## ##CITE-Seq 1 Script - Cluster Analysis####

####Setup####

###Load required packages

library(Seurat)

library(ggplot2)

library(tidyverse)

library(patchwork)

library(Matrix)

library(RColorBrewer)

library(writexl)

library(ggridges)

library(clustree)

library(scRepertoire)

library(future)

library(alakazam)

library(immunarch)

library(airr)

library(biomaRt)

library(SeuratDisk)

library(SeuratData)

###Load Seurat object

experiment <- LoadH5Seurat("SeuratProject.h5Seurat")

###Individual plots

DefaultAssay(experiment) <- "RNA"

p1=DimPlot(experiment, label = TRUE,cols=colbig,reduction = "rna.umap", label.size = 2.5) + NoLegend()

p2=DimPlot(experiment, label = TRUE,cols=colbig,reduction = "adt.umap", label.size = 2.5) + NoLegend()

p3=DimPlot(experiment, label = TRUE,cols=colbig, reduction = "wnn.umap", label.size = 2.5) + NoLegend()

p1

p2

p3

DefaultAssay(experiment) <- "RNA"

DefaultAssay(experiment) <- "ADT"

####Simple plotting####

###Umap-wnn by mouse

plot\_mouse <- DimPlot(experiment, label = TRUE,reduction = "wnn.umap", label.size = 2.5, group.by = "orig.ident") + ggtitle("Coloured by mouse")

###Umap-wnn by sample

DimPlot(experiment, label = TRUE,cols=colbig, reduction = "wnn.umap", label.size = 2.5, split.by = "orig.ident", ncol = 2) + NoLegend()

###Umap-wnn by cell cycle stage

DimPlot(experiment, label = TRUE,reduction = "wnn.umap", label.size = 2.5, group.by = "Phase") + ggtitle("Coloured by cell cycle stage")

###Feature and violin plot

FeaturePlot(experiment, features = c("CXCR5"), reduction = "wnn.umap")

VlnPlot(experiment, feature = "IgM")

####Finding all the markers####

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

experiment.markers %>%

group\_by(cluster) %>%

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

DoHeatmap(experiment, features = top10$gene) + NoLegend()

####DEG of each cluster####

###Find DEGs

Cluster\_2 <- FindMarkers(experiment, ident.1 = 2, assay = "RNA")

Cluster\_2\_adt <- FindMarkers(experiment, ident.1 = 2, assay = "ADT")

###Export Cluster-associated gene lists

Cluster\_2\_exp <- tibble::rownames\_to\_column(Cluster\_2, "Genes")

write\_xlsx(Cluster\_2\_exp, "\\Cluster\_2\_exp.xlsx")

####TFIDF####

###Define the function

tfidf = function(data,target,universe){

if(!all(target %in% universe))

stop('Target must be a subset of universe')

nObs = Matrix::rowSums(data[,target,drop=FALSE]>0)

nTot = Matrix::rowSums(data[,universe,drop=FALSE]>0)

tf = nObs/length(target)

idf = log(length(universe)/nTot)

score = tf\*idf

#Calculate p-value for significance based on using a hypergeometric distribution to simulate the results of infinite random sampling

pvals = phyper(nObs-1,nTot,length(universe)-nTot,length(target),lower.tail=FALSE)

qvals = p.adjust(pvals,method='BH')

ntf = (exp(-idf)\*length(universe)-tf\*length(target))/(length(universe)-length(target))

return(data.frame(geneFrequency=tf,

geneFrequencyOutsideCluster=ntf,

geneFrequencyGlobal=exp(-idf),

geneExpression=Matrix::rowMeans(data[,target,drop=FALSE]),

geneExpressionOutsideCluster = Matrix::rowMeans(data[,universe[!(universe%in%target)],drop=FALSE]),

geneExpressionGlobal = Matrix::rowMeans(data),

idf=idf,

tfidf=score,

qval=qvals)[order(score,decreasing=TRUE),])

}

###Select cluster

TFIDF.c2 <- WhichCells(object = experiment, ident = 2)

###Set to RNA assay

DefaultAssay(experiment)<-"RNA"

##Run function - Higher TFIDF score and lower q value genes are more uniquely expressed

TFIDF.c2.genes <- tfidf(GetAssayData(experiment), TFIDF.c2, colnames(experiment))

###Export table

TFIDF\_cluster\_2 <- tibble::rownames\_to\_column(TFIDF.c2.genes, "Genes")

write\_xlsx(TFIDF\_cluster\_2, "\\TFIDF\_cluster\_2.xlsx")

####Addmodulescore####

###Load gene signature file

Hallmark\_sig <- read.csv("Population\_signatures/Hallmark\_signatures.csv",header = T, sep = ',')

###Load individual signature lists

Apoptosis\_list <- list(Hallmark\_sig$Apoptosis)

PI3Ksig\_list <- list(Hallmark\_sig$PI3K\_AKT\_MTOR\_Signalling)

Cholesterol\_list <- list(Hallmark\_sig$Cholesterol\_Homeostasis)

IL2sig\_list <- list(Hallmark\_sig$IL2\_STAT5\_Signalling)

IL6sig\_list <- list(Hallmark\_sig$IL6\_JAK\_STAT3\_Signalling)

###convertMouseGeneList - function

convertHumanGeneList <- function(x){

require("biomaRt")

human = useMart("ensembl", dataset = "hsapiens\_gene\_ensembl")

mouse = useMart("ensembl", dataset = "mmusculus\_gene\_ensembl")

genesV2 = getLDS(attributes = c("hgnc\_symbol"), filters = "hgnc\_symbol", values = x , mart = human, attributesL = c("mgi\_symbol"), martL = mouse, uniqueRows=T)

mousex <- unique(genesV2[, 2])

# Print the first 6 genes found to the screen

print(head(mousex))

return(mousex)

}

###Convert human to mouse genes

Apoptosis\_list <- convertHumanGeneList(Apoptosis\_list)

lapply(Apoptosis\_list, toupper)

str(Apoptosis\_list)

###Set correct assay

DefaultAssay(experiment) <- "RNA"

##A#ddmodulescore

experiment <-AddModuleScore(experiment, features = Apoptosis\_list, name = "Apoptosis\_enrichment")

experiment <-AddModuleScore(experiment, features = PI3Ksig\_list, name = "PI3Ksig\_enrichment")

experiment <-AddModuleScore(experiment, features = Cholesterol\_list, name = "Cholesterol\_enrichment")

experiment <-AddModuleScore(experiment, features = IL2sig\_list, name = "IL2sig\_enrichment")

experiment <-AddModuleScore(experiment, features = IL6sig\_list, name = "IL6sig\_enrichment")

###Visualise enrichment scores

VlnPlot(experiment, c("IL6sig\_enrichment1"), pt.size = 0.1)+NoLegend()

FeaturePlot(experiment, c("PI3Ksig\_enrichment1"),cols=c("Blue", "Yellow"), reduction = "wnn.umap")

####Subsetting unknown cluster####

Unknown\_cells <- subset(experiment, idents = c(2, 12, 27, 33))

Unknown\_cells <- FindClusters(Unknown\_cells, resolution = 0.8, verbose = FALSE, graph.name = "wsnn")

Unknown\_cells <- RunUMAP(Unknown\_cells, dims = 1:30, reduction.name = "unknown.umap")

DimPlot(Unknown\_cells, label = TRUE, cols=colbig, reduction = "unknown.umap", label.size = 2.5) + NoLegend()

FeaturePlot(Unknown\_cells, "KLS", reduction = "unknown.umap")