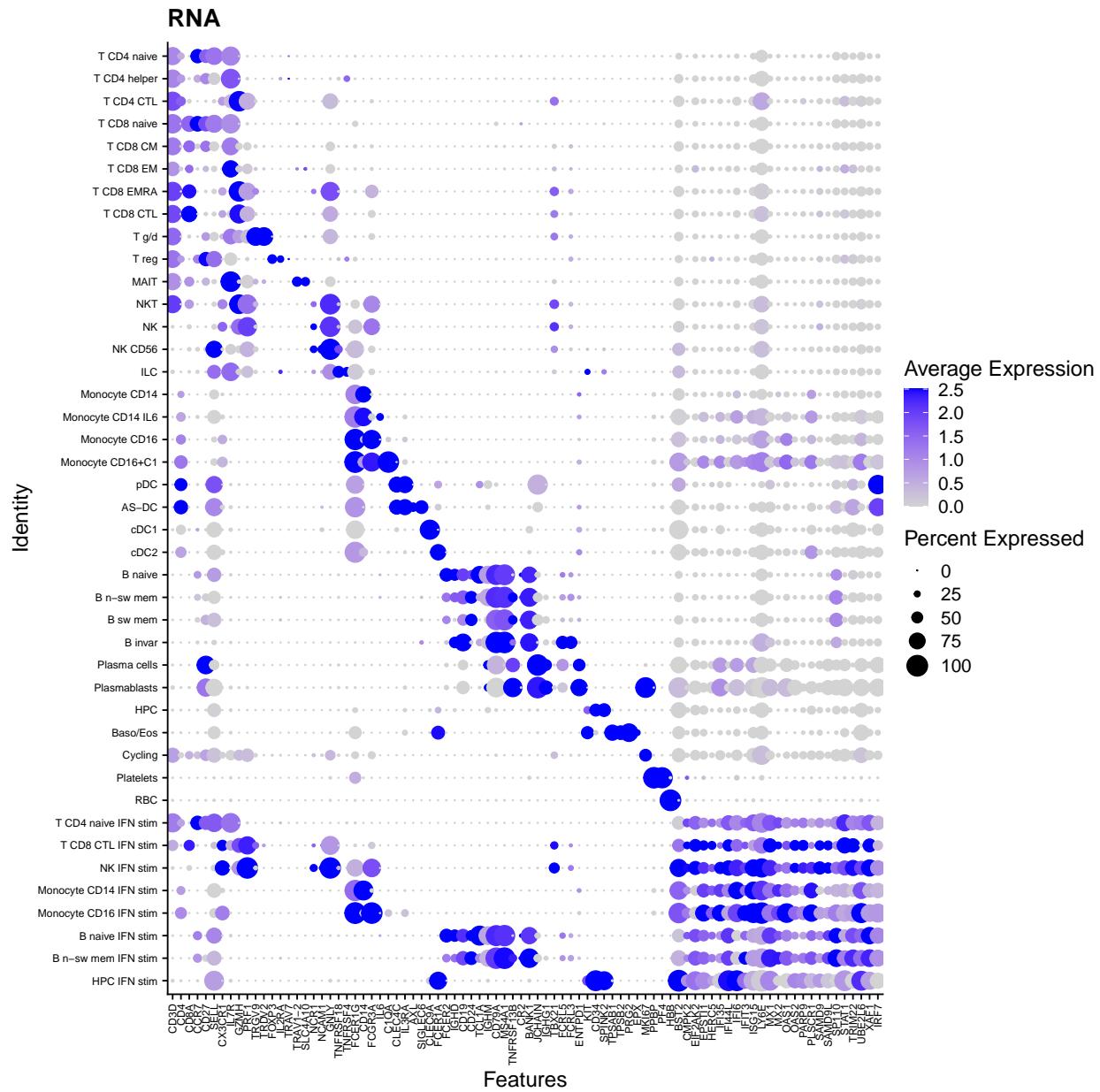
```

# unique(c('CD3D', 'CD4', 'CD8A', 'CCR7', 'CD27', 'SELL',
# 'CX3CR1',
# 'IL7R', 'PTPRC-2', 'PTPRC-3', 'GZMH', 'PRF1', 'TRGV9', 'TRDV2', 'FOXP3', 'IL2RA', 'TRAV1-2', 'SLC4A10', 'NCR1', '),
# markersGex <- unique(c("CD3D", "CD4", "CD8A", "CCR7", "CD27",
# "SELL", "CX3CR1", "IL7R", "PTPRC-2", "PTPRC-3", "GZMH", "PRF1",
# "TRGV9", "TRDV2", "FOXP3", "IL2RA", "TRAV7", "TRAV1-2", "SLC4A10",
# "NCR1", "NCAM1", "GNLY", "TNFRSF18", "TNFRSF4", "FCER1G",
# "CD14", "FCGR3A", "IL6", "C1QA", "CLEC4C", "IL3RA", "AXL",
# "SIGLEC6", "CLEC9A", "FCER1A", "FCER2", "IGHD", "CD19", "CD24",
# "TCL1A", "IGHM", "CD79A", "MS4A1", "TNFRSF13B", "CR2", "BANK1",
# "JCHAIN", "IGHG1", "TNFRSF13B", "TBX21", "FCRL5", "FCRL3",
# "ENTPD1", "KIT", "CD34", "SPINK2", "TPSAB1", "TPSB2", "PRG2",
# "EPX", "MKI67", "PPBP", "PF4", "HBB", stimGenes))
markersAdt <- paste0("AB-", markersGex)

markersGex <- markersGex[markersGex %in% rownames(cv[["RNA"]])]
markersAdt <- markersAdt[markersAdt %in% rownames(cv[["ADT"]])]

(DotPlot(cv, group.by = "cell_annot_revision_short", features = markersGex,
  cluster.idents = F, assay = "RNA", col.min = 0, cols = c("lightgrey",
  "blue")) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("RNA"))

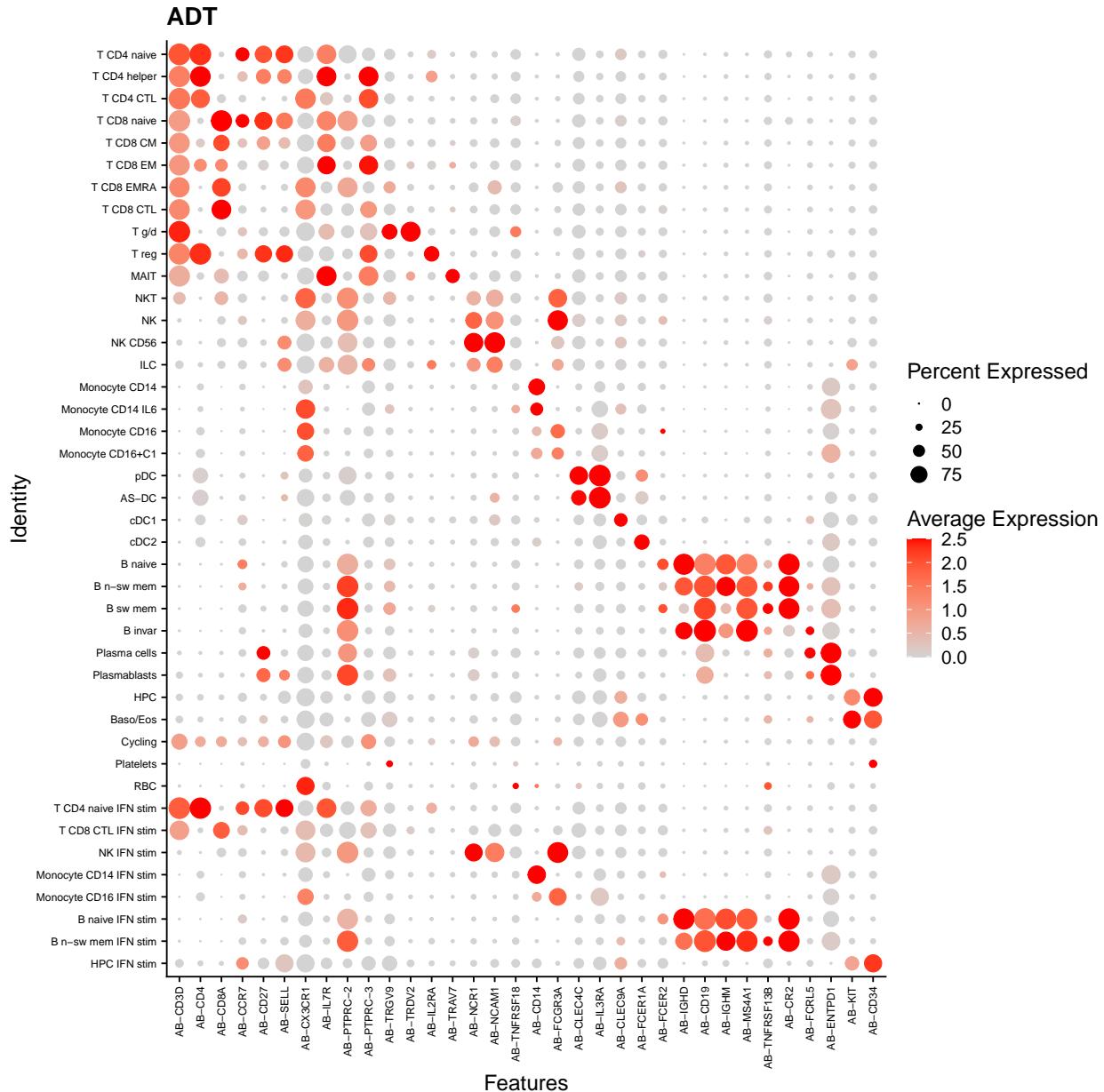
```



```

markersAdt <- markersAdt[!markersAdt %in% c("AB-CD24", "AB-IGHG1",
                                             "TNFRSF18")]
(DotPlot(cv, group.by = "cell_annotation_revision_short", features = markersAdt,
        cluster.idents = F, assay = "ADT", col.min = 0, cols = c("lightgrey",
                                                               "red")) + theme(axis.text.x = element_text(angle = 90,
                                                               hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)) +
  ggtitle("ADT"))

```



Plot the relative contribution of the protein data within each cell type by quantifying the weight in the shared nn graph

```
# 6 Violin plot with adt weight over cell types
colsForNewAnnot <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot2.rds")

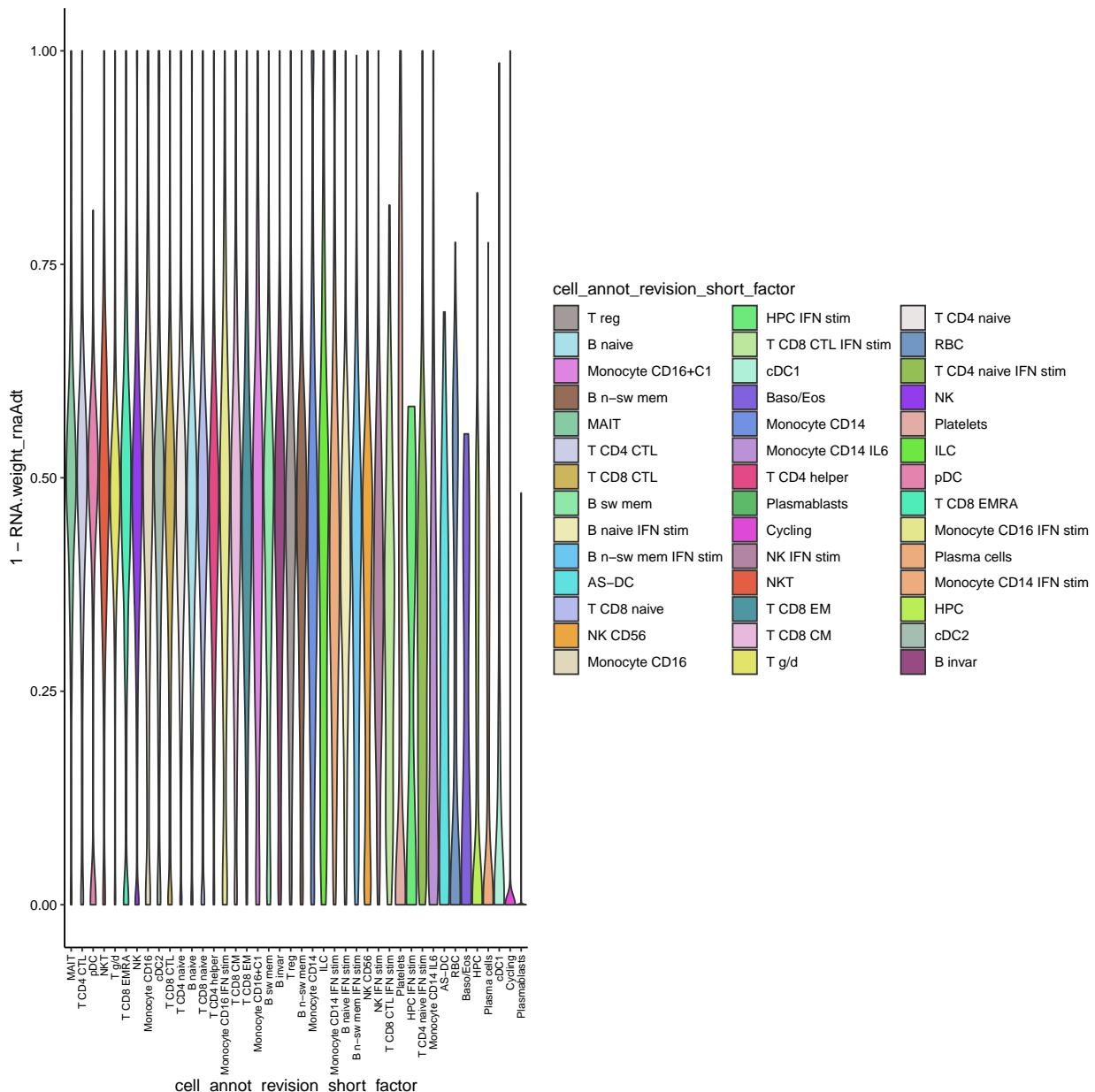
# VlnPlot(cv, features = 'RNA.weight_rnaAdt', group.by =
# 'cell_annot_revision_short', cols=colsForNewAnnot, sort =
# TRUE, pt.size = NULL) +theme(axis.text.x =
# element_text(angle = 90,hjust=1,vjust=0.5,size = 7))

wDat <- FetchData(cv, c("RNA.weight_rnaAdt", "cell_annot_revision_short"))
modW <- wDat %>%
  group_by(cell_annot_revision_short)
myMedians <- modW %>%
```

```

summarise(medianRnaModalityWeight = median(RNA.weight_rnaAdt))
modW$cell_annot_revision_short_factor <- factor(modW$cell_annot_revision_short,
  levels = myMedians$cell_annot_revision_short[order(myMedians$medianRnaModalityWeight)])
ggplot(modW, aes(cell_annot_revision_short_factor, 1 - RNA.weight_rnaAdt,
  fill = cell_annot_revision_short_factor)) + geom_violin(scale = "width") +
  scale_fill_manual(values = colsForNewAnnot) + theme_classic() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5,
    size = 7, colour = "black"), axis.text.y = element_text(colour = "black"))

```



Compare our own annotation to a publicly available annotation tool called Azimuth

```

# 14 Azimuth comparison
azi <- read.csv("/mnt/projects/RL003_allCitePbmcsTheta/azimuth_pred.tsv",
  header = T, stringsAsFactors = F, sep = "\t")

```

```

colnames(azi) <- paste0(colnames(azi), "_azimuth")
aziCp <- cv@meta.data[azi$cell_azimuth[azi$cell_azimuth %in%
  rownames(cv@meta.data)], ]
aziCp[, colnames(azi)] <- azi[azi$cell_azimuth %in% rownames(cv@meta.data),
  ]

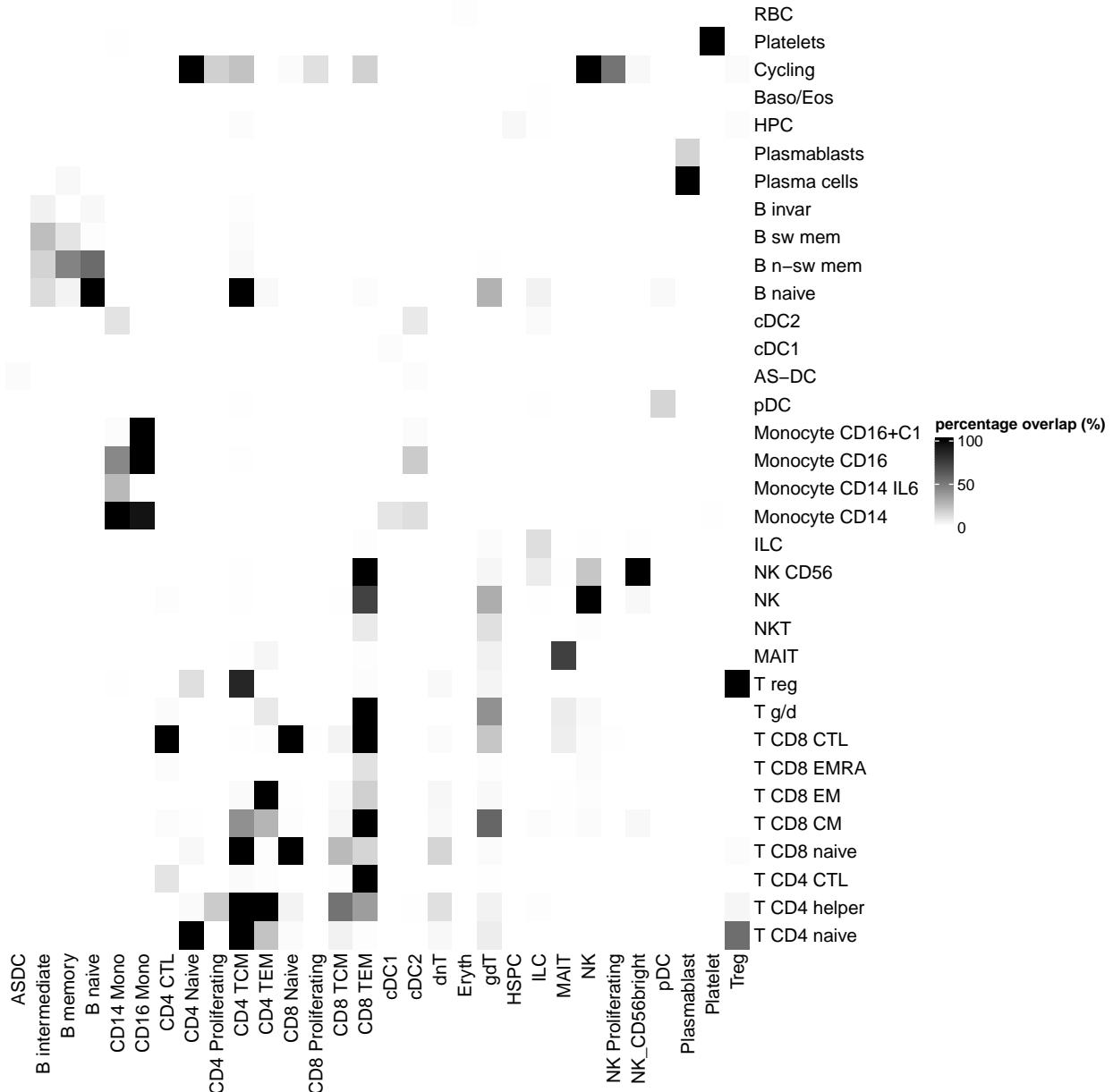
matchTable <- as.data.frame.matrix(table(aziCp$cell_annot_revision_short_woIfnStim,
  aziCp$predicted.id_azimuth)/rowSums(table(aziCp$cell_annot_revision_short_woIfnStim,
  aziCp$predicted.id_azimuth)))

Heatmap(as.data.frame.matrix(table(aziCp$cell_annot_revision_short_woIfnStim,
  aziCp$predicted.id_azimuth)/rowSums(table(aziCp$new_annot_rik,
  aziCp$predicted.id_azimuth))) * 100, cluster_rows = F, cluster_columns = F,
  col = circlize::colorRamp2(c(0, 100), c("white", "black")),
  name = "percentage overlap (%)")

## Warning in table(aziCp$cell_annot_revision_short_woIfnStim,
## aziCp$predicted.id_azimuth)/rowSums(table(aziCp$new_annot_rik, : longer object
## length is not a multiple of shorter object length

## Warning: The input is a data frame, convert it to the matrix.

```



```

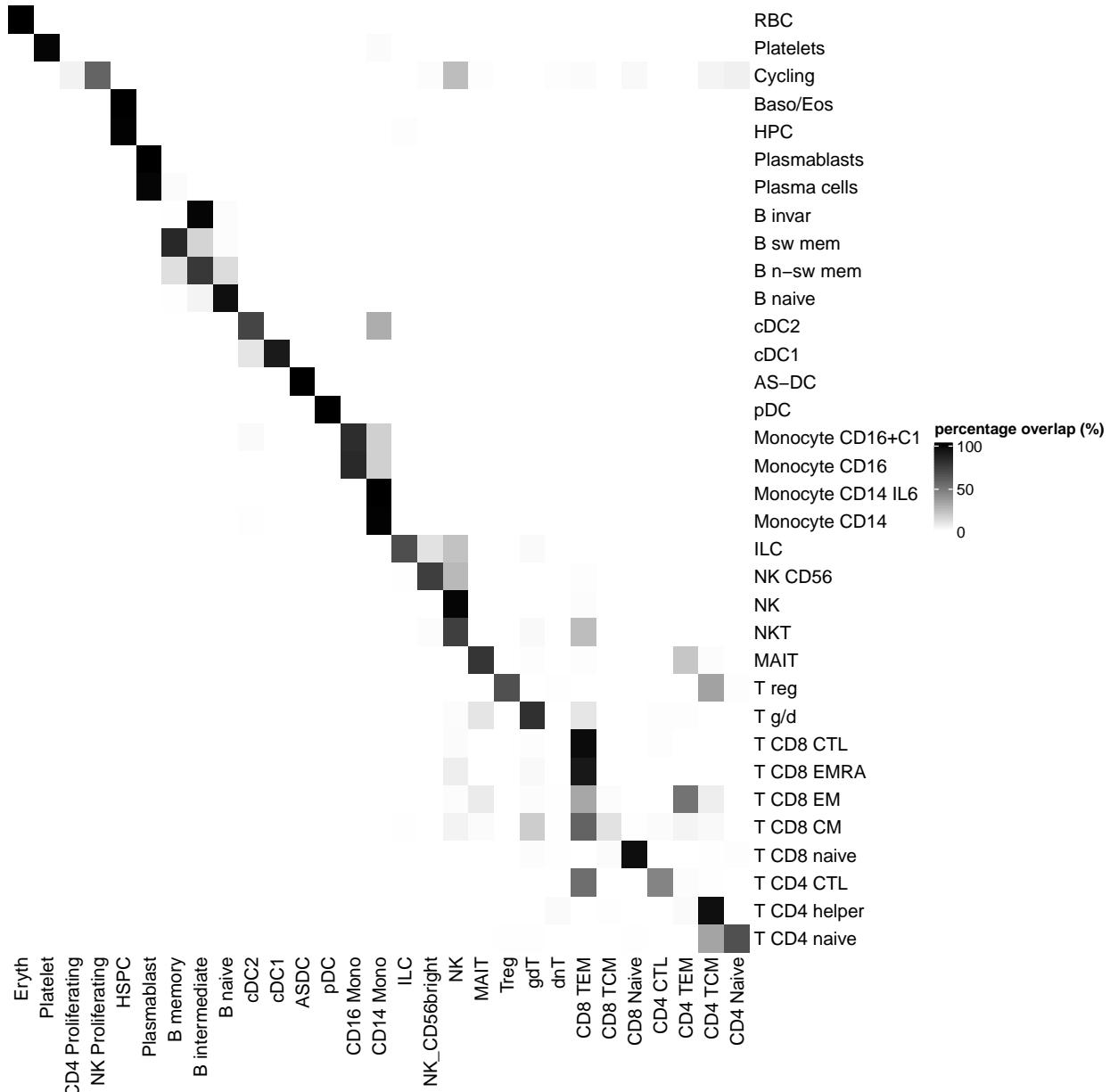
myMatrix <- as.data.frame.matrix(table(aziCp$cell_annotation_revision_short_woIfnStim[aziCp$predicted.score_>0.75], aziCp$predicted.id_azimuth[aziCp$predicted.score_azimuth > 0.75])/rowSums(table(aziCp$cell_annotation_revision_short_woIfnStim[aziCp$predicted.score_azimuth > 0.75], aziCp$predicted.id_azimuth[aziCp$predicted.score_azimuth > 0.75])) * 100

matchedOrder <- c("Eryth", "Platelet", "CD4 Proliferating", "NK Proliferating",
                  "HSPC", "Plasmablast", "B memory", "B intermediate", "B naive",
                  "cDC2", "cDC1", "ASDC", "pDC", "CD16 Mono", "CD14 Mono",
                  "ILC", "NK_CD56bright", "NK", "MAIT", "Treg", "gdT", "dNT",
                  "CD8 TEM", "CD8 TCM", "CD8 Naive", "CD4 CTL", "CD4 TEM",
                  "CD4 TCM", "CD4 Naive")
Heatmap(myMatrix[, matchedOrder], cluster_rows = F, cluster_columns = F,
        col = circlize::colorRamp2(c(0, 100), c("white", "black"))),

```

```
name = "percentage overlap (%)")
```

```
## Warning: The input is a data frame, convert it to the matrix.
```



Plot some general QCs

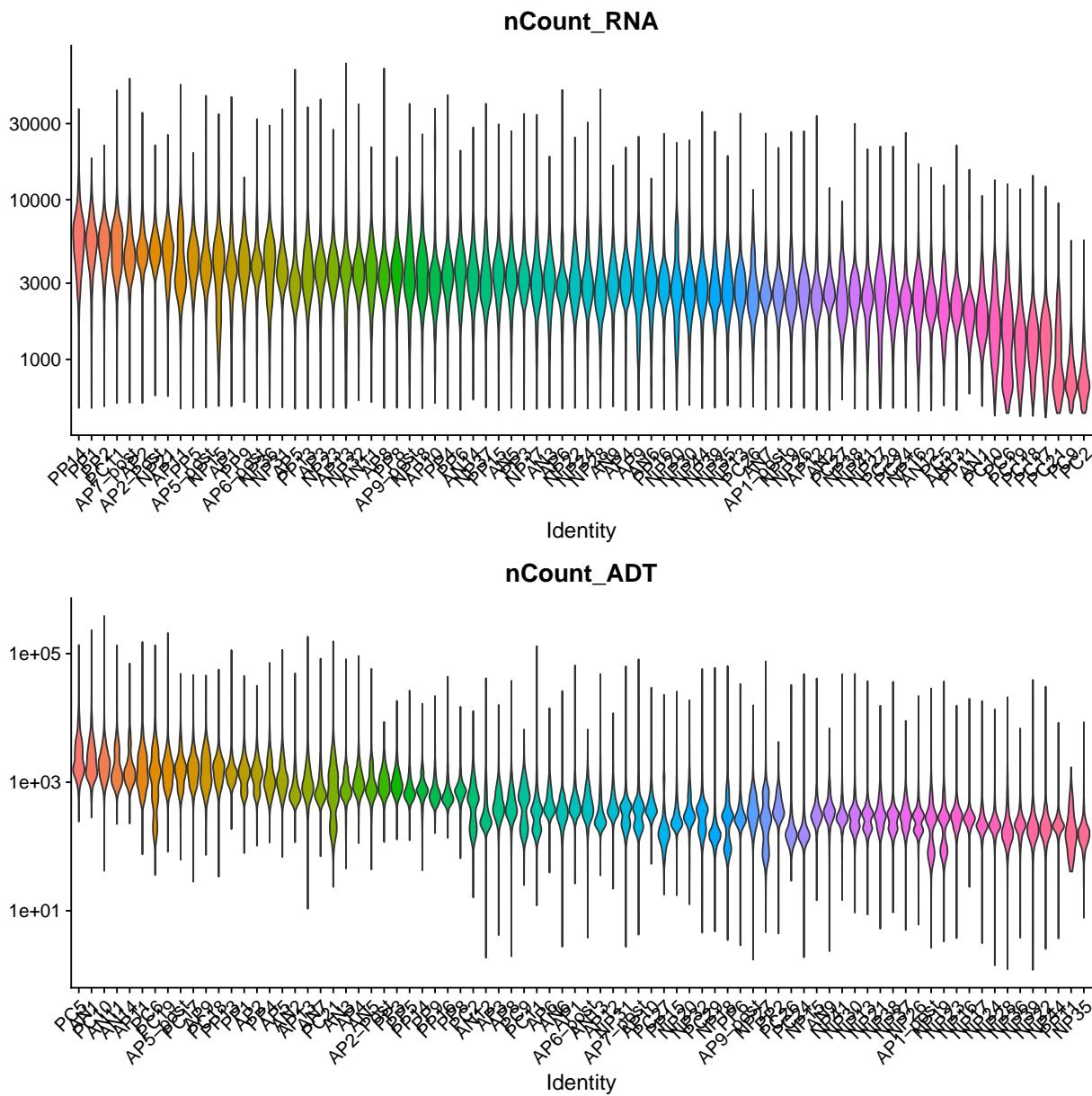
```
# 13 QC plots
cv@meta.data$select <- paste(cv$covid_status, "--", cv$ageGroup,
"--", cv$patient_id)
cv@meta.data$select_factor <- factor(cv@meta.data$select, levels = rev(unique(cv@meta.data$select)[order(
unique(cv@meta.data$select)), grep("COVID", unique(cv@meta.data$select)),
grep("Conval", unique(cv@meta.data$select)), grep("Neonate",
unique(cv@meta.data$select)), grep("Infant", unique(cv@meta.data$select)))]))
```

```

grepl("Young", unique(cv@meta.data$select)), grepl("Child",
unique(cv@meta.data$select)), grepl("Adolesc", unique(cv@meta.data$select)),
grepl("Adult", unique(cv@meta.data$select)), grepl("Elderly",
unique(cv@meta.data$select))))])

VlnPlot(cv, features = c("nCount_RNA", "nCount_ADT"), group.by = "patient_id",
pt.size = 0, sort = T, log = 10) + plot_layout(ncol = 1)

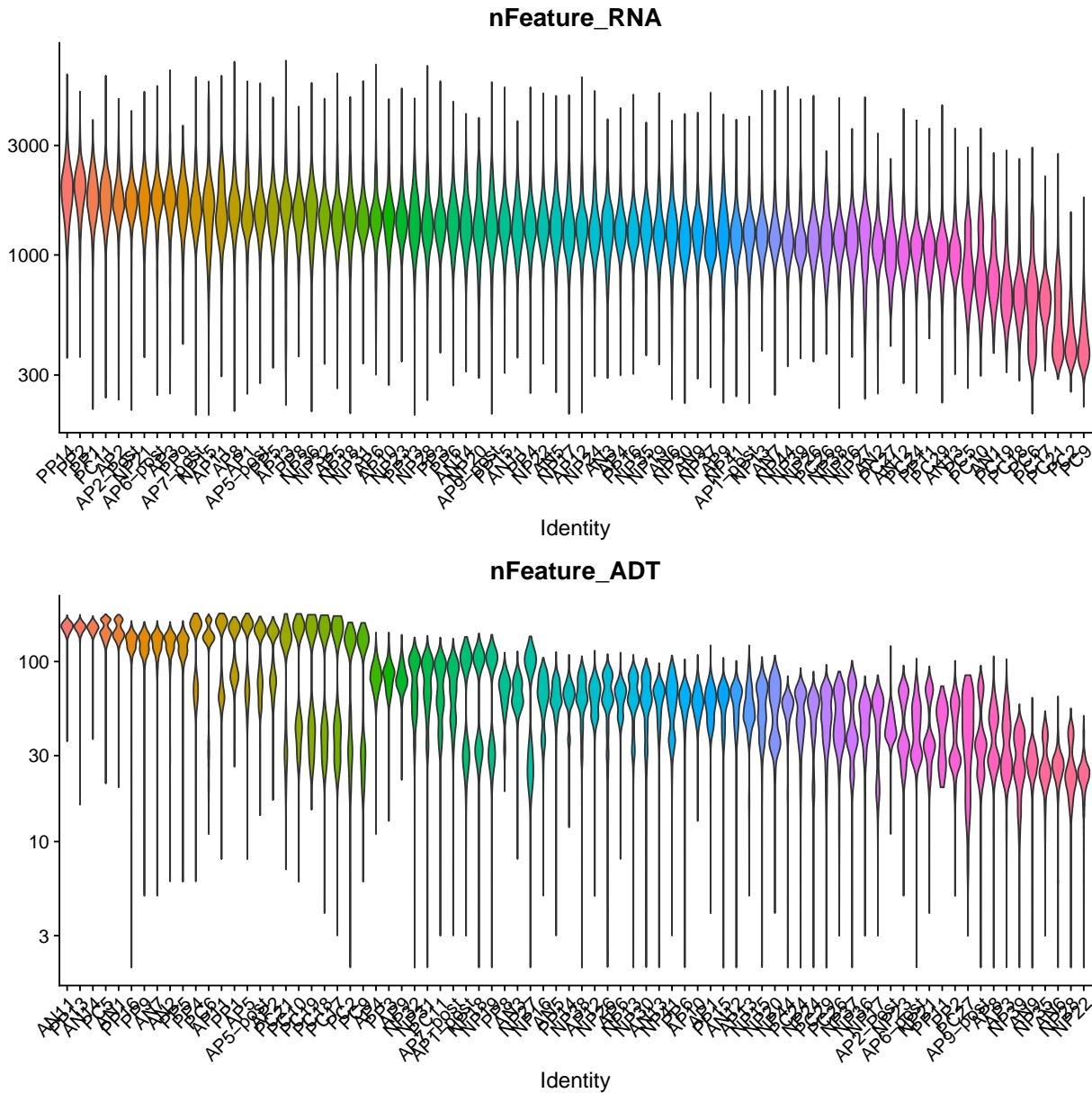
```



```

VlnPlot(cv, features = c("nFeature_RNA", "nFeature_ADT"), group.by = "patient_id",
pt.size = 0, sort = T, log = 10) + plot_layout(ncol = 1)

```



```

myColsPatient <- randomcoloR::distinctColorPalette(length(unique(cv@meta.data$patient_id)))
myColsBatch <- randomcoloR::distinctColorPalette(length(unique(cv@meta.data$orig.ident)))
(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "patient_id",
        shuffle = T, cols = myColsPatient) + theme(aspect.ratio = 1))/(DimPlot(cv,
        reduction = "wnn.umap_rnaAdt", group.by = "orig.ident", shuffle = T) +
        theme(aspect.ratio = 1))/(DimPlot(cv, reduction = "wnn.umap_rnaAdt",
        group.by = "Male.Female", shuffle = T) + theme(aspect.ratio = 1))

```

```

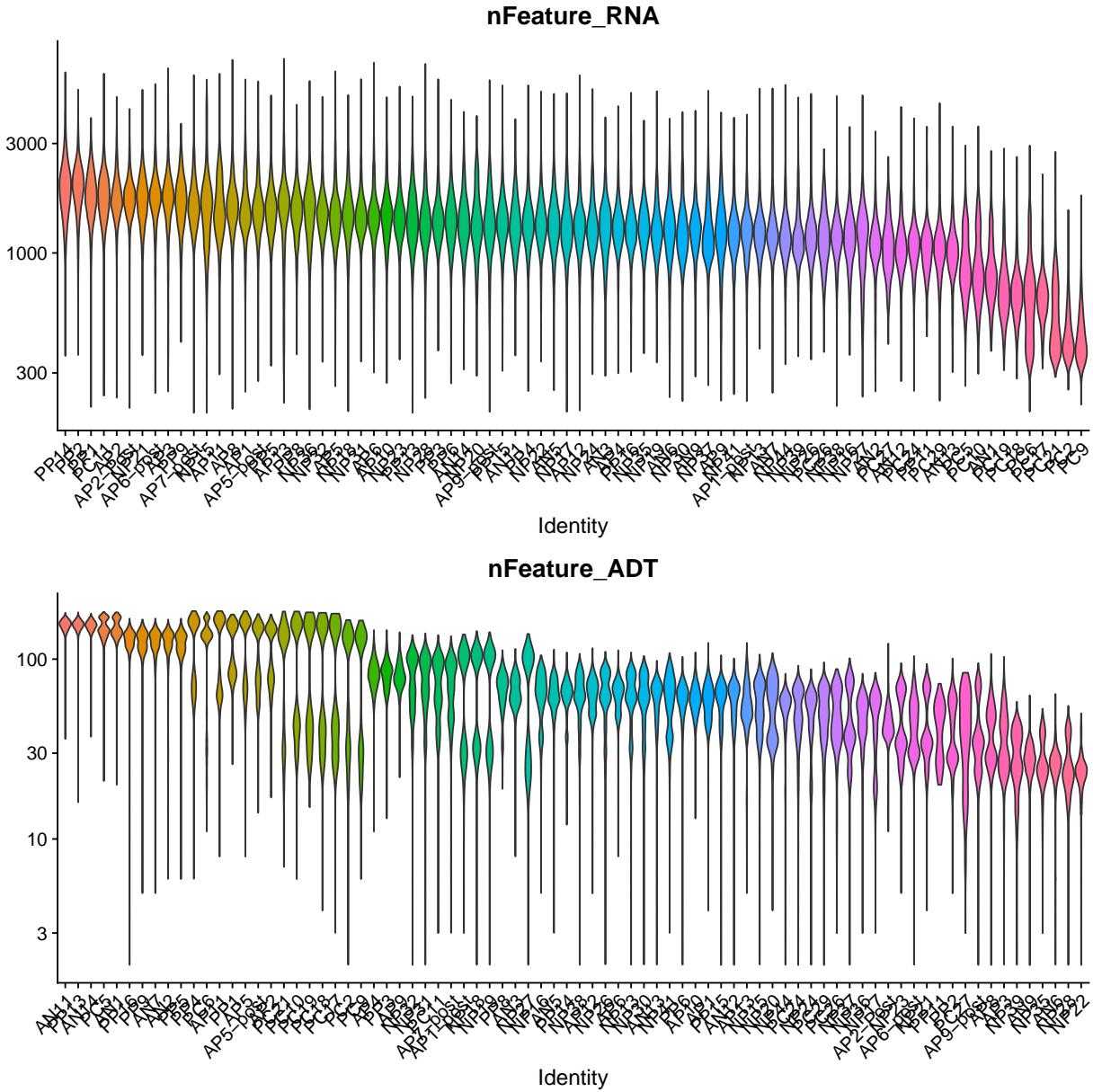
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

```



Female
Female
Male
Male

```
myColsBatch <- randomcoloR::distinctColorPalette(length(unique(cv@meta.data$orig.ident)))
myColsPatient <- randomcoloR::distinctColorPalette(length(unique(cv@meta.data$patient_id)))
VlnPlot(cv, features = c("nFeature_RNA", "nFeature_ADT"), group.by = "patient_id",
        pt.size = 0, sort = T, log = 10) + plot_layout(ncol = 1)
```



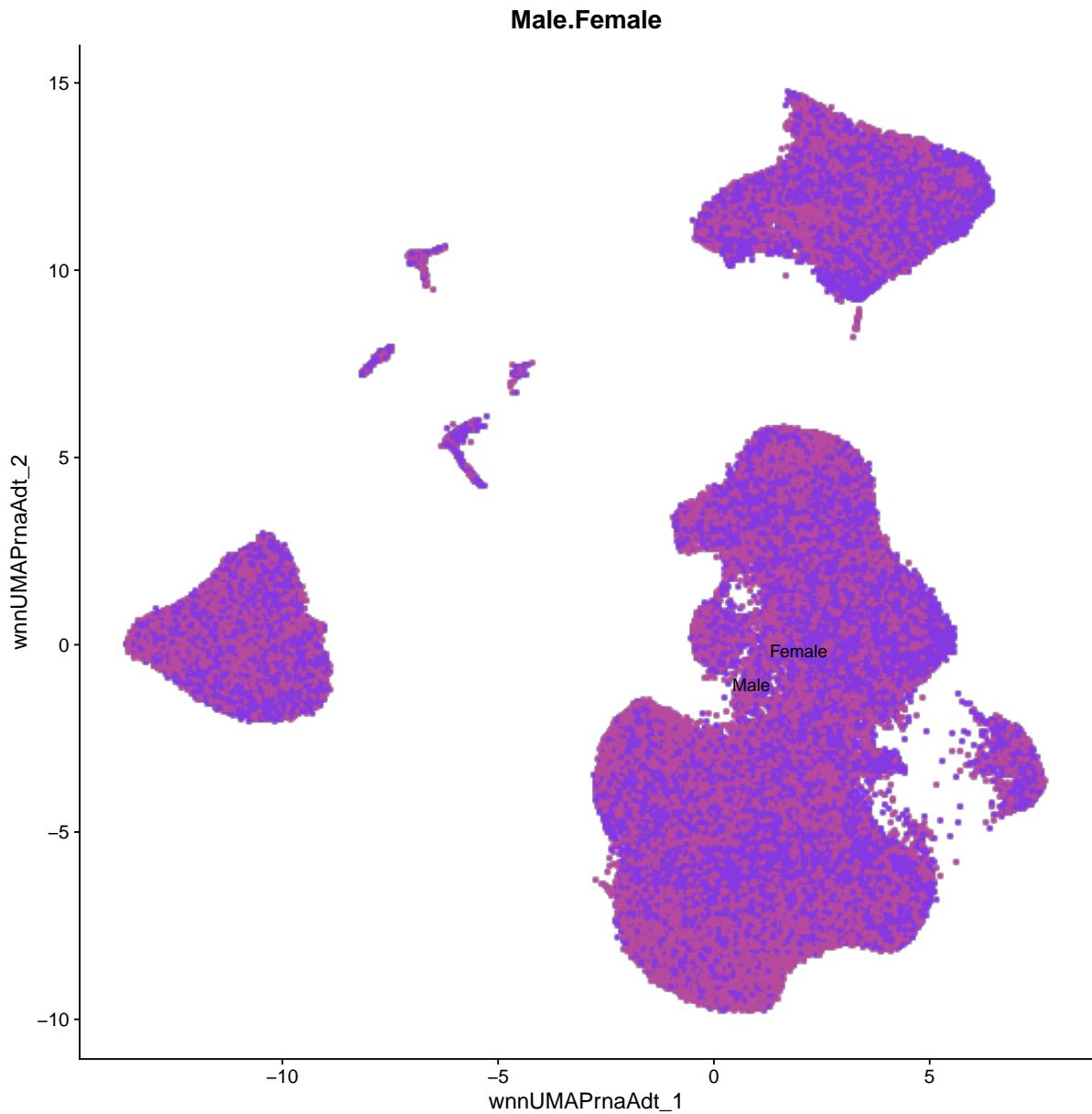
```

cv@meta.data$log10_nFeature_RNA <- log10(cv@meta.data$nFeature_GEX +
  1)
cv@meta.data$log10_nFeature_ADT <- log10(cv@meta.data$nFeature_ADT +
  1)

(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "Male.Female",
  na.value = NA, shuffle = T, cols = myColsBatch, raster = T,
  label = T) + theme(aspect.ratio = 1)) + NoLegend()

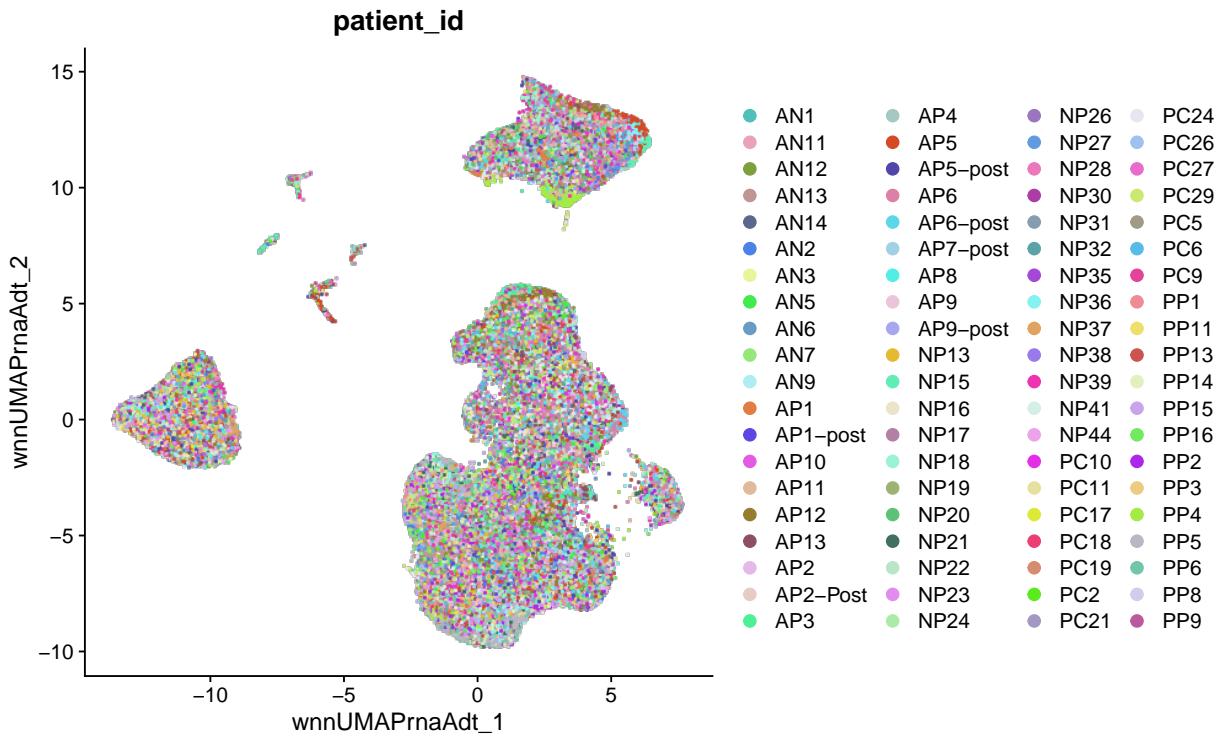
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'

```



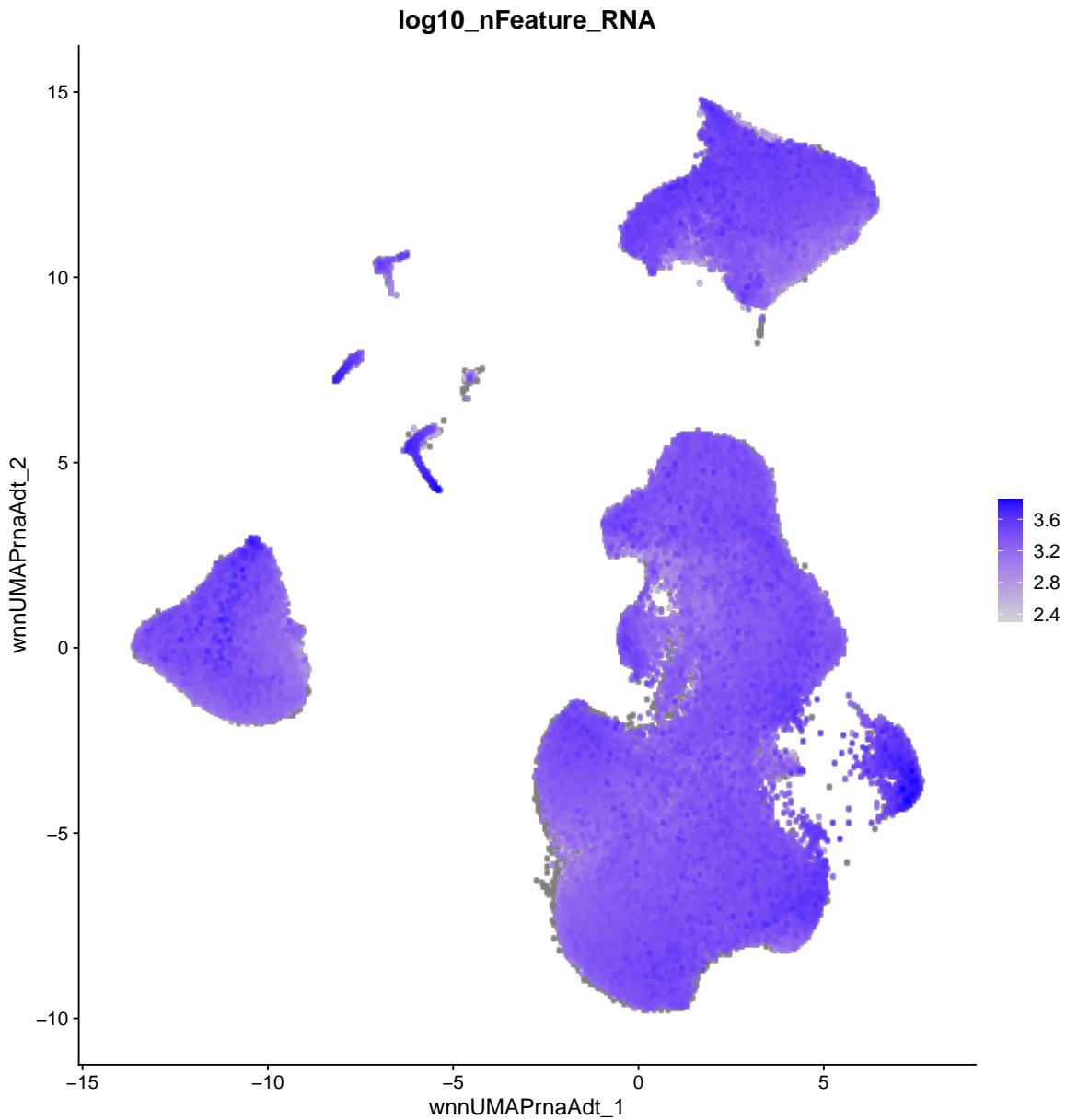
```
(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "patient_id",
    shuffle = T, cols = myColsPatient, raster = T) + theme(aspect.ratio = 1))
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



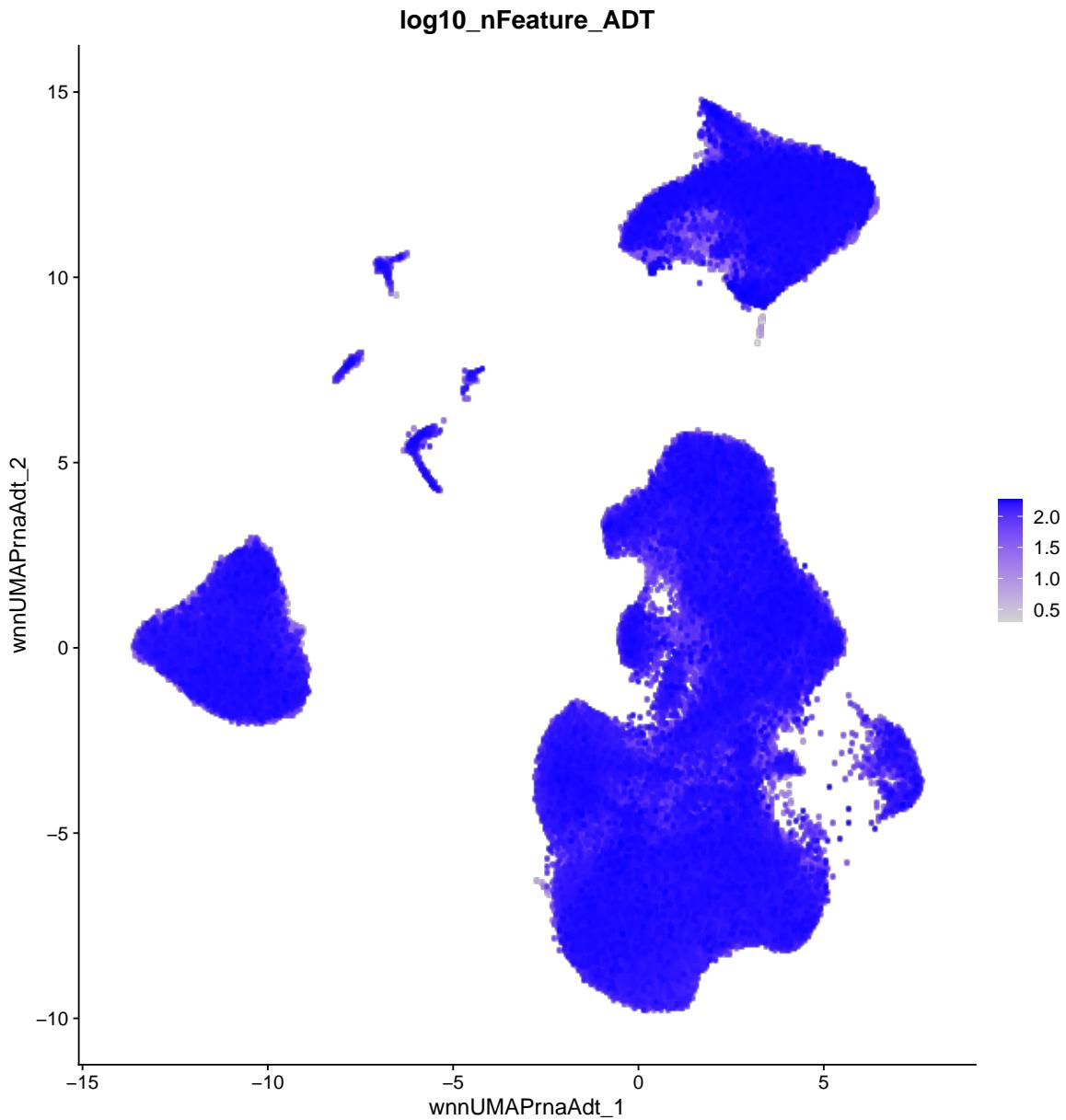
```
FeaturePlot(cv, reduction = "wnn.umap_rnaAdt", features = "log10_nFeature_RNA",
keep.scale = "all", order = T, raster = T, coord.fixed = T)
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



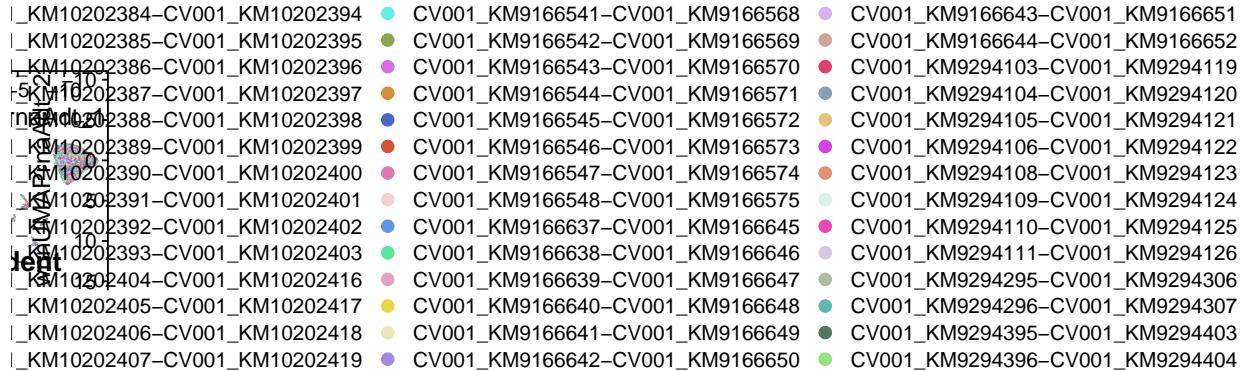
```
FeaturePlot(cv, reduction = "wnn.umap_rnaAdt", features = "log10_nFeature_ADT",
           order = T, raster = T, coord.fixed = T)
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



```
(DimPlot(cv, reduction = "wnn.umap_rnaAdt", group.by = "orig.ident",
      shuffle = T, cols = myColsBatch, raster = T) + theme(aspect.ratio = 1))
```

```
## Rasterizing points since number of points exceeds 100,000.
## To disable this behavior set 'raster=FALSE'
```



```

qcData <- FetchData(cv, c("nFeature_RNA", "nFeature_ADT", "nCount_RNA",
  "nCount_ADT", "orig.ident", "Male.Female", "patient_id",
  "wnnUMAPrnaAdt_1", "wnnUMAPrnaAdt_2", "percent.mt"))
qcData <- qcData[sample(1:nrow(qcData), size = nrow(qcData),
  replace = F), ]

qcData$nFeature_RNA_log10 <- log10(qcData$nFeature_RNA + 1)
qcData$nFeature_ADT_log10 <- log10(qcData$nFeature_ADT + 1)
qcData$nCount_RNA_log10 <- log10(qcData$nCount_RNA + 1)
qcData$nCount_ADT_log10 <- log10(qcData$nCount_ADT + 1)
qcData$batch <- paste0("L", as.numeric(as.factor(qcData$orig.ident)))

# cowplot::plot_grid(ncol=2,
# ggplot(qcData, aes(x=wnnUMAPrnaAdt_1, y=wnnUMAPrnaAdt_2, col=nFeature_RNA_log10))

```

```

# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_gradientn(colours =
# c('yellow','blue'),guide = 'colourbar',limits=c(2,4)) +
# theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=nFeature_ATD_log10))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_gradientn(colours =
# c('yellow','blue'),guide = 'colourbar',limits=c(0,2.5)) +
# theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=nFeature_RNA_log10))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_gradientn(colours =
# c('yellow','blue'),guide = 'colourbar',limits=c(0,5)) +
# theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=nCount_ATD_log10))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_gradientn(colours =
# c('yellow','blue'),guide = 'colourbar',limits=c(0,5)) +
# theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=Male.Female))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_manual(values =
# randomcoloR::distinctColorPalette(length(unique(cv@meta.data$Male.Female))))
# + theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=batch))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_manual(values =
# randomcoloR::distinctColorPalette(length(unique(qcData$batch))))
# + theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=percent_mt))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_gradientn(colours =
# c('yellow','blue'),guide = 'colourbar',limits=c(0,10)) +
# theme(aspect.ratio = 1) + theme_classic(),
# ggplot(qcData,aes(x=wnnUMAPrnaAdt_1,y=wnnUMAPrnaAdt_2,col=patient_id))
# +
# ggrastr::rasterise(geom_point(size=.1,shape=16),dpi=300)
# + scale_color_manual(values =
# randomcoloR::distinctColorPalette(length(unique(qcData$patient_id))))
# + theme(aspect.ratio = 1) + theme_classic(), align =
# 'hv')

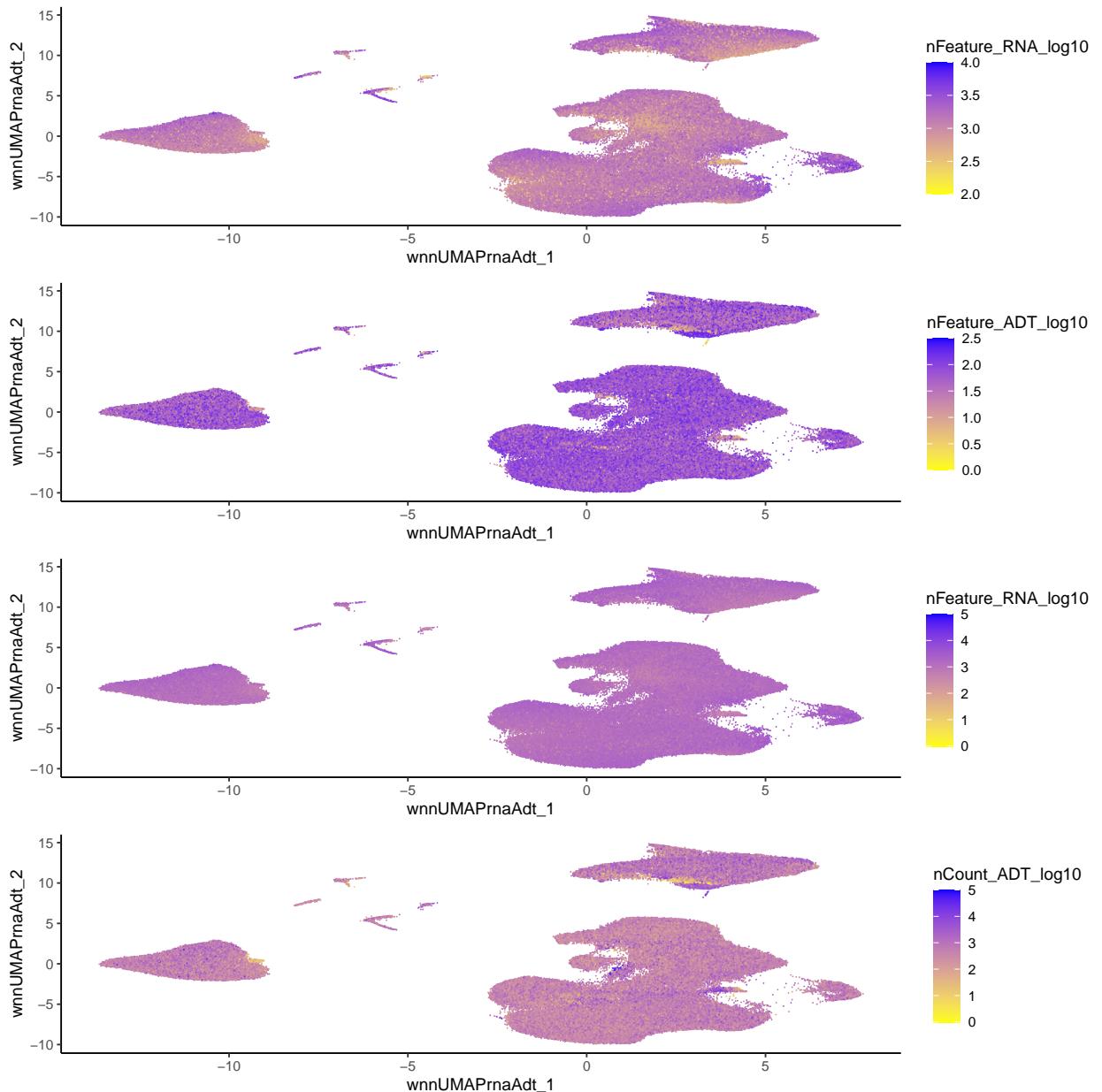
cowplot::plot_grid(ncol = 1, ggplot(qcData, aes(x = wnnUMAPrnaAdt_1,
y = wnnUMAPrnaAdt_2, col = nFeature_RNA_log10)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_gradientn(colours = c("yellow",
"blue"), guide = "colourbar", limits = c(2, 4)) + theme(aspect.ratio = 1) +

```

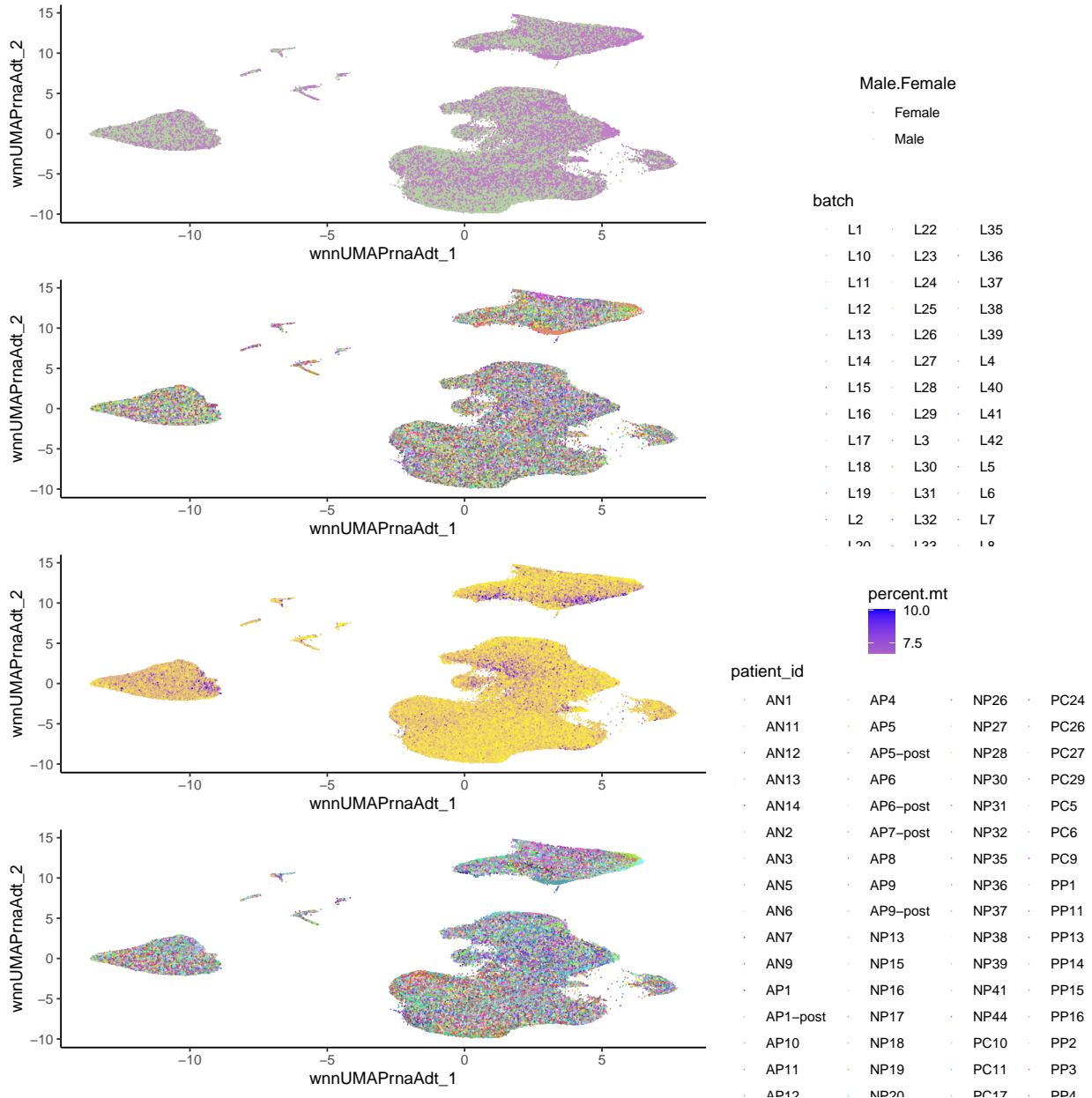
```

theme_classic(), ggplot(qcData, aes(x = wnnUMAPrnaAdt_1,
y = wnnUMAPrnaAdt_2, col = nFeature_RNA_log10)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_gradientn(colours = c("yellow",
"blue"), guide = "colourbar", limits = c(0, 2.5)) + theme(aspect.ratio = 1) +
theme_classic(), ggplot(qcData, aes(x = wnnUMAPrnaAdt_1,
y = wnnUMAPrnaAdt_2, col = nFeature_RNA_log10)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_gradientn(colours = c("yellow",
"blue"), guide = "colourbar", limits = c(0, 5)) + theme(aspect.ratio = 1) +
theme_classic(), ggplot(qcData, aes(x = wnnUMAPrnaAdt_1,
y = wnnUMAPrnaAdt_2, col = nCount_ADT_log10)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_gradientn(colours = c("yellow",
"blue"), guide = "colourbar", limits = c(0, 5)) + theme(aspect.ratio = 1) +
theme_classic(), align = "hv"

```



```
cowplot::plot_grid(ncol = 1, ggplot(qcData, aes(x = wnnUMAPrnaAdt_1,
y = wnnUMAPrnaAdt_2, col = Male.Female)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_manual(values = randomcoloR::distinctColorPalette(length(unique(qcData$Male.Female))),
theme(aspect.ratio = 1) + theme_classic(), ggplot(qcData,
aes(x = wnnUMAPrnaAdt_1, y = wnnUMAPrnaAdt_2, col = batch)) +
ggrastr::rasterise(geom_point(size = 0.1, shape = 16), dpi = 300) +
scale_color_manual(values = randomcoloR::distinctColorPalette(length(unique(qcData$batch)))), theme(aspect.ratio = 1) + theme_classic(), ggplot(qcData,
aes(x = wnnUMAPrnaAdt_1, y = wnnUMAPrnaAdt_2, col = percent.mt)) +
ggrastr::rasterise(geom_point(size = 0.1, shape = 16), dpi = 300) +
scale_color_gradientn(colours = c("yellow", "blue"), guide = "colourbar",
limits = c(0, 10)) + theme(aspect.ratio = 1) + theme_classic(),
ggplot(qcData, aes(x = wnnUMAPrnaAdt_1, y = wnnUMAPrnaAdt_2,
col = patient_id)) + ggrastr::rasterise(geom_point(size = 0.1,
shape = 16), dpi = 300) + scale_color_manual(values = randomcoloR::distinctColorPalette(length(unique(qcData$patient_id))),
theme(aspect.ratio = 1) + theme_classic(), align = "hv")
```

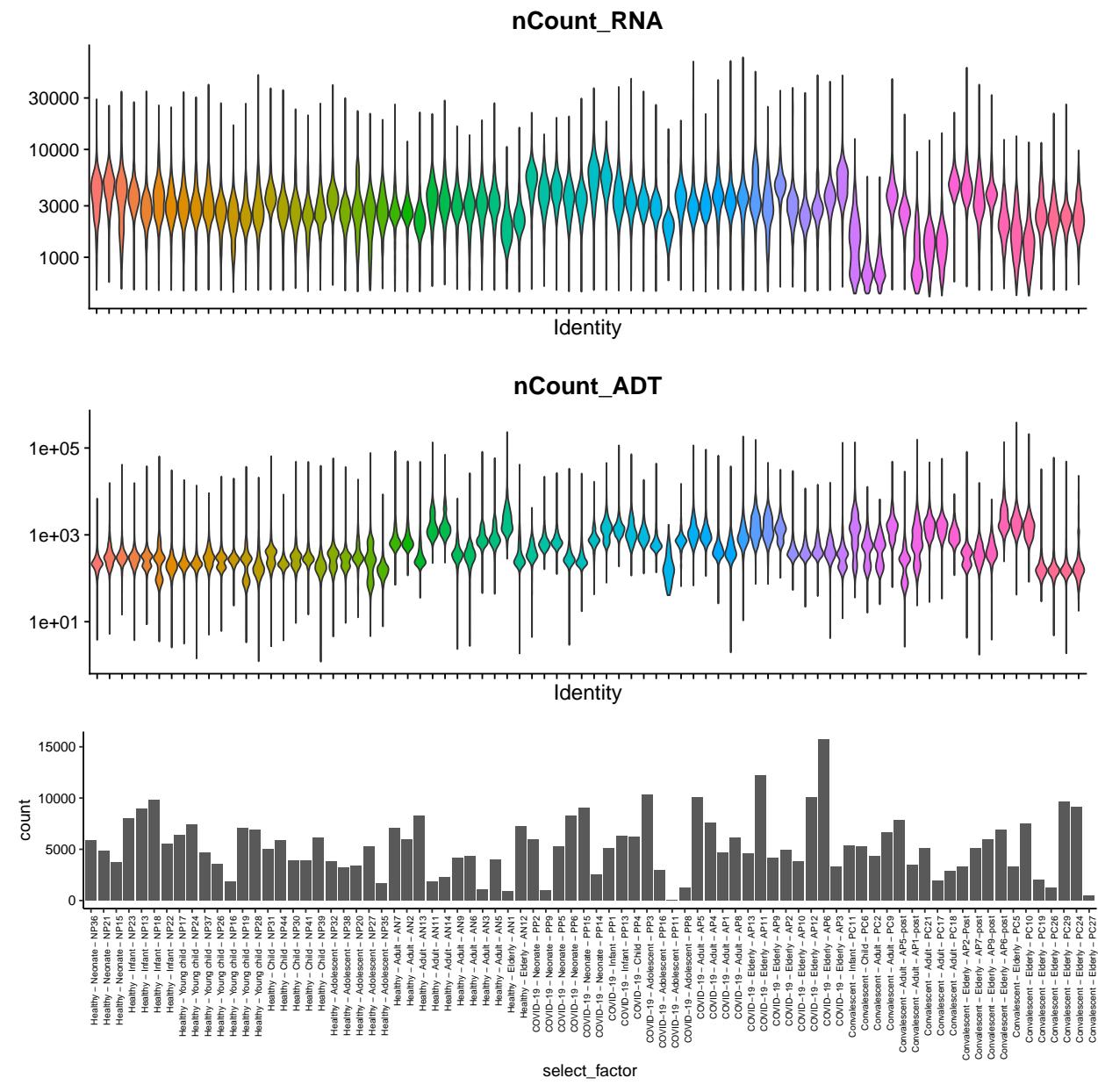


```
cowplot::plot_grid(ncol = 1, VlnPlot(cv, features = c("nCount_RNA"),
  group_by = "select_factor", pt.size = 0, sort = F, log = 10) +
  NoLegend() + theme(axis.text.x = element_blank()), VlnPlot(cv,
  features = c("nCount_ADT"), group_by = "select_factor", pt.size = 0,
  sort = F, log = 10) + NoLegend() + theme(axis.text.x = element_blank()),
  ggplot(cv@meta.data, aes(x = select_factor)) + geom_bar() +
  theme_classic() + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, colour = "black", size = 6), axis.text.y = element_text(colour = "black")),
  align = "hv")
```

Warning: Graphs cannot be vertically aligned unless the axis parameter is set.
Placing graphs unaligned.

Warning: Graphs cannot be horizontally aligned unless the axis parameter is set.

```
## Placing graphs unaligned.
```



```
theme(axis.text.y = element_blank(), axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust = 1))
```

```
## List of 2
## $ axis.text.x:List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : num 1
##   ..$ vjust       : num 0.5
```

```

##   ..$ angle      : num 90
##   ..$ lineheight : NULL
##   ..$ margin     : NULL
##   ..$ debug      : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y: list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##   - attr(*, "class")= chr [1:2] "theme" "gg"
##   - attr(*, "complete")= logi FALSE
##   - attr(*, "validate")= logi TRUE

```

We rank our patients by the proportion of IFN stimulated PBMCs, to find correlates in the nasal data

```

# 11* Something to show ranked by blood IFN Color code bars
# to make the more 'squisable'
stimGenes <- rev(c("IRF7", "XAF1", "UBE2L6", "TRIM22", "STAT1",
  "SP110", "SAMD9L", "SAMD9", "PLSCR1", "PARP9", "OAS2", "OAS1",
  "MX2", "MX1", "LY6E", "ISG15", "IFIT3", "IFI6", "IFI44L",
  "IFI35", "HERC5", "EPSTI1", "EIF2AK2", "CMPK2", "BST2"))

nasal <- AddModuleScore(nasal, features = list(stimGenes), name = "IFN_stimulation_signature")
ifnMolecules <- rownames(nasal[["RNA"]])[grepl("IFN", rownames(nasal[["RNA"]])) &
  !grepl("IFN.*(R|AS)", rownames(nasal[["RNA"]]))]
names(ifnMolecules)[grepl("IFN", ifnMolecules)] <- "Type I"
names(ifnMolecules)[grepl("IFNG", ifnMolecules)] <- "Type II"
names(ifnMolecules)[grepl("IFNL", ifnMolecules)] <- "Type III"
nasal <- AddModuleScore(nasal, features = list(ifnMolecules[names(ifnMolecules) ==
  "Type I"]), name = "IFN_Type_I")
nasal <- AddModuleScore(nasal, features = list(ifnMolecules[names(ifnMolecules) ==
  "Type II"]), name = "IFN_Type_II")
nasal <- AddModuleScore(nasal, features = list(ifnMolecules[names(ifnMolecules) ==
  "Type III"]), name = "IFN_Type_III")

bloodIFNiProps <- table(grepl("IFN stim", cv@meta.data$cell_annotation_short[cv@meta.data$patient_id %
  nasal@meta.data$donor]), cv@meta.data$patient_id[cv@meta.data$patient_id %in%
  nasal@meta.data$donor])
bloodIFNiProps <- apply(bloodIFNiProps, 2, function(x) x/sum(x))["TRUE",
  ]
bloodIFNiProps_table <- as.data.frame(bloodIFNiProps)
bloodIFNiProps_table$id <- factor(rownames(bloodIFNiProps_table),
  levels = rownames(bloodIFNiProps_table)[order(bloodIFNiProps_table)])
bloodIFNiProps_table$ident <- sapply(bloodIFNiProps_table$id,
  function(x) paste(unique(cv$paedOrAdult[cv$patient_id ==
    x]), unique(cv$covid_status[cv$patient_id == x])))
bloodIFNiProps_table$ident <- factor(bloodIFNiProps_table$ident,
  levels = c("Paediatric Healthy", "Adult Healthy", "Paediatric COVID-19",
  "Adult COVID-19", "Paediatric Convalescent", "Adult Convalescent"))

mySubsetDcOnly <- subset(nasal, cells = rownames(nasal@meta.data)[nasal@meta.data$donor %in%
  names(bloodIFNiProps) & grepl("DC", nasal@meta.data$v6_annotation)]) # For revision: also create a very
mySubsetDcOnly@meta.data$donor_factor <- factor(mySubsetDcOnly@meta.data$donor,
  levels = names(bloodIFNiProps)[order(bloodIFNiProps)])
mySubset <- subset(nasal, cells = rownames(nasal@meta.data)[nasal@meta.data$donor %in%

```

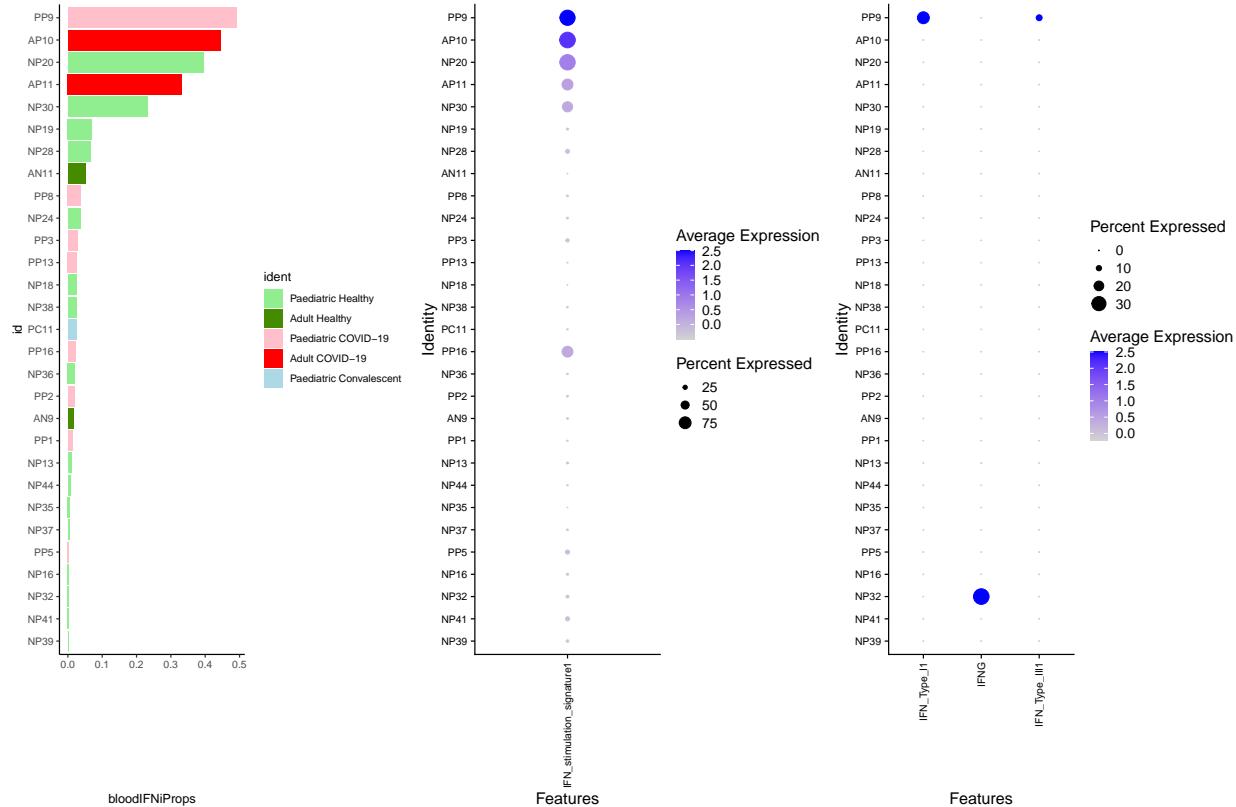
```

  names(bloodIFNiProps) & nasal$donor %in% mySubsetDcOnly$donor])
mySubset@meta.data$donor_factor <- factor(mySubset@meta.data$donor,
  levels = names(bloodIFNiProps)[order(bloodIFNiProps)])
mySubsetPp9 <- subset(mySubset, cells = rownames(nasal@meta.data)[nasal@meta.data$donor ==
  "PP9"])
mySubsetPp9_pbmc <- subset(cv, cells = rownames(cv@meta.data)[cv@meta.data$patient_id ==
  "PP9"])

covidPosCellsMat <- table(mySubset@meta.data$COVID19_infected_cell,
  mySubset@meta.data$donor_factor)
covidPosCellsMat <- apply(covidPosCellsMat, 2, function(x) x/sum(x))["COVID19_infected_cell",
  ]
covidPosCells <- data.frame(posCells = covidPosCellsMat)
covidPosCells$donor <- factor(rownames(covidPosCells), levels = names(bloodIFNiProps)[order(bloodIFNiProps)])
bloodIFNiProps_table <- bloodIFNiProps_table[rownames(bloodIFNiProps_table) %in%
  mySubset$donor, ]

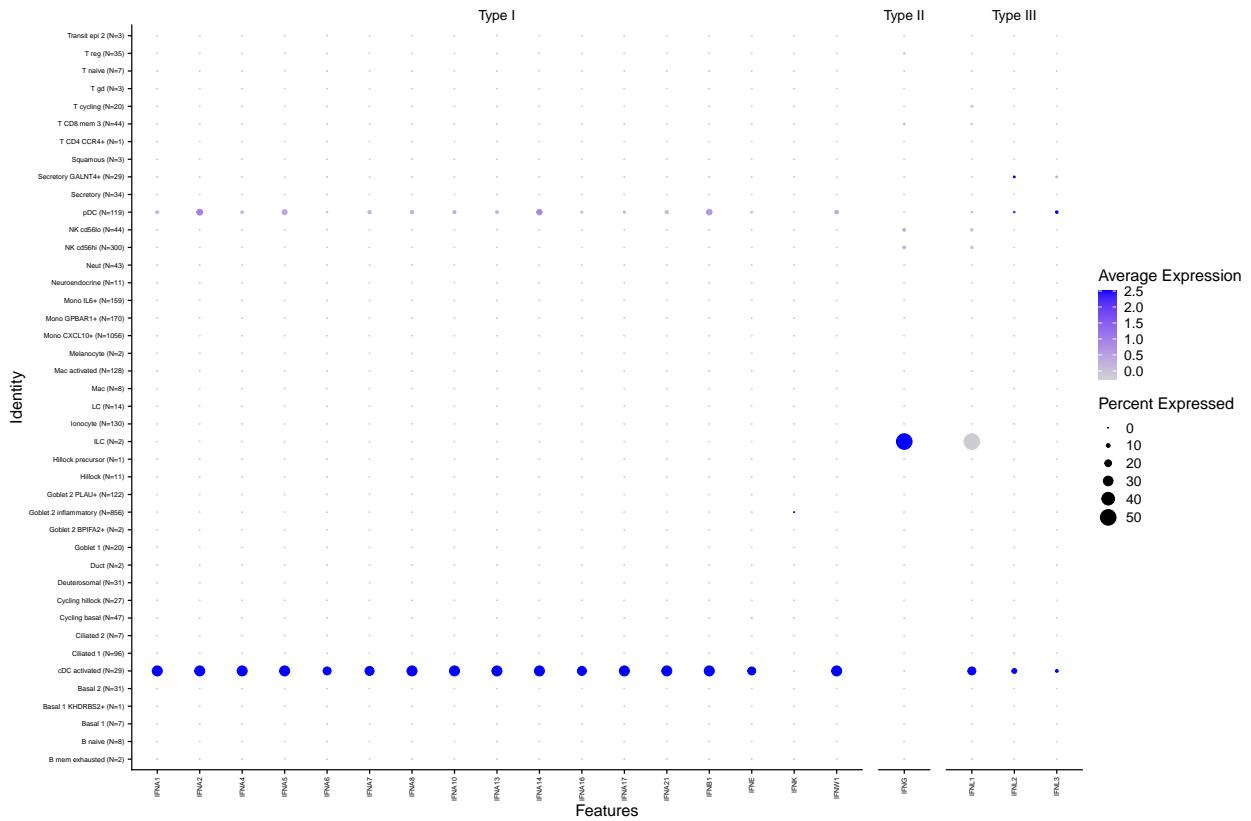
(ggplot(bloodIFNiProps_table, aes(x = id, y = bloodIFNiProps,
  fill = ident)) + scale_fill_manual(values = c("lightgreen",
  "chartreuse4", "pink", "red", "lightblue", "blue")) + geom_bar(stat = "identity") +
  coord_flip() + theme_classic() + (DotPlot(mySubset, features = c("IFN_stimulation_signature1"),
  group.by = "donor_factor", cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 9), axis.text.y = element_text(size = 9))) +
  (DotPlot(mySubsetDcOnly, features = c("IFN_Type_I1", "IFNG",
  "IFN_Type_III1"), group.by = "donor_factor", cluster.idents = F) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1,
  vjust = 0.5, size = 9), axis.text.y = element_text(size = 9)))

```

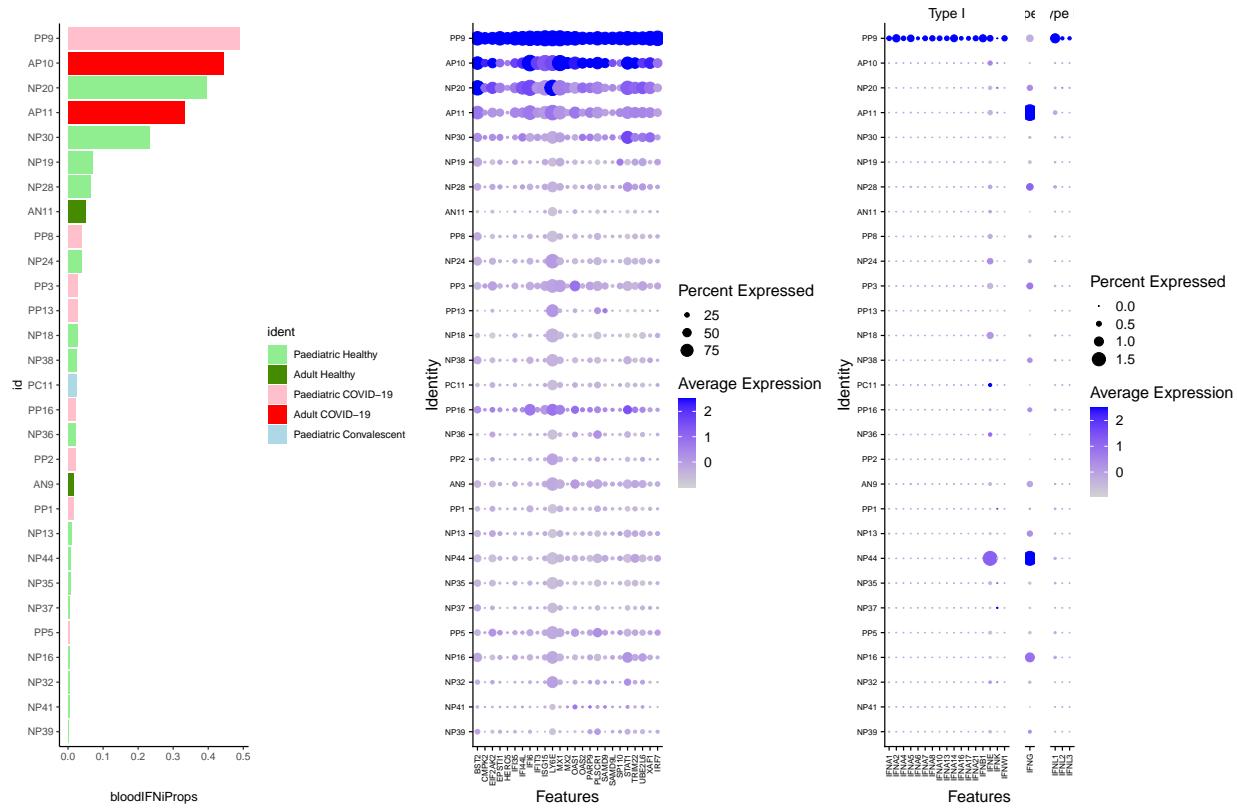


```
# 12* Show DCs in PP9 (w cell counts) (Perhaps full in
# supplement, collapse on IFN type 1, 2, 3 in main)
mySubsetPp9@meta.data$annot_wN <- NA
for (i in unique(mySubsetPp9$v6_annot2)) {
  mySubsetPp9$annot_wN[mySubsetPp9$v6_annot2 == i] <- paste0(i,
    " (N=", sum(mySubsetPp9$v6_annot2 == i), ")")
}

ifnMolecules <- ifnMolecules[order(names(ifnMolecules), !grep("^\u039bIFNA[0-9]$", names(ifnMolecules)), ifnMolecules)]
(DotPlot(mySubsetPp9, features = ifnMolecules, group.by = "annot_wN",
  cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 6), axis.text.y = element_text(size = 6)))
```



```
# Also with full genes for supplemental figures
(ggplot(bloodIFNiProps_table, aes(x = id, y = bloodIFNiProps,
  fill = ident)) + scale_fill_manual(values = c("lightgreen",
  "chartreuse4", "pink", "red", "lightblue", "blue")) + geom_bar(stat = "identity") +
  coord_flip() + theme_classic() + (DotPlot(mySubset, features = c(stimGenes),
  group.by = "donor_factor", cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7))) +
  (DotPlot(mySubset, features = c(ifnMolecules), group.by = "donor_factor",
  cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 7), axis.text.y = element_text(size = 7)))
```

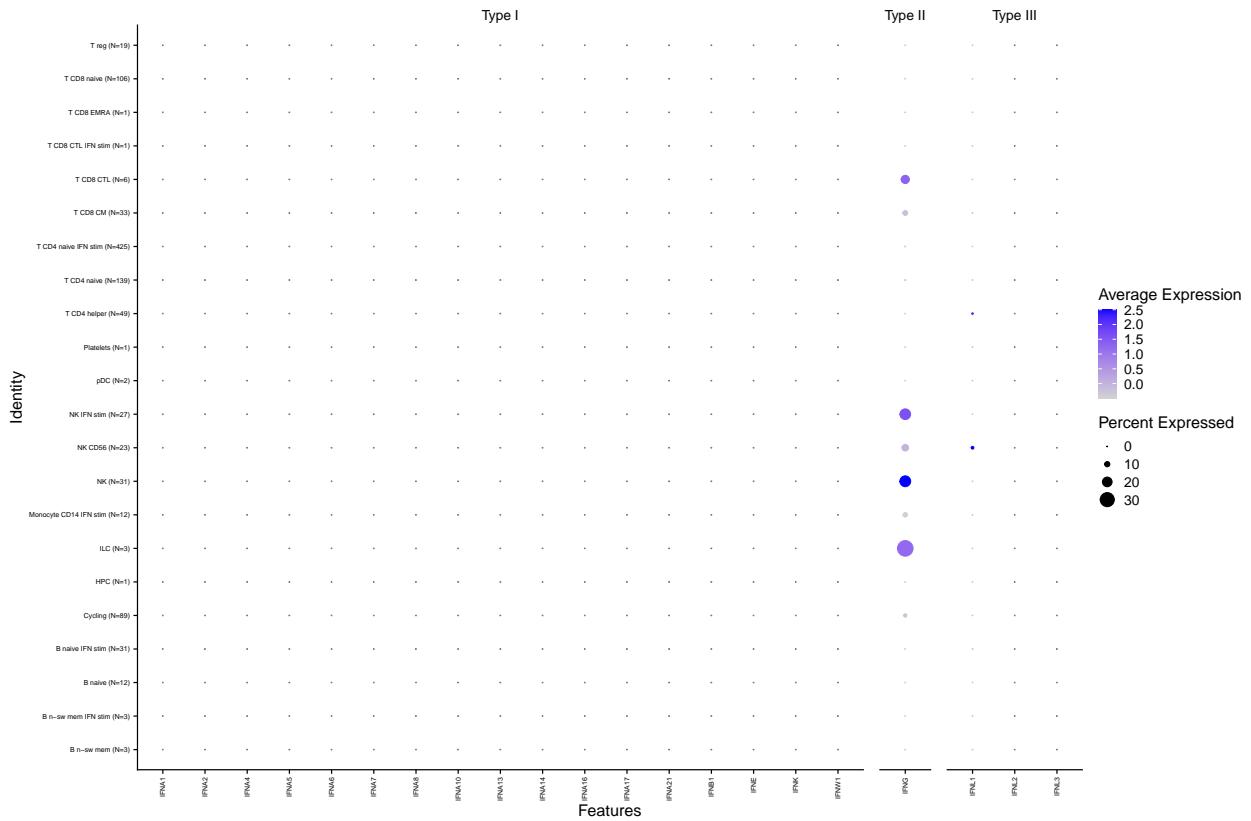


```

mySubsetPp9_pbmc@meta.data$annot_wN <- NA
for (i in unique(mySubsetPp9_pbmc$cell_annot_revision_short)) {
  mySubsetPp9_pbmc$annot_wN[mySubsetPp9_pbmc$cell_annot_revision_short ==
    i] <- paste0(i, " (N=", sum(mySubsetPp9_pbmc$cell_annot_revision_short ==
      i), ")")
}

(DotPlot(mySubsetPp9_pbmc, features = ifnMolecules, assay = "RNA",
  group.by = "annot_wN", cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
  hjust = 1, vjust = 0.5, size = 6), axis.text.y = element_text(size = 6)))

```

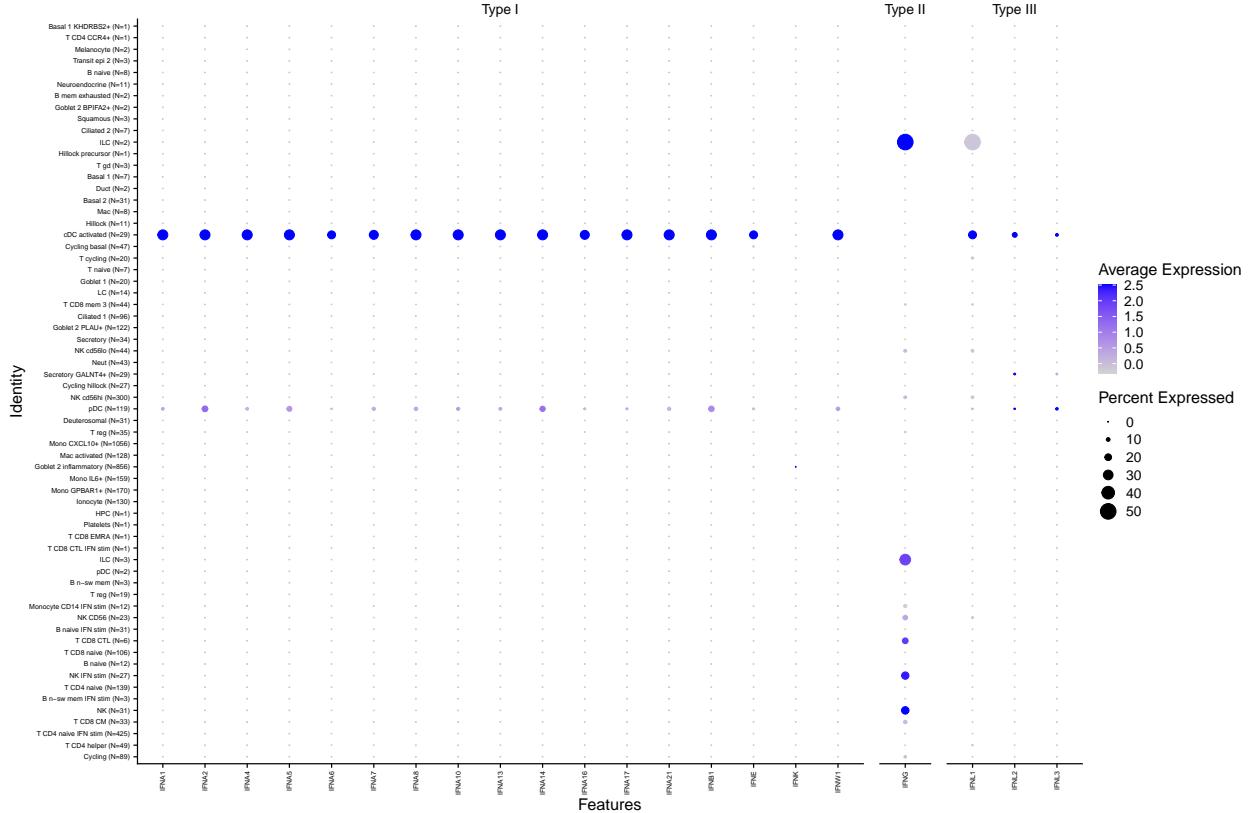


```

pp9Merge <- merge(mySubsetPp9, mySubsetPp9_pbmc, merge.data = TRUE)
pp9Merge$annot_wN_factor <- factor(pp9Merge$annot_wN, levels = c(unique(mySubsetPp9_pbmc$annot_wN),
unique(mySubsetPp9$annot_wN)))
Idents(pp9Merge) <- pp9Merge$annot_wN_factor

(DotPlot(pp9Merge, features = ifnMolecules, assay = "RNA", group.by = "annot_wN_factor",
cluster.idents = F) + theme(axis.text.x = element_text(angle = 90,
hjust = 1, vjust = 0.5, size = 6), axis.text.y = element_text(size = 6)))

```



The proportion of IFN stimulated PBMCs is plotted over time since onset of symptoms

```
# 7* Boxplot over onset after symptoms
cv@meta.data$weeksSinceOnsetSymptoms <- (cv@meta.data$If.COVID.19...Interval.between.first.symptoms.and
cv@meta.data$weeksSinceOnsetSymptoms <- floor(as.numeric(cv@meta.data$weeksSinceOnsetSymptoms)/7)
```

Warning: NAs introduced by coercion

```
cv@meta.data$weeksSinceOnsetSymptoms [cv@meta.data$weeksSinceOnsetSymptoms >
  2] <- ">=3"
cv@meta.data$weeksSinceOnsetSymptoms [is.na(cv@meta.data$weeksSinceOnsetSymptoms)] <- cv@meta.data$sever
```



```
bloodIFNiProps <- table(grep("IFN stim", cv@meta.data$cell_annot_revision_short),
  cv@meta.data$patient_id)
bloodIFNiProps <- apply(bloodIFNiProps, 2, function(x) x/sum(x))["TRUE",
  ]
bloodIFNiProps_table <- as.data.frame(bloodIFNiProps)
bloodIFNiProps_table$id <- factor(rownames(bloodIFNiProps_table),
  levels = rownames(bloodIFNiProps_table)[order(bloodIFNiProps_table)])
```



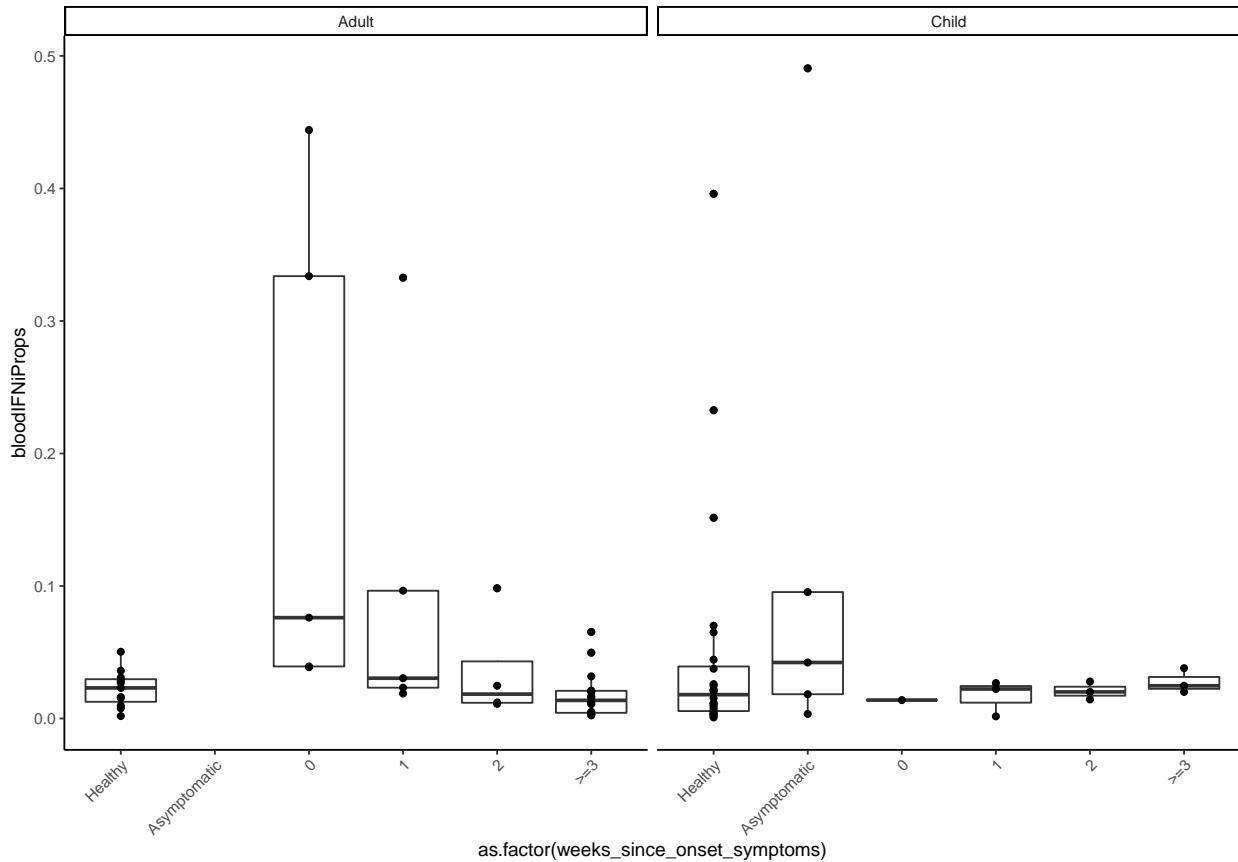
```
bloodIFNiProps_table$weeks_since_onset_symptoms <- sapply(bloodIFNiProps_table$id,
  function(x) unique(cv@meta.data$weeksSinceOnsetSymptoms[cv@meta.data$patient_id ==
    x]))
bloodIFNiProps_table$weeks_since_onset_symptoms <- factor(bloodIFNiProps_table$weeks_since_onset_symptom
```

```

    "0", "1", "2", ">=3"))
bloodIFNiProps_table$age_year <- as.numeric(as.character(sapply(bloodIFNiProps_table$id,
  function(x) unique(cv@meta.data$age_year[cv@meta.data$patient_id ==
    x]))))
bloodIFNiProps_table$severity_simple <- (as.character(sapply(bloodIFNiProps_table$id,
  function(x) unique(cv@meta.data$severity_simple[cv@meta.data$patient_id ==
    x]))))
bloodIFNiProps_table$severity <- (as.character(sapply(bloodIFNiProps_table$id,
  function(x) unique(cv@meta.data$severity[cv@meta.data$patient_id ==
    x]))))
bloodIFNiProps_table$paedOrAdult <- ifelse(bloodIFNiProps_table$age_year >
  18, "Adult", "Child")

ggplot(bloodIFNiProps_table[!is.na(bloodIFNiProps_table$weeks_since_onset_symptoms),
  ], aes(as.factor(weeks_since_onset_symptoms), bloodIFNiProps)) +
  geom_boxplot() + geom_point() + facet_wrap(~paedOrAdult) +
  theme_classic() + theme(axis.text.x = element_text(angle = 45,
  hjust = 1, vjust = 1))

```



We decompose the changes in cell type proportions using a glm with poisson outcome that accounts for covariates and includes random terms on them Model implementation is done by Natsuhiko Kumasaki

```

cv@meta.data$patient_id_sample <- paste0(cv@meta.data$patient_id, ";", cv@meta.data$pool_name) # Using our
Y = table(cv@meta.data$patient_id_sample, cv@meta.data$cell_annot_revision_short)

```

```

mymetadata <- cv@meta.data[!duplicated(cv@meta.data$patient_id_sample),]
mymetadata$Sex <- mymetadata$Sex_pred
mymetadata$Ethnicity <- mymetadata$Ethnicity_pred

metadata <- mymetadata[,c("ageGroup","covid_status","severity_simple","Sex","Ethnicity","patient_id_sample")]
Y <- Y[rownames(Y)%in%metadata$patient_id_sample,]

# number of samples / number of cell types
nsamples = nrow(Y)
ncells = ncol(Y)

# repeating the meta data table by the number of cell types
metadataExp=cbind(metadata[rep(match(rownames(Y),as.character(metadata$patient_id_sample)),ncells),],Celltype)

res.prop=glmer(I(c(Y))~
(1|Celltype)
+(1|patient_id_sample)
+(1|Sex)
+(1|covid_status)
+(1|Ethnicity)
+(1|ageGroup)
+(1|paedOrAdult)

+(1|patient_id_sample:Celltype)
+(1|Sex:Celltype)
+(1|paedOrAdult:covid_status:Celltype)
+(1|Ethnicity:Celltype)
+(1|ageGroup:Celltype)
,
family=poisson,data=metadataExp,control=glmerControl(optimizer="bobyqa", optCtrl=list(maxfun=2e5)))

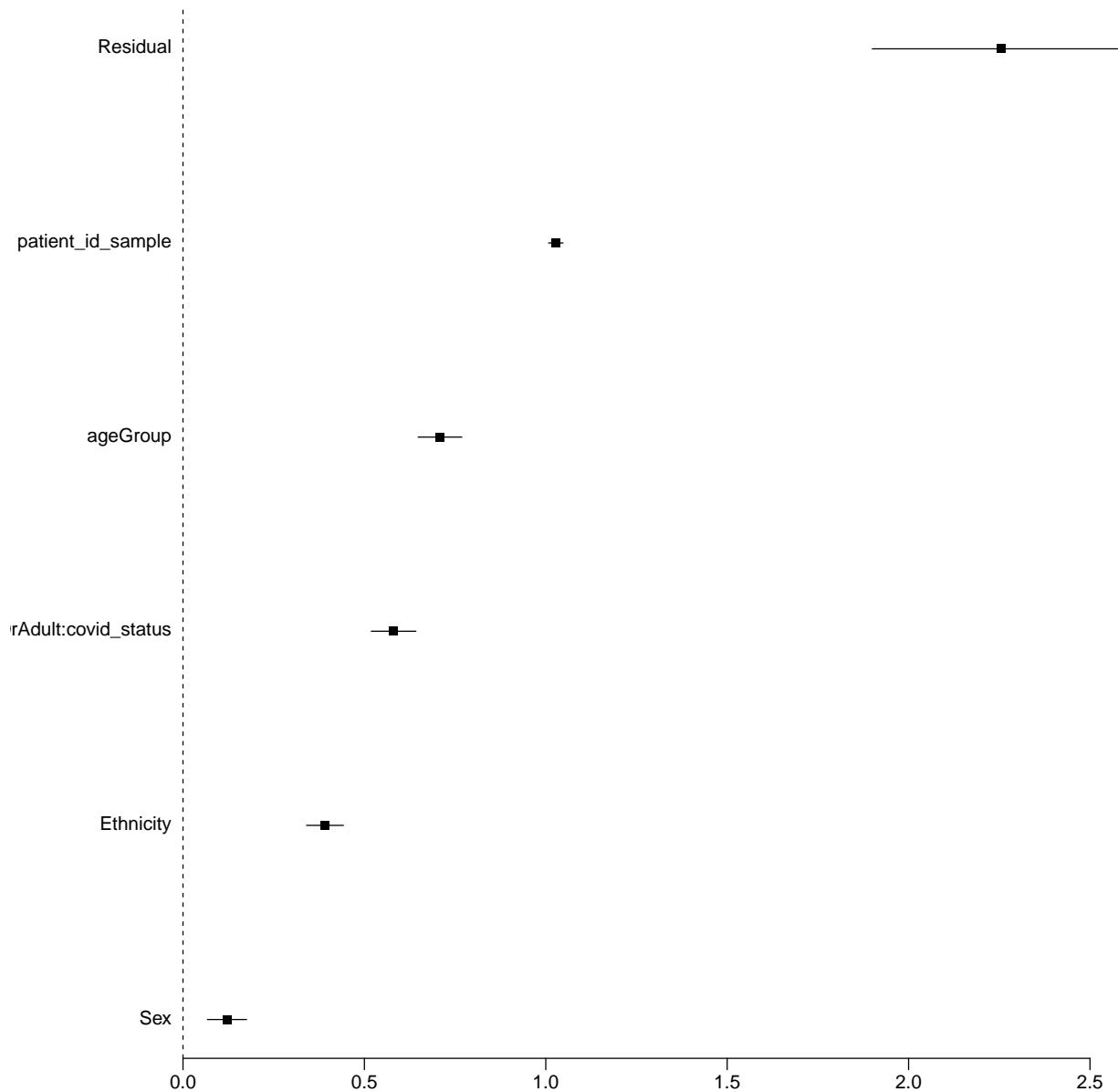
## boundary (singular) fit: see ?isSingular

# standard errors of standard deviations (square root of the variance parameters)
devfun = update(res.prop, devFunOnly=T)
pars = getME(res.prop, c("theta","fixef"))
hess = hessian(devfun, unlist(pars))
sdse.prop = data.frame(sd=unlist(pars), se=sqrt(diag(solve(hess)))) 

# posterior means and their standard deviations
res.prop.ranef = ranef(res.prop)

# Forest plot
rownames(sdse.prop)[rownames(sdse.prop)== "theta.Celltype.(Intercept)"] <- "Residual"
par(mar=c(3,6,1,1),mgp=c(1.2,0.5,0))
Forest(sdse.prop[grep("(Celltype|Residual)",rownames(sdse.prop)),],xlim=c(0,2.5))

```



```

##                                     sd      se
## Residual                         2.2548480 0.18109586
## theta.patient_id_sample:Celltype.(Intercept) 1.0274621 0.01020347
## theta.ageGroup:Celltype.(Intercept)        0.7083200 0.03074516
## theta.paedOrAdult:covid_status:Celltype.(Intercept) 0.5805261 0.03141509
## theta.Ethnicity:Celltype.(Intercept)       0.3913360 0.02596854
## theta.Sex:Celltype.(Intercept)            0.1213297 0.02754274
##                                     x[, 1] - x[, 2] * 1.96
## Residual                         1.89990014
## theta.patient_id_sample:Celltype.(Intercept) 1.00746333
## theta.ageGroup:Celltype.(Intercept)        0.64805945
## theta.paedOrAdult:covid_status:Celltype.(Intercept) 0.51895250
## theta.Ethnicity:Celltype.(Intercept)       0.34043771
## theta.Sex:Celltype.(Intercept)            0.06734593

```

```

##                                     x[, 1] + x[, 2] * 1.96
## Residual                           2.6097959
## theta.patient_id_sample:Celltype.(Intercept) 1.0474609
## theta.ageGroup:Celltype.(Intercept)          0.7685805
## theta.paedOrAdult:covid_status:Celltype.(Intercept) 0.6420997
## theta.Ethnicity:Celltype.(Intercept)         0.4422344
## theta.Sex:Celltype.(Intercept)                0.1753135

postmean = cbind(
  getCondVal(res.prop.ranef,"ageGroup:Celltype",ncells,celltype=colnames(Y))[[1]][,c(6,5,7,3,1,2,4)],
  NA,
  getCondVal(res.prop.ranef,"paedOrAdult:covid_status:Celltype",ncells,celltype=colnames(Y),nfactors = 1)
)

lfsr = cbind(
  getCondVal(res.prop.ranef,"ageGroup:Celltype",ncells,celltype=colnames(Y))[[2]][,c(6,5,7,3,1,2,4)],
  NA,
  getCondVal(res.prop.ranef,"paedOrAdult:covid_status:Celltype",ncells,celltype=colnames(Y),nfactors = 1)
)

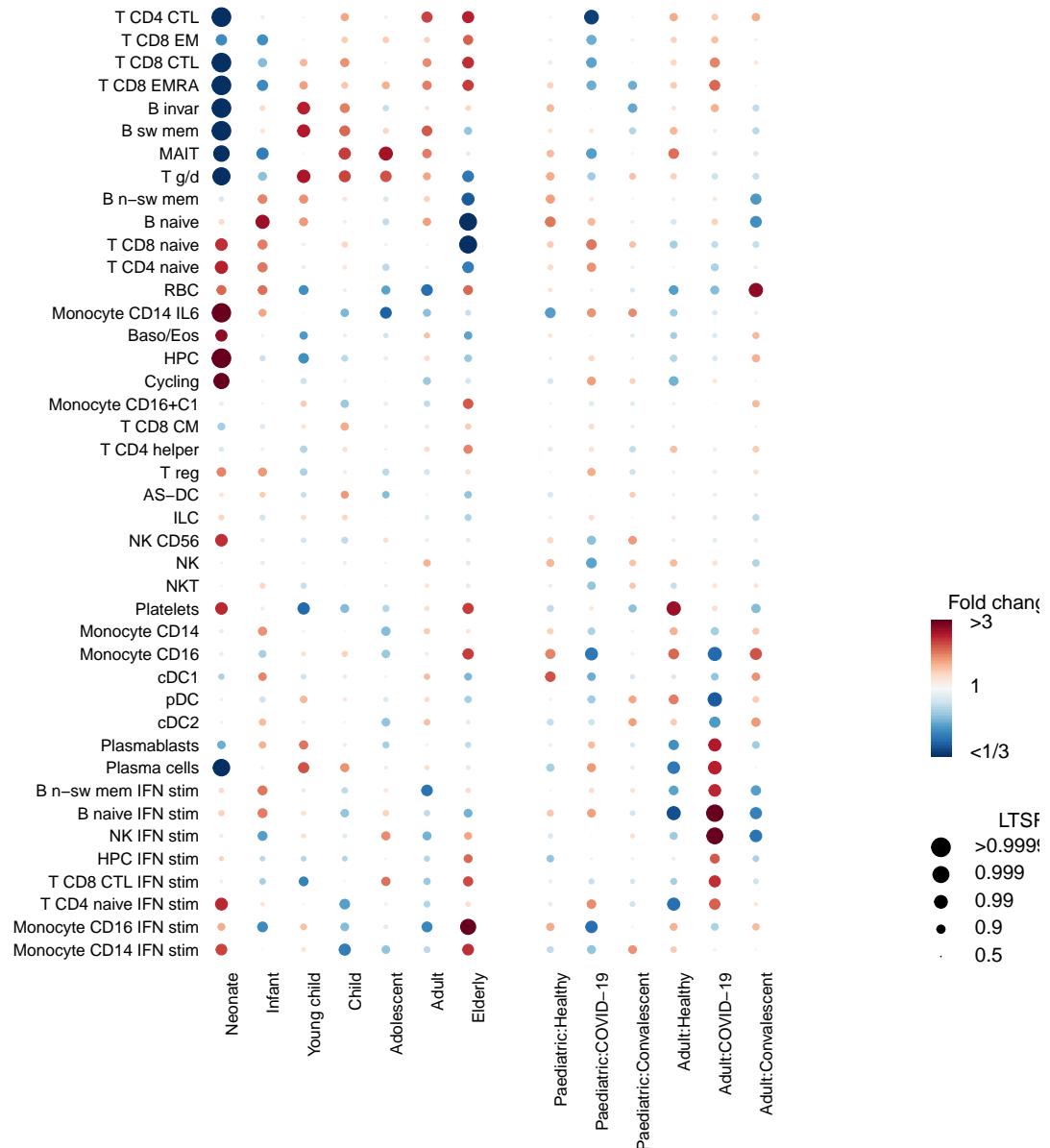
# Dotplot
postmean_oldAgeGroupsPlusSeverity <- postmean
lfsr_oldAgeGroupsPlusSeverity <- lfsr

myClust <- hclust(dist(postmean_oldAgeGroupsPlusSeverity*(1-lfsr_oldAgeGroupsPlusSeverity)),method = "completeness")
postmean_oldAgeGroupsPlusSeverity <- postmean_oldAgeGroupsPlusSeverity[myClust,]
lfsr_oldAgeGroupsPlusSeverity <- lfsr_oldAgeGroupsPlusSeverity[myClust,]
reorderIfn <- order(grep("IFN stim",rownames(postmean_oldAgeGroupsPlusSeverity)),grep("Plasma",rownames(postmean_oldAgeGroupsPlusSeverity)))
postmean_oldAgeGroupsPlusSeverity <- postmean_oldAgeGroupsPlusSeverity[reorderIfn,]
lfsr_oldAgeGroupsPlusSeverity <- lfsr_oldAgeGroupsPlusSeverity[reorderIfn,]

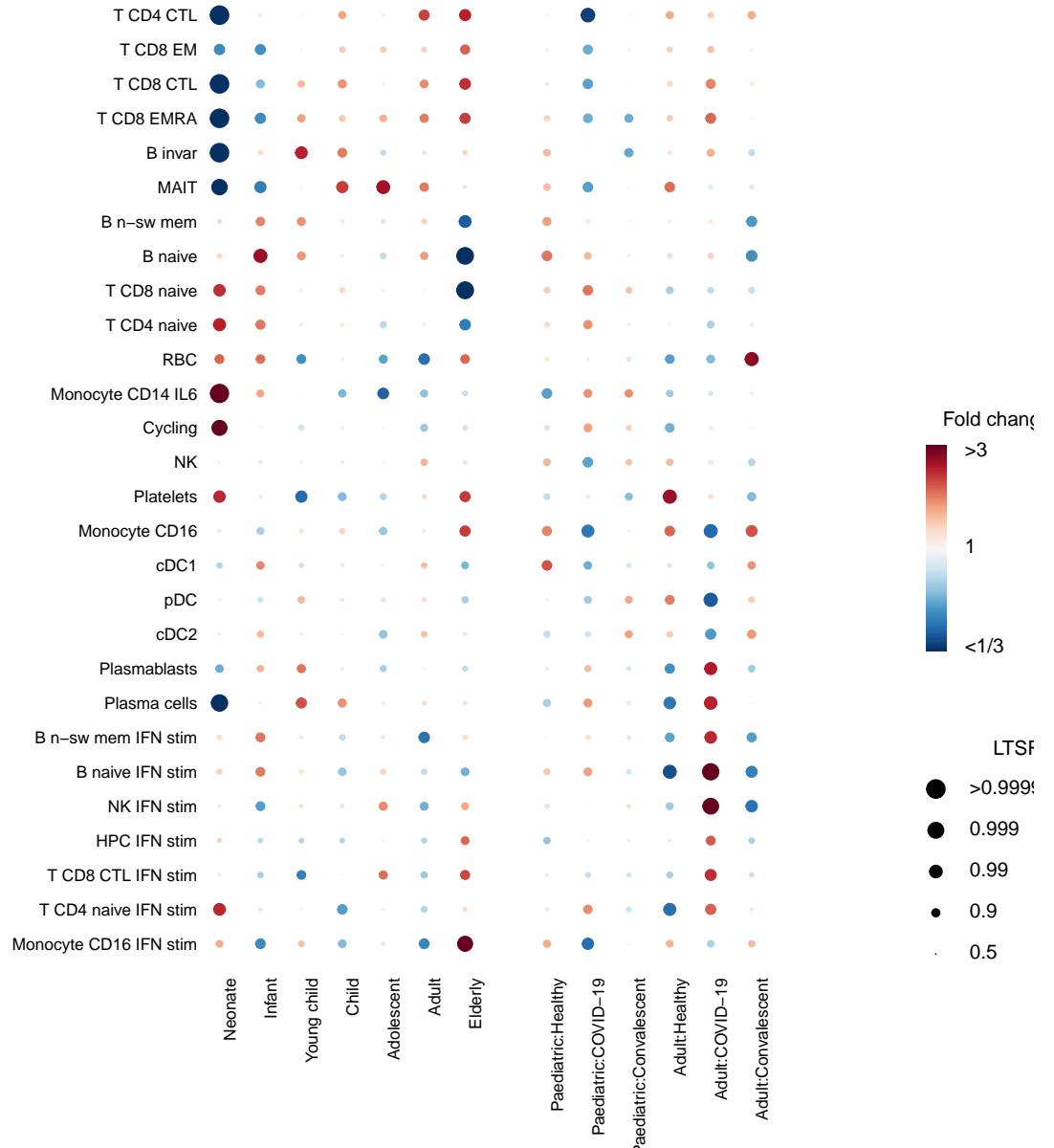
covidSignificant <- apply(lfsr_oldAgeGroupsPlusSeverity[,9:14],1,min)<.1

par(mar=c(8,15,0,10))
Dotplot(postmean_oldAgeGroupsPlusSeverity, SORT=c(F,F),zlim=c(log(1/3),log(3)),ltsr=1-lfsr_oldAgeGroupsPlusSeverity)

```



```
par(mar=c(8,15,0,10))
Dotplot(postmean_oldAgeGroupsPlusSeverity[covidSignificant,], SORT=c(F,F), zlim=c(log(1/3),log(3)), ltsr=
```



```
# 9* Barplots
colsForNewAnnot <- read_rds("/mnt/projects/RL003_allCitePbmcsTheta/dataRevision/colsForNewAnnot2.rds")

ctData <- cv@meta.data
ctData <- ctData[ctData$patient_id != "PP11", ] # PP11 only contains a few dozens of cells
ctData$ageGroup[ctData$covid_status != "Healthy"] <- ctData$paedOrAdult[ctData$covid_status != "Healthy"]
ctData$ageGroup <- ctData$paedOrAdult
ctData$ageAndStatus <- paste0(ctData$ageGroup, " - ", ctData$covid_status)

for (i in unique(ctData$ageAndStatus)) {
  ctData$ageAndStatus[ctData$ageAndStatus == i] <- paste0(ctData$ageAndStatus[ctData$ageAndStatus == i], " (N=", length(unique(ctData$patient_id[ctData$ageAndStatus == i])), ", K=", length(ctData$patient_id[ctData$ageAndStatus ==
```

```

        i]), ")")
}
ctData$ageAndStatus_factor <- factor(ctData$ageAndStatus)

allCellProp <- as.data.frame(matrix(nrow = length(levels(ctData$cell_annot_revision_short)),
  ncol = length(levels(ctData$ageAndStatus_factor))))
rownames(allCellProp) <- levels(ctData$cell_annot_revision_short)
colnames(allCellProp) <- levels(ctData$ageAndStatus_factor)
for (i in levels(ctData$ageAndStatus_factor)) {
  cellProp <- as.data.frame.matrix(table(ctData$cell_annot_revision_short[ctData$ageAndStatus_factor ==
    i], ctData$patient_id[ctData$ageAndStatus_factor == i]))
  cellProp <- apply(cellProp, 2, function(x) x/sum(x))
  # meanCellProp <- MatrixGenerics::rowMedians(cellProp)
  meanCellProp <- rowMeans(cellProp)
  allCellProp[rownames(cellProp), i] <- meanCellProp
}
allCellProp$cell_annot_revision_short <- rownames(allCellProp)
gatAllCellProp <- gather(allCellProp, key = "ageAndStatus_factor",
  value = "cell_type_proportion", -cell_annot_revision_short)

gatAllCellProp$ageAndStatus_factor <- factor(gatAllCellProp$ageAndStatus_factor,
  levels = unique(gatAllCellProp$ageAndStatus_factor[c(grep("Neonate - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Infant - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Young child - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Child - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Adolescent - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Paediatric - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Paediatric - COVID-19",
    gatAllCellProp$ageAndStatus_factor), grep("Paediatric - Convalescent",
    gatAllCellProp$ageAndStatus_factor), grep("Adult - Healthy",
    gatAllCellProp$ageAndStatus_factor), grep("Adult - COVID-19",
    gatAllCellProp$ageAndStatus_factor), grep("Adult - Convalescent",
    gatAllCellProp$ageAndStatus_factor))])))

gatAllCellProp$cell_annot_revision_short_factor <- factor(gatAllCellProp$cell_annot_revision_short,
  levels = rev(levels(cv$cell_annot_revision_short)))
ggplot(gatAllCellProp, aes(x = ageAndStatus_factor, y = cell_type_proportion,
  fill = cell_annot_revision_short_factor)) + geom_bar(position = "fill",
  stat = "identity", width = 0.8, colour = "black") + theme(aspect.ratio = 1.5) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
    hjust = 1, size = 10)) + theme(axis.text.y = element_text(size = 10)) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
  theme(legend.text = element_text(size = 10)) + scale_fill_manual(values = colsForNewAnnot) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(), axis.line = element_line(colour = "black"))

```

