library(dplyr)

library(Seurat)

library(patchwork)

# download datasets

Sample1 <- readRDS('GSM5575227\_NGS19\_I415\_Dim\_LBA\_Hum3.seuratObject.rds')

Sample2 <- readRDS('GSM5575231\_NGS19\_J142\_Dim\_LBA\_Hum8.seuratObject.rds')

Sample3 <- readRDS('GSM5575233\_NGS19\_J264\_Dim\_LBA\_Hum10.seuratObject.rds')

Sample4 <- readRDS('GSM5575232\_NGS19\_J263\_Dim\_LBA\_Hum9.seuratObject.rds')

Sample5 <- readRDS('GSM5575234\_NGS19\_K359\_Dim\_LBA\_Hum11.seuratObject.rds')

Sample6<- readRDS('GSM5575235\_NGS19\_K360\_Dim\_LBA\_Hum12.seuratObject.rds')

# merge of dataset

# groupA= Healthy, groupB= COPD

groupA <- merge(Sample1, y = c(Sample2, Sample3), add.cell.ids = c("Sample1", " Sample2", " Sample3"))

groupB <- merge(Sample4, y = c(Sample5, Sample6), add.cell.ids = c("Sample4", " Sample5", " Sample6"))

# merage of groups

AM <- merge(groupA, y = c(groupB), add.cell.ids = c("groupA", "groupB"))

# Show QC metrics for the first 5 cells

head(AM@meta.data, 5)

# Visualize QC metrics as a violin plot

VlnPlot(AM, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

# FeatureScatter is typically used to visualize feature-feature relationships, but can be used

# for anything calculated by the object, i.e. columns in object metadata, PC scores etc.

plot1 <- FeatureScatter(AM, feature1 = "nCount\_RNA", feature2 = "percent.mt")

plot2 <- FeatureScatter(AM, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

plot1 + plot2

library(gridExtra)

library(tidyverse)

library(ggplot2)

#QC & filtering process

# This metadata contains Loegeois et al. created cluster information, but we initially calculated and reclustered the cells.

View(AM@meta.data)

# create a sample column

AM$sample <- rownames(AM@meta.data)

View(AM@meta.data)

# split the sample column

AM@meta.data <- separate(AM@meta.data, col = 'sample', into = c('stimulation', 'patient', 'Barcode'), sep = '\_')

View(AM@meta.data)

unique(AM@meta.data$stimulation)

unique(AM@meta.data$patient)

# Caluculate mitcondrial percentage

AM$mitoPercent <- PercentageFeatureSet(AM, pattern='^MT-')

# QC metrics

AM\_QC <- subset(AM, subset = nFeature\_RNA > 200 & nFeature\_RNA < 5000 & percent.mt < 20)

# perform integration to correct for batch effects

obj.list <- SplitObject(AM\_QC, split.by = 'patient')

for(i in 1:length(obj.list)){

obj.list[[i]] <- NormalizeData(object = obj.list[[i]])

obj.list[[i]] <- FindVariableFeatures(object = obj.list[[i]])

}

# Select integration features

features <- SelectIntegrationFeatures(object.list = obj.list)

# Find integration anchors (CCA)

anchors <- FindIntegrationAnchors(object.list = obj.list, anchor.features = features)

# Integrate data

seurat.integrated <- IntegrateData(anchorset = anchors)

# scale data, run PCA, UMAP, and cisualize integrated data

seurat.integrated <- ScaleData(object = seurat.integrated)

seurat.integrated <- RunPCA(object = seurat.integrated)

seurat.integrated <- RunUMAP(object = seurat.integrated, dims = 1:50)

p3 <- DimPlot(seurat.integrated, reduction = 'umap', group.by ='stimulation')

p4 <- DimPlot(seurat.integrated, reduction = 'umap', group.by ='patient')

grid.arrange(p3,p4,ncol = 2)

#Elbow polt

ElbowPlot(seurat.integrated)

#Cluster the cells

seurat.integrated <- FindNeighbors(seurat.integrated, dims = 1:10)

seurat.integrated <- FindClusters(seurat.integrated, resolution = 0.5)

seurat.integrated <- RunUMAP(seurat.integrated, dims = 1:10)

# Clusters in integrated data

p3 <- DimPlot(seurat.integrated, reduction = 'umap', group.by ='stimulation')

p5 <- DimPlot(seurat.integrated, reduction = 'umap', label = TRUE, repel = TRUE)

p3+p5

# Clusters in each stimulation

DimPlot(seurat.integrated, reduction = 'umap', split.by = 'stimulation', label = TRUE, repel = TRUE)

# Clusters in each patients

DimPlot(seurat.integrated, reduction = 'umap', split.by = 'patient', label = TRUE, repel = TRUE)

#Check the markers in each cluster:

VlnPlot(seurat.integrated, features = c("MARCO", "FCGR3A", "CD14", "CD3E", "EPCAM", "MUC1", "MUC4", "TEKT2", "CLEC9A", "CLEC10A", "CD1C"), ncol = 2, pt.size = 0)

# Exclude the Cluster7,12(Lymphocytes, and epithelial cells)

# Reclustering the data into small counts

AM <- subset(x = seurat.integrated, idents = c("6", "11", "12"), invert = TRUE)

AM <- FindNeighbors(AM, dims = 1:6)

AM <- FindClusters(AM, resolution = 0.15)

AM <- RunUMAP(AM, dims = 1:6)

# scale data, run PCA, UMAP, and cisualize AM

AM <- RunPCA(object = AM)

AM <- RunUMAP(object = AM, dims = 1:6)

p6 <- DimPlot(AM, reduction = 'umap', group.by ='stimulation')

p7 <- DimPlot(AM, reduction = 'umap', group.by ='patient')

grid.arrange(p6,p7,ncol = 2)

# Display clusters by data

p8 <- DimPlot(AM, reduction = 'umap', group.by ='stimulation')

p9 <- DimPlot(AM, reduction = 'umap', label = TRUE, repel = TRUE)

p8+p9

#clusters in each stimulation

DimPlot(AM, reduction = 'umap', split.by = 'stimulation', label = TRUE, repel = TRUE)

# Display clusters per patient

DimPlot(AM, reduction = 'umap', split.by = 'patient', label = TRUE, repel = TRUE)

# check the dendritic cells specific gene

VlnPlot(AM, features = c("FCER1A", "CD1A", "CD1C", "CD1E", "CLEC10A", "CD1C"), ncol = 2, pt.size = 0)

# exclude the cluster 4 (it contains DCs)

AM1 <- subset(x = AM, idents = c("4"), invert = TRUE)

AM1 <- FindNeighbors(AM1, dims = 1:6)

AM1 <- FindClusters(AM1, resolution = 0.12)

AM1 <- RunUMAP(AM1, dims = 1:6)

# scale data, run PCA, UMAP, and cisualize AM

AM1 <- RunPCA(object = AM1)

AM1 <- RunUMAP(object = AM1, dims = 1:10)

p10 <- DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', group.by ='stimulation')

p11 <- DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', group.by ='patient')

grid.arrange(p10,p11,ncol = 2)

p12 <- DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', group.by ='stimulation')

p13 <- DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', label = TRUE, repel = TRUE)

p12+p13

DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', split.by = 'stimulation', label = TRUE, repel = TRUE)

DimPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), reduction = 'umap', split.by = 'patient', label = TRUE, repel = TRUE)

# check the macrophage specific gene

VlnPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), features = c("MARCO", "FCGR3A", "CD14", "CD3E", "EPCAM", "MUC1", "MUC4", "TEKT2", "CLEC9A", "CLEC10A", "CD1C"), ncol = 2, pt.size = 0)

# CigaretteResAM

VlnPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), features = c("FABP4", "LIMA1", "RBP4"), ncol = 2, pt.size = 0)

# moAM specific

VlnPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), features = c("CCR2", "SPP1", "CCL2", "S100A8", "CD14"), ncol = 2, pt.size = 0)

# cellcycle specific

VlnPlot(AM1, cols = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), features = c("CCNB1", "CDK1", "PIMREG", "TOP2A", "MKI67"), ncol = 2, pt.size = 0)

# find all markers of clusters

cluster0.markers <- FindMarkers(AM1, only.pos = TRUE, ident.1 = 0, min.pct = 0.25)

head(cluster0.markers, n = 20)

cluster1.markers <- FindMarkers(AM1, only.pos = TRUE, ident.1 = 1, min.pct = 0.25)

head(cluster1.markers, n = 20)

cluster2.markers <- FindMarkers(AM1, only.pos = TRUE, ident.1 = 2, min.pct = 0.25)

head(cluster2.markers, n = 20)

cluster3.markers <- FindMarkers(AM1, only.pos = TRUE, ident.1 = 3, min.pct = 0.25)

head(cluster3.markers, n = 20)

# Identify significant items in each cluster that are significantly different from other clusters.

# Extraction and delineation of characteristic genes that are elevated in each cluster(DoHeatmap)

AM1.markers <- FindAllMarkers(AM1, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

AM1.markers %>%

group\_by(cluster) %>%

slice\_max(n = 2, order\_by = avg\_log2FC)

AM1.markers %>%

group\_by(cluster) %>%

top\_n(n = 10, wt = avg\_log2FC) -> top10

DoHeatmap(AM1, group.colors = c('0' = 'darkorange', '1' = 'dodgerblue','2' = 'gold','3' = 'purple'), features = top10$gene) + NoLegend()

# Fitting the above genes to each cluster

DotPlot(AM1, features = c("FCGR1A", "FCGR3A", "MRC1", "CD68", "MARCO", "SPI1", "PPARG", # high expression of core macrophage genes

"RBP4", "FABP4", "ALDH2", "APOL1", "APOE", "MGST1", "LIPA", # Cluster 0-1 upregulated

"ITGAM", "SPP1", "CCL2", "CCR2", "CCR5", "S100B", "LPCAT2", "PLA2G7", # Cluster 2 overexpressed transcripts encoding monocyte lineage-associated molecules

"TOP2A", "CDK1", "CCNB1", "PIMREG", "MKI67" # Cluster 3 upregulated cycling-related genes

), assay = "RNA",

scale.by = "size",

cols = c("dark gray", "blue")) +

theme(axis.text.x = element\_text(angle = 90,

vjust = 0.5,

hjust=1))

# Verification restricted to LPCAT2 and PLA2G7

features2 <- c("LPCAT2", "PLA2G7")

# Dot plots - the size of the dot corresponds to the percentage of cells expressing the

DotPlot(AM1, features = features2, split.by = 'stimulation', assay = "RNA", scale.by = "size", cols = c("blue", "blue", "blue"))+ theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))

# Cell-cycle scoring

# This program confirms that Cluster3 is G2M

s.genes <- cc.genes$s.genes

g2m.genes <- cc.genes$g2m.genes

AM1 <- CellCycleScoring(AM1, s.features = s.genes, g2m.features = g2m.genes, set.ident = TRUE)

# view cell cycle scores and phase assignments

head(AM1[[]])

DimPlot(AM1, cols = c('G1' = 'darkorange', 'G2M' = 'dodgerblue','S' = 'gold'), group.by= "Phase")

DimPlot(AM1, cols = c('G1' = 'darkorange', 'G2M' = 'dodgerblue','S' = 'gold'), group.by= "Phase", split.by = "stimulation")

# view cell cycle scores and phase assignments

head(AM1[[]])

# export of AM1@meta.data

# These data are described in the sourcedata.

# The folder name has been removed from the last code.

export\_df <- AM1@meta.data %>%

rownames\_to\_column("barcodes") %>%

select(seurat\_clusters, stimulation, patient, barcodes, nCount\_RNA, nFeature\_RNA, percent.mt, S.Score, G2M.Score, Phase)

write.csv(export\_df, "/ /file.csv")