library(Seurat)

library(ggplot2)

library(clustree)

library(cowplot)

library(data.table)

library(dplyr)

mito\_genes=rownames(data)[grep("^MT-", rownames(data),ignore.case = T)]

data=PercentageFeatureSet(data, features = mito\_genes, col.name = "percent\_mito")

fivenum([data@meta.data$percent\_mito](mailto:data@meta.data$percent_mito))

ribo\_genes=rownames(data)[grep("^Rp[sl]", rownames(data),ignore.case = T)]

data =PercentageFeatureSet(data, features = ribo\_genes, col.name = "percent\_ribo")

fivenum(data @meta.data$percent\_ribo)

Hb\_genes=rownames(data)[grep("^Hb[^(p)]", rownames(data),ignore.case = T)]

print(Hb\_genes)

data=PercentageFeatureSet(data, features = Hb\_genes,col.name = "percent\_hb")

fivenum(data@meta.data$percent\_hb)

data <- subset(data, subset = nFeature\_RNA > 200 & nFeature\_RNA < 4000 & percent.mt < 20 & nCount\_RNA < 30000)

feats <- c("nFeature\_RNA", "nCount\_RNA")

p1\_filtered=VlnPlot(data, group.by = "orig.ident", features = feats, pt.size = 0, ncol = 2) +

NoLegend()

w=length(unique(data$orig.ident))/3+5;w

ggsave(filename="Vlnplot1\_filtered.pdf",plot=p1\_filtered,width = w,height = 5)

feats <- c("percent\_mito", "percent\_ribo", "percent\_hb")

p2\_filtered=VlnPlot(data, group.by = "orig.ident", features = feats, pt.size = 0, ncol = 3) +

NoLegend()

w=length(unique(data$orig.ident))/2+5;w

ggsave(filename="Vlnplot2\_filtered.pdf",plot=p2\_filtered,width = w,height = 5)

data <- NormalizeData(data,

normalization.method = "LogNormalize",

scale.factor = 1e4)

data <- FindVariableFeatures(data)

p3 <- VariableFeaturePlot(data)

p3

data <- ScaleData(data)

data <- RunPCA(data, features = VariableFeatures(object = data))

VizDimLoadings(data, dims = 1:2, reduction = "pca")

DimPlot(data, reduction = "pca") + NoLegend()

DimHeatmap(data, dims = 1:12, cells = 500, balanced = TRUE)

data <- RunUMAP(data, dims = 1:15)

data <- FindNeighbors(data, dims = 1:15)

for (res in c(0.01, 0.05, 0.1, 0.2, 0.3, 0.5,0.8,1)) {

data=FindClusters(data, #graph.name = "CCA\_snn",

resolution = res, algorithm = 1)

}

colnames(data@meta.data)

apply(data@meta.data[,grep("RNA\_snn",colnames(data@meta.data))],2,table)

p1\_dim=plot\_grid(ncol = 3, DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.0.01") +

ggtitle("louvain\_0.01"), DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.0.1") +

ggtitle("louvain\_0.1"), DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.0.2") +

ggtitle("louvain\_0.2"))

ggsave(plot=p1\_dim, filename="Dimplot\_diff\_resolution\_low.pdf",width = 14)

p1\_dim=plot\_grid(ncol = 3, DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.0.8") +

ggtitle("louvain\_0.8"), DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.1") +

ggtitle("louvain\_1"), DimPlot(data, reduction = "umap", group.by = "RNA\_snn\_res.0.3") +

ggtitle("louvain\_0.3"))

ggsave(plot=p1\_dim, filename="Dimplot\_diff\_resolution\_high.pdf",width = 18)

p2\_tree=clustree(data@meta.data, prefix = "RNA\_snn\_res.")

ggsave(plot=p2\_tree, filename="Tree\_diff\_resolution.pdf")

sel.clust = "RNA\_snn\_res.0.1"

data <- SetIdent(data, value = sel.clust)

table(data@active.ident)

colnames([data@meta.data](mailto:data@meta.data))

genes\_to\_check = c('EPCAM','KRT19','CLDN4',

'PECAM1' , 'CLO1A2', 'VWF',

'CD3D', 'CD3E', 'CD8A', 'CD4','CD2',

'CDH5', 'PECAM1', 'VWF',

'LUM' , 'FGF7', 'MME',

'AIF1', 'C1QC','C1QB','LYZ',

'MKI67', 'STMN1', 'PCNA',

'CPA3' ,'CST3', 'KIT', 'TPSAB1','TPSB2',

'GOS2', 'S100A9','S100A8','CXCL8',

'KLRD1', 'GNLY', 'KLRF1','AREG', 'XCL2','HSPA6',

'MS4A1','CD19', 'CD79A','IGHG1','MZB1', 'SDC1',

'CSF1R', 'CSF3R', 'CD68')

P4 = DotPlot(data, features = unique(genes\_to\_check),

assay='RNA' ) + coord\_flip()

celltype=data.frame(ClusterID=0:12,

celltype= 0:12)

celltype[celltype$ClusterID %in% c(0),2]='CD8+ T cell'

celltype[celltype$ClusterID %in% c(1),2]='T cell'

celltype[celltype$ClusterID %in% c(10),2]='Hepatic stellate cell'

celltype[celltype$ClusterID %in% c(12),2]='Neutrophil'

celltype[celltype$ClusterID %in% c(2),2]='Fibroblast'

celltype[celltype$ClusterID %in% c(4,6),2]='Macrophage'

celltype[celltype$ClusterID %in% c(2),2]='Basal cell'

celltype[celltype$ClusterID %in% c(5),2]='Hepatocyte'

celltype[celltype$ClusterID %in% c(7),2]='Blood vessel endothelial cell'

celltype[celltype$ClusterID %in% c(8),2]='B cell'

celltype[celltype$ClusterID %in% c(9),2]='germ cell-like cell'

celltype[celltype$ClusterID %in% c(0),2]='CD8+ T cell'

celltype[celltype$ClusterID %in% c(1),2]='T cell'

celltype[celltype$ClusterID %in% c(10),2]='Hepatic stellate cell'

celltype[celltype$ClusterID %in% c(12),2]='Neutrophil'

celltype[celltype$ClusterID %in% c(2),2]='Fibroblast'

celltype[celltype$ClusterID %in% c(3),2]='Macrophage'

celltype[celltype$ClusterID %in% c(4,6),2]='Basal cell'

celltype[celltype$ClusterID %in% c(5),2]='Hepatocyte'

celltype[celltype$ClusterID %in% c(7),2]='Blood vessel endothelial cell'

celltype[celltype$ClusterID %in% c(8),2]='B cell'

celltype[celltype$ClusterID %in% c(9),2]='germ cell-like cell'

celltype[celltype$ClusterID %in% c(11),2]='Epithelial cell'

data@meta.data$celltype = "NA"

for(i in 1:nrow(celltype)){

data@meta.data[which(data@meta.data$RNA\_snn\_res.0.1 == celltype$ClusterID[i]),'celltype'] <- celltype$celltype[i]}

table(data@meta.data$celltype)

th=theme(axis.text.x = element\_text(angle = 45,

vjust = 0.5, hjust=0.5))

library(patchwork)

celltype\_umap =DimPlot(data, reduction = "umap",cols = colors,pt.size = 1,

group.by = "celltype",label = T)

library(singleseqgset)

ratio <- table(data$celltype) %>% as.numeric()

pielabel <- paste0(names," (", round(ratio/sum(ratio)\*100,2), "%)")

p <- pie(ratio, labels=pielabel,

radius = 1.0,clockwise=T,

main = "celltype",col = colors)

pie3D(ratio,labels = pielabel,explode = 0.1,

main = "Cell Proption",

height = 0.3,

labelcex = 1,col=colors)

data <- JoinLayers(data)

deg <- FindAllMarkers(data, min.pct = 0.25, logfc.threshold =0.25)

deg\_top5 <- deg %>%

dplyr::group\_by(cluster) %>%

dplyr::top\_n(n = 5, wt = avg\_log2FC)

library(scRNAtoolVis)

p<- averageHeatmap(object = data,

markerGene = deg\_top5$gene,

group.by = "celltype",

gene.order = deg\_top5$gene)

library(irGSEA)

data.final <- irGSEA.score(object = data,assay = "RNA",

slot = "data", seeds = 123, ncores = 8,

min.cells = 3, min.feature = 0,

custom = F, geneset = NULL, msigdb = T,

species = "Homo sapiens", category = "H",

subcategory = NULL, geneid = "symbol",

method = c("AUCell","UCell","singscore","ssgsea", "JASMINE", "viper"),

aucell.MaxRank = NULL, ucell.MaxRank = NULL,

kcdf = 'Gaussian')

result.dge <- irGSEA.integrate(object = data.final,

group.by = "celltype",

method = c("AUCell","UCell","singscore","ssgsea", "JASMINE", "viper"))

irGSEA.upset.plot <- irGSEA.upset(object = result.dge,

method = "RRA",

mode = "intersect",

upset.width = 20,

upset.height = 10,

set.degree = 2,

pt\_size = grid::unit(2, "mm"))

data<-sc.metabolism.Seurat(obj = data, method = "VISION", imputation = F, ncores = 8, metabolism.type = "KEGG")

metabolism.score <- function(data,

method = "VISION", #AUCell,ssGSEA,GSVA

imputation = F,

ncores = 8,

metabolism.type = "KEGG") {

require(Seurat)

require(GSVA)

require(VISION)

require(AUCell)

require(scMetabolism)

data\_type <- class(obj)

if(data\_type == "Seurat"){

countexp<-GetAssayData(obj, layer='counts')

countexp<-as.matrix(countexp)

}else{

countexp <- obj

}

#imputation

if (imputation == F) {

countexp2<-countexp

}

if (imputation == T) {

result.completed <- alra(as.matrix(countexp))

countexp2 <- result.completed[[3]]; row.names(countexp2) <- row.names(countexp)

}

signatures\_KEGG\_metab <- system.file("data", "KEGG\_metabolism\_nc.gmt", package = "scMetabolism")

signatures\_REACTOME\_metab <- system.file("data", "REACTOME\_metabolism.gmt", package = "scMetabolism")

if (metabolism.type == "KEGG") {gmtFile<-signatures\_KEGG\_metab; cat("Your choice is: KEGG\n")}

if (metabolism.type == "REACTOME") {gmtFile<-signatures\_REACTOME\_metab; cat("Your choice is: REACTOME\n")}

#VISION

if (method == "VISION") {

library(VISION)

n.umi <- colSums(countexp2)

scaled\_counts <- t(t(countexp2) / n.umi) \* median(n.umi)

vis <- Vision(scaled\_counts, signatures = gmtFile)

options(mc.cores = ncores)

vis <- analyze(vis)

signature\_exp<-data.frame(t(vis@SigScores))

}

metabolism <- sc.metabolism.score(obj = data, method = 'AUCell')

DotPlot.metabolism(obj = metabolism,

pathway = rownames(metabolism @assays[["METABOLISM"]][["score"]])[1:40],

phenotype = "celltype", norm = "y")

cellchat <- createCellChat(object = data, group.by = "ident", assay = "RNA")

CellChatDB <- CellChatDB.human

CellChatDB.use <- subsetDB(CellChatDB, search = "Secreted Signaling", key = "annotation")

cellchat@DB <- CellChatDB.use

cellchat <- subsetData(cellchat)

future::plan("multisession", workers = 4)

cellchat <- identifyOverExpressedGenes(cellchat)

cellchat <- identifyOverExpressedInteractions(cellchat)

ptm = Sys.time()

cellchat <- computeCommunProb(cellchat, type = "triMean")

cellchat <- filterCommunication(cellchat, min.cells = 10)

cellchat <- aggregateNet(cellchat)

groupSize <- as.numeric(table(cellchat@idents))

par(mfrow = c(1,2), xpd=TRUE)

netVisual\_circle(cellchat@net$count, vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Number of interactions")

netVisual\_circle(cellchat@net$weight, vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Interaction weights/strength")

cellchat <- netAnalysis\_computeCentrality(cellchat, slot.name = "netP")

p1 <- netAnalysis\_signalingRole\_heatmap(cellchat, pattern = "outgoing")

p2 <- netAnalysis\_signalingRole\_heatmap(cellchat, pattern = "incoming")

p1 + p2

library(NMF)

library(ggalluvial)

selectK(cellchat, pattern = "outgoing")

nPatterns = 3

cellchat <- identifyCommunicationPatterns(cellchat, pattern = "outgoing", k = nPatterns)

netAnalysis\_dot(cellchat, pattern = "outgoing")

selectK(cellchat, pattern = "incoming")

nPatterns = 2

cellchat <- identifyCommunicationPatterns(cellchat, pattern = "incoming", k = nPatterns)

netAnalysis\_dot(cellchat, pattern = "incoming")