**Supplementary File 2. The whole code script in this study.**

rm(list = ls())

#install.packages("Seurat")

####逐个处理单细胞数据

library(Seurat)##version 4.0.5

library(dplyr)

library(future)

library(future.apply)

library(DoubletFinder)

plan("multiprocess", workers = 8) ###compute cores

options(future.globals.maxSize = 8000 \* 1024^2)

getwd()

setwd("E:\\宫颈癌单细胞\\单细胞\\1、原始数据处理")

list.files(path = "./",pattern = "^CC")

####SRR781####

SRR781<-Read10X("./CC4/")

SRR781[1:5,1:5]

SRR781\_object<- CreateSeuratObject(counts = SRR781, project = "SRR781", min.cells = 0, min.features = 200)

SRR781\_object[["percent.mt"]] <- PercentageFeatureSet(SRR781\_object, pattern = "^MT-")

hist(SRR781\_object[["percent.mt"]]$percent.mt)

VlnPlot(SRR781\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

plot1 <- FeatureScatter(SRR781\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

plot2 <- FeatureScatter(SRR781\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

CombinePlots(plots = list(plot1,plot2))

SRR781\_object

##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

SRR781\_val<- subset(SRR781\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

SRR781\_val@meta.data[1:5,]

colname<-paste("CC4\_",colnames(SRR781\_val),sep="")

colname

SRR781\_val<-RenameCells(object = SRR781\_val,colname)

#rm(list = c("BC2","bc2\_object"))

SRR781\_val@meta.data[1:5,]

SRR781\_val$tissue\_type<-"Normal"

SRR781\_val <- NormalizeData(SRR781\_val, normalization.method = "LogNormalize", scale.factor = 10000)

SRR781\_val <- FindVariableFeatures(SRR781\_val, selection.method = "vst", nfeatures = 2000)

###scaling the data###

all.genes <- rownames(SRR781\_val)

SRR781\_val <- ScaleData(SRR781\_val,vars.to.regress = c("percent.mt"))

SRR781\_val

###perform linear dimensional reduction###

SRR781\_val <- RunPCA(SRR781\_val, features = VariableFeatures(object = SRR781\_val))

print(SRR781\_val[["pca"]], dims = 1:5, nfeatures = 5)

#dev.off()

VizDimLoadings(SRR781\_val, dims = 1:2, reduction = "pca")

ElbowPlot(SRR781\_val,ndims = 50)

####cluster the cells###

SRR781\_val <- FindNeighbors(SRR781\_val, dims = 1:30)

####Run non-linear dimensional reduction (UMAP/tSNE)###

SRR781\_val <- RunUMAP(SRR781\_val, dims = 1:30)

SRR781\_val<-RunTSNE(SRR781\_val,dims=1:30)

SRR781\_val <- FindClusters(SRR781\_val, resolution = 0.6)

DimPlot(SRR781\_val, reduction = "umap",label = T)

#?DimPlot

DimPlot(SRR781\_val, reduction = "tsne",label = T)

SRR781.markers <- FindAllMarkers(SRR781\_val, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

write.csv(SRR781.markers,"./CC4/CC4.markers.csv")

#marker <- FindMarkers(SRR781\_val, ident.1 = "3",ident.2 = "7",only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

######biomarkers###

###epithelial cell

FeaturePlot(SRR781\_val,features = c("EPCAM","CDH1","KRT7","KRT5","UPK1A","KRT20"),reduction = "tsne",label = T)

##TNK cell

FeaturePlot(SRR781\_val,features = c("CD3D","CD8A","CD4","NCAM1"),reduction = "tsne",label = T)

##endothelial cell

FeaturePlot(SRR781\_val,features = c("VWF","PECAM1"),reduction = "tsne",label = T)

####fibroblasts and pericytes

FeaturePlot(SRR781\_val,features = c("RGS5","COL1A1","PDGFRA","PDGFRB","DES"),reduction = "tsne",label = T)

###B cell and plasma

FeaturePlot(SRR781\_val,features = c("CD19","JCHAIN","CD79A"),reduction = "tsne",label = T)

###myeloid cell

FeaturePlot(SRR781\_val,features = c("CD74","CD14","FCGR3A","S100A8"),reduction = "tsne",label = T)

###mast cell

FeaturePlot(SRR781\_val,features = c("CPA3"),reduction = "tsne",label = T)

Idents(SRR781\_val)<-SRR781\_val$RNA\_snn\_res.0.6

SRR781\_val<-RenameIdents(SRR781\_val,"5"="Epithelial","21"="Epithelial","16"="Epithelial","1"="Fibroblast","22"="Fibroblast","23"="Fibroblast","4"="Pericytes","8"="Pericytes","25"="Pericytes",

 "14"="MAST","10"="B",

 "11"="Plasma\_B","6"="T/NK","3"="T/NK","12"="T/NK","19"="T/NK","9"="Myeloid","17"="Myeloid","18"="Myeloid","0"="Endothelial","2"="Endothelial","7"="Endothelial","13"="Endothelial","15"="Endothelial","24"="Endothelial","20"="Neuron")

DimPlot(SRR781\_val,reduction = "tsne",label = T)

DimPlot(SRR781\_val,reduction = "umap",label = T)

table(Idents(SRR781\_val))###查看有多少个细胞

SRR781\_val$cell\_cluster\_origin<-Idents(SRR781\_val)

SRR781\_val@meta.data[1:5,]

#save.image(file="./CC4/CC4.RData")

#load("./SRR781\_outs/SRR781\_val.RData")

SRR781\_val$cluster<-Idents(SRR781\_val)

saveRDS(SRR781\_val,file = "./CC4/CC4\_val.rds")

SRR781\_val<-readRDS("./SRR81\_/SRR781\_val.rds")

####auto cellular annotation#

#install.packages("shinythemes")

#devtools::install\_github("ggjlab/scHCL")###郭国冀老师团队

library(scHCL)

hcl\_result <- scHCL(scdata = SRR781\_val@assays$RNA@data, numbers\_plot = 1)###time comsuming take several mins，上课时不要运行这个，需要很长时间，建议空闲时跑

cellassign<-table(Idents(SRR781\_val),hcl\_result$scHCL\_probility$`Cell type`)

cellassign

cellassign<-table(SRR781\_val$RNA\_snn\_res.0.6,hcl\_result$scHCL\_probility$`Cell type`)

cellassign

library(stringr)

cell\_predict<-data.frame(hcl\_result$scHCL\_probility)

table(rownames(SRR781\_val@meta.data)==cell\_predict$Cell)

cell\_predict$cell\_predict<-str\_split(cell\_predict$Cell.type,"[.]",simplify = T)[,1]

table(SRR781\_val$RNA\_snn\_res.0.6,cell\_predict$cell\_predict)

table(SRR781\_val$cluster,cell\_predict$cell\_predict)

SRR781\_val$cell\_predict<-cell\_predict$cell\_predict

SRR781\_val@meta.data[1:3,]

####Doublets identification#

library(DoubletFinder)

###完整的分析：前面和建立单细胞Seurat对象一样，因为前面跑过了，这里都省略

if (TRUE){

 SRR781<-Read10X(data.dir = "SRR80\_outs/filtered\_feature\_bc\_matrix/")

 SRR781[1:5,1:5]

 SRR781\_object<- CreateSeuratObject(counts = SRR781, project = "SRR781", min.cells = 0, min.features = 200)

 SRR781\_object[["percent.mt"]] <- PercentageFeatureSet(SRR781\_object, pattern = "^MT-")

 hist(SRR781\_object[["percent.mt"]]$percent.mt)

 VlnPlot(SRR781\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

 plot1 <- FeatureScatter(SRR781\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

 plot2 <- FeatureScatter(SRR781\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

 CombinePlots(plots = list(plot1,plot2))

 SRR781\_object

 ##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

 SRR781\_val<- subset(SRR781\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

 SRR781\_val@meta.data[1:5,]

 colname<-paste("SRR781\_",colnames(SRR781\_val),sep="")

 colname

 SRR781\_val<-RenameCells(object = SRR781\_val,colname)

 #rm(list = c("BC2","bc2\_object"))

 SRR781\_val@meta.data[1:5,]

 SRR781\_val$tissue\_type<-"Normal"

 SRR781\_val <- NormalizeData(SRR781\_val, normalization.method = "LogNormalize", scale.factor = 10000)

 SRR781\_val <- FindVariableFeatures(SRR781\_val, selection.method = "vst", nfeatures = 2000)

 ###scaling the data###

 all.genes <- rownames(SRR781\_val)

 SRR781\_val <- ScaleData(SRR781\_val,vars.to.regress = c("percent.mt"))

 SRR781\_val

 ###perform linear dimensional reduction###

 SRR781\_val <- RunPCA(SRR781\_val, features = VariableFeatures(object = SRR781\_val))

 print(SRR781\_val[["pca"]], dims = 1:5, nfeatures = 5)

 #dev.off()

 VizDimLoadings(SRR781\_val, dims = 1:2, reduction = "pca")

 ElbowPlot(SRR781\_val,ndims = 50)

 ####cluster the cells###

 SRR781\_val <- FindNeighbors(SRR781\_val, dims = 1:30)

 ####Run non-linear dimensional reduction (UMAP/tSNE)###

 SRR781\_val <- RunUMAP(SRR781\_val, dims = 1:30)

 SRR781\_val<-RunTSNE(SRR781\_val,dims=1:30)

 SRR781\_val <- FindClusters(SRR781\_val, resolution = 0.6)

}

SRR781\_val@meta.data[1:5,]

####因为前面跑过，所以直接从这步开始计算###

SRR781\_val\_db<-SRR781\_val

sweep.res.list\_SRR781\_val <- paramSweep\_v3(SRR781\_val\_db, PCs = 1:30)

sweep.stats\_SRR781\_val <- summarizeSweep(sweep.res.list\_SRR781\_val, GT = FALSE)

bcmvn\_SRR781\_val<-find.pK(sweep.stats\_SRR781\_val)

pK\_value <- as.numeric(as.character(bcmvn\_SRR781\_val$pK[bcmvn\_SRR781\_val$BCmetric == max(bcmvn\_SRR781\_val$BCmetric)]))

annotations <- SRR781\_val\_db@meta.data$RNA\_snn\_res.0.6

homotypic.prop <- modelHomotypic(annotations) ####get the estimated number

nExp\_poi <- round(homotypic.prop\*length(SRR781\_val\_db@meta.data$orig.ident))

nExp\_poi.adj <- round(nExp\_poi\*(1-homotypic.prop))

pN\_value <- 0.25

pANN\_value <- paste0("pANN\_",pN\_value,"\_",pK\_value,'\_',nExp\_poi)

SRR781\_val\_db <- doubletFinder\_v3(SRR781\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi, reuse.pANN = FALSE)

SRR781\_val\_db <- doubletFinder(SRR781\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi.adj, reuse.pANN = pANN\_value)

SRR781\_val\_db@meta.data[1:5,]

SRR781\_val$doublets<-SRR781\_val\_db$DF.classifications\_0.25\_0.2\_497

SRR781\_val@meta.data[1:5,]

####saveRDS###

saveRDS(SRR781\_val,file = "./SRR81\_outs/SRR781\_val.rds")

DimPlot(SRR781\_val,reduction = "tsne",group.by = "doublets")

####SRR782####

rm(list = ls())

SRR782<-Read10X(data.dir = "./CC1/")

SRR782[1:5,1:5]

SRR782\_object<- CreateSeuratObject(counts = SRR782, project = "SRR782", min.cells = 0, min.features = 200)

SRR782\_object[["percent.mt"]] <- PercentageFeatureSet(SRR782\_object, pattern = "^MT-")

hist(SRR782\_object[["percent.mt"]]$percent.mt)

VlnPlot(SRR782\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

plot1 <- FeatureScatter(SRR782\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

plot2 <- FeatureScatter(SRR782\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

CombinePlots(plots = list(plot1,plot2))

SRR782\_object

##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

SRR782\_val<- subset(SRR782\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

SRR782\_val@meta.data[1:5,]

colname<-paste("CC1",colnames(SRR782\_val),sep="")

colname

SRR782\_val<-RenameCells(object = SRR782\_val,colname)

#rm(list = c("BC2","bc2\_object"))

SRR782\_val@meta.data[1:5,]

SRR782\_val$tissue\_type<-"Tumor"

SRR782\_val <- NormalizeData(SRR782\_val, normalization.method = "LogNormalize", scale.factor = 10000)

SRR782\_val <- FindVariableFeatures(SRR782\_val, selection.method = "vst", nfeatures = 2000)

###scaling the data###

all.genes <- rownames(SRR782\_val)

SRR782\_val <- ScaleData(SRR782\_val,vars.to.regress = c("percent.mt"))

SRR782\_val

###perform linear dimensional reduction###

SRR782\_val <- RunPCA(SRR782\_val, features = VariableFeatures(object = SRR782\_val))

print(SRR782\_val[["pca"]], dims = 1:5, nfeatures = 5)

#dev.off()

VizDimLoadings(SRR782\_val, dims = 1:2, reduction = "pca")

ElbowPlot(SRR782\_val,ndims = 50)

####cluster the cells###

SRR782\_val <- FindNeighbors(SRR782\_val, dims = 1:30)

####Run non-linear dimensional reduction (UMAP/tSNE)###

SRR782\_val <- RunUMAP(SRR782\_val, dims = 1:30)

SRR782\_val<-RunTSNE(SRR782\_val,dims=1:30)

SRR782\_val <- FindClusters(SRR782\_val, resolution = 0.8)

DimPlot(SRR782\_val, reduction = "umap",label = T)

?DimPlot

DimPlot(SRR782\_val, reduction = "tsne",label = T)

SRR782.markers <- FindAllMarkers(SRR782\_val, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

write.csv(SRR782.markers,"./SRR82\_outs/SRR782.markers.csv")

#marker <- FindMarkers(SRR782\_val, ident.1 = "3",ident.2 = "7",only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

######biomarkers###

###epithelial cell

FeaturePlot(SRR782\_val,features = c("EPCAM","CDH1","KRT7","KRT5","UPK1A","KRT20"),reduction = "tsne",label = T)

##TNK cell

FeaturePlot(SRR782\_val,features = c("CD3D","CD8A","CD4","NCAM1"),reduction = "tsne",label = T)

##endothelial cell

FeaturePlot(SRR782\_val,features = c("VWF","PECAM1"),reduction = "tsne",label = T)

####fibroblasts and pericytes

FeaturePlot(SRR782\_val,features = c("RGS5","COL1A1","PDGFRA","PDGFRB","DES"),reduction = "tsne",label = T)

###B cell and plasma

FeaturePlot(SRR782\_val,features = c("CD19","JCHAIN","CD79A"),reduction = "tsne",label = T)

###myeloid cell

FeaturePlot(SRR782\_val,features = c("CD74","CD14","FCGR3A","S100A8"),reduction = "tsne",label = T)

###mast cell

FeaturePlot(SRR782\_val,features = c("CPA3"),reduction = "tsne",label = T)

Idents(SRR782\_val)<-SRR782\_val$RNA\_snn\_res.0.6

SRR782\_val<-RenameIdents(SRR782\_val,"0"="Epithelial","1"="Epithelial","2"="Epithelial","3"="Epithelial","4"="Epithelial","5"="Epithelial","9"="Epithelial","10"="Epithelial",

 "8"="Fibroblast","6"="T/NK","7"="Myeloid")

DimPlot(SRR782\_val,reduction = "tsne",label = T)

DimPlot(SRR782\_val,reduction = "umap",label = T)

table(Idents(SRR782\_val))###查看有多少个细胞

SRR782\_val$cell\_cluster\_origin<-Idents(SRR782\_val)

SRR782\_val@meta.data[1:5,]

saveRDS(SRR782\_val,file = "./SRR82\_outs/SRR782\_val.rds")

#save.image(file="./SRR782\_outs/SRR782\_val.RData")

#load("./SRR782\_outs/SRR782\_val.RData")

SRR782\_val$cluster<-Idents(SRR782\_val)

####auto cellular annotation#

#install.packages("shinythemes")

#devtools::install\_github("ggjlab/scHCL")###郭国冀老师团队

library(scHCL)

hcl\_result <- scHCL(scdata = SRR782\_val@assays$RNA@data, numbers\_plot = 1)###time comsuming take several mins

cellassign<-table(Idents(SRR782\_val),hcl\_result$scHCL\_probility$`Cell type`)

cellassign

cellassign<-table(SRR782\_val$RNA\_snn\_res.0.6,hcl\_result$scHCL\_probility$`Cell type`)

cellassign

library(stringr)

cell\_predict<-data.frame(hcl\_result$scHCL\_probility)

table(rownames(SRR782\_val@meta.data)==cell\_predict$Cell)

cell\_predict$cell\_predict<-str\_split(cell\_predict$Cell.type,"[.]",simplify = T)[,1]

table(SRR782\_val$RNA\_snn\_res.0.6,cell\_predict$cell\_predict)

table(SRR782\_val$cluster,cell\_predict$cell\_predict)

SRR782\_val$cell\_predict<-cell\_predict$cell\_predict

SRR782\_val@meta.data[1:3,]

####Doublets identification#

library(DoubletFinder)

###完整的分析：前面和建立单细胞Seurat对象一样，因为前面跑过了，这里都省略

if (TRUE){

 SRR782<-Read10X(data.dir = "SRR80\_outs//filtered\_feature\_bc\_matrix/")

 SRR782[1:5,1:5]

 SRR782\_object<- CreateSeuratObject(counts = SRR782, project = "SRR782", min.cells = 0, min.features = 200)

 SRR782\_object[["percent.mt"]] <- PercentageFeatureSet(SRR782\_object, pattern = "^MT-")

 hist(SRR782\_object[["percent.mt"]]$percent.mt)

 VlnPlot(SRR782\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

 plot1 <- FeatureScatter(SRR782\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

 plot2 <- FeatureScatter(SRR782\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

 CombinePlots(plots = list(plot1,plot2))

 SRR782\_object

 ##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

 SRR782\_val<- subset(SRR782\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

 SRR782\_val@meta.data[1:5,]

 colname<-paste("SRR782\_",colnames(SRR782\_val),sep="")

 colname

 SRR782\_val<-RenameCells(object = SRR782\_val,colname)

 #rm(list = c("BC2","bc2\_object"))

 SRR782\_val@meta.data[1:5,]

 SRR782\_val$tissue\_type<-"Normal"

 SRR782\_val <- NormalizeData(SRR782\_val, normalization.method = "LogNormalize", scale.factor = 10000)

 SRR782\_val <- FindVariableFeatures(SRR782\_val, selection.method = "vst", nfeatures = 2000)

 ###scaling the data###

 all.genes <- rownames(SRR782\_val)

 SRR782\_val <- ScaleData(SRR782\_val,vars.to.regress = c("percent.mt"))

 SRR782\_val

 ###perform linear dimensional reduction###

 SRR782\_val <- RunPCA(SRR782\_val, features = VariableFeatures(object = SRR782\_val))

 print(SRR782\_val[["pca"]], dims = 1:5, nfeatures = 5)

 #dev.off()

 VizDimLoadings(SRR782\_val, dims = 1:2, reduction = "pca")

 ElbowPlot(SRR782\_val,ndims = 50)

 ####cluster the cells###

 SRR782\_val <- FindNeighbors(SRR782\_val, dims = 1:30)

 ####Run non-linear dimensional reduction (UMAP/tSNE)###

 SRR782\_val <- RunUMAP(SRR782\_val, dims = 1:30)

 SRR782\_val<-RunTSNE(SRR782\_val,dims=1:30)

 SRR782\_val <- FindClusters(SRR782\_val, resolution = 0.6)

}

SRR782\_val@meta.data[1:5,]

####因为前面跑过，所以直接从这步开始计算###

SRR782\_val\_db<-SRR782\_val

sweep.res.list\_SRR782\_val <- paramSweep\_v3(SRR782\_val\_db, PCs = 1:30)

sweep.stats\_SRR782\_val <- summarizeSweep(sweep.res.list\_SRR782\_val, GT = FALSE)

bcmvn\_SRR782\_val<-find.pK(sweep.stats\_SRR782\_val)

pK\_value <- as.numeric(as.character(bcmvn\_SRR782\_val$pK[bcmvn\_SRR782\_val$BCmetric == max(bcmvn\_SRR782\_val$BCmetric)]))

annotations <- SRR782\_val\_db@meta.data$RNA\_snn\_res.0.6

homotypic.prop <- modelHomotypic(annotations) ####get the estimated number

nExp\_poi <- round(homotypic.prop\*length(SRR782\_val\_db@meta.data$orig.ident))

nExp\_poi.adj <- round(nExp\_poi\*(1-homotypic.prop))

pN\_value <- 0.25

pANN\_value <- paste0("pANN\_",pN\_value,"\_",pK\_value,'\_',nExp\_poi)

SRR782\_val\_db <- doubletFinder\_v3(SRR782\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi, reuse.pANN = FALSE)

SRR782\_val\_db <- doubletFinder(SRR782\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi.adj, reuse.pANN = pANN\_value)

SRR782\_val\_db@meta.data[1:5,]

SRR782\_val$doublets<-SRR782\_val\_db$DF.classifications\_0.25\_0.06\_181

SRR782\_val@meta.data[1:5,]

####saveRDS###

saveRDS(SRR782\_val,file = "./SRR82\_outs/SRR782\_val.rds")

DimPlot(SRR782\_val,reduction = "tsne",group.by = "doublets")

####SRR783####

rm(list = ls())

SRR783<-Read10X(data.dir = "SRR83\_outs/filtered\_feature\_bc\_matrix/")

SRR783[1:5,1:5]

SRR783\_object<- CreateSeuratObject(counts = SRR783, project = "SRR783", min.cells = 0, min.features = 200)

SRR783\_object[["percent.mt"]] <- PercentageFeatureSet(SRR783\_object, pattern = "^MT-")

hist(SRR783\_object[["percent.mt"]]$percent.mt)

VlnPlot(SRR783\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

plot1 <- FeatureScatter(SRR783\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

plot2 <- FeatureScatter(SRR783\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

CombinePlots(plots = list(plot1,plot2))

SRR783\_object

##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

SRR783\_val<- subset(SRR783\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

SRR783\_val@meta.data[1:5,]

colname<-paste("SRR783\_",colnames(SRR783\_val),sep="")

colname

SRR783\_val<-RenameCells(object = SRR783\_val,colname)

#rm(list = c("BC2","bc2\_object"))

SRR783\_val@meta.data[1:5,]

SRR783\_val$tissue\_type<-"Tumor"

SRR783\_val

SRR783\_val <- NormalizeData(SRR783\_val, normalization.method = "LogNormalize", scale.factor = 10000)

SRR783\_val <- FindVariableFeatures(SRR783\_val, selection.method = "vst", nfeatures = 2000)

###scaling the data###

all.genes <- rownames(SRR783\_val)

SRR783\_val <- ScaleData(SRR783\_val,vars.to.regress = c("percent.mt"))

SRR783\_val

###perform linear dimensional reduction###

SRR783\_val <- RunPCA(SRR783\_val, features = VariableFeatures(object = SRR783\_val))

print(SRR783\_val[["pca"]], dims = 1:5, nfeatures = 5)

#dev.off()

VizDimLoadings(SRR783\_val, dims = 1:2, reduction = "pca")

ElbowPlot(SRR783\_val,ndims = 50)

####cluster the cells###

SRR783\_val <- FindNeighbors(SRR783\_val, dims = 1:30)

####Run non-linear dimensional reduction (UMAP/tSNE)###

SRR783\_val <- RunUMAP(SRR783\_val, dims = 1:30)

SRR783\_val<-RunTSNE(SRR783\_val,dims=1:30)

SRR783\_val <- FindClusters(SRR783\_val, resolution = 0.6)

DimPlot(SRR783\_val, reduction = "umap",label = T)

?DimPlot

DimPlot(SRR783\_val, reduction = "tsne",label = T)

SRR783.markers <- FindAllMarkers(SRR783\_val, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

write.csv(SRR783.markers,"./SRR83\_outs/SRR783.markers.csv")

#marker <- FindMarkers(SRR783\_val, ident.1 = "3",ident.2 = "7",only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

######biomarkers###

###epithelial cell

FeaturePlot(SRR783\_val,features = c("EPCAM","CDH1","KRT7","KRT5","UPK1A","KRT20"),reduction = "tsne",label = T)

##TNK cell

FeaturePlot(SRR783\_val,features = c("CD3D","CD8A","CD4","NCAM1"),reduction = "tsne",label = T)

##endothelial cell

FeaturePlot(SRR783\_val,features = c("VWF","PECAM1"),reduction = "tsne",label = T)

####fibroblasts and pericytes

FeaturePlot(SRR783\_val,features = c("RGS5","COL1A1","PDGFRA","PDGFRB","DES"),reduction = "tsne",label = T)

###B cell and plasma

FeaturePlot(SRR783\_val,features = c("CD19","JCHAIN","CD79A"),reduction = "tsne",label = T)

###myeloid cell

FeaturePlot(SRR783\_val,features = c("CD74","CD14","FCGR3A","S100A8"),reduction = "tsne",label = T)

###mast cell

FeaturePlot(SRR783\_val,features = c("CPA3"),reduction = "tsne",label = T)

Idents(SRR783\_val)<-SRR783\_val$RNA\_snn\_res.0.6

SRR783\_val<-RenameIdents(SRR783\_val,"0"="Epithelial","1"="Epithelial","2"="Epithelial","3"="Epithelial","5"="Epithelial","6"="Epithelial","7"="Epithelial","12"="Epithelial",

 "13"="Fibroblast","9"="Pericytes","14"="MAST","15"="B",

 "4"="T/NK","10"="T/NK","8"="Myeloid","11"="Endothelial")

DimPlot(SRR783\_val,reduction = "tsne",label = T)

DimPlot(SRR783\_val,reduction = "umap",label = T)

table(Idents(SRR783\_val))###查看有多少个细胞

SRR783\_val$cell\_cluster\_origin<-Idents(SRR783\_val)

SRR783\_val@meta.data[1:5,]

saveRDS(SRR783\_val,file = "./SRR83\_outs/SRR783\_val.rds")

#save.image(file="./SRR783\_outs/SRR783\_val.RData")

#load("./SRR783\_outs/SRR783\_val.RData")

SRR783\_val$cluster<-Idents(SRR783\_val)

SRR783\_val<-SRR783\_val[,1:3000]

####auto cellular annotation#

#install.packages("shinythemes")

#devtools::install\_github("ggjlab/scHCL")###郭国冀老师团队

library(scHCL)

hcl\_result <- scHCL(scdata = SRR783\_val@assays$RNA@data, numbers\_plot = 1)###time comsuming take several mins

cellassign<-table(Idents(SRR783\_val),hcl\_result$scHCL\_probility$`Cell type`)

cellassign

cellassign<-table(SRR783\_val$RNA\_snn\_res.0.6,hcl\_result$scHCL\_probility$`Cell type`)

cellassign

library(stringr)

cell\_predict<-data.frame(hcl\_result$scHCL\_probility)

table(rownames(SRR783\_val@meta.data)==cell\_predict$Cell)

cell\_predict$cell\_predict<-str\_split(cell\_predict$Cell.type,"[.]",simplify = T)[,1]

table(SRR783\_val$RNA\_snn\_res.0.6,cell\_predict$cell\_predict)

table(SRR783\_val$cluster,cell\_predict$cell\_predict)

SRR783\_val$cell\_predict<-cell\_predict$cell\_predict

SRR783\_val@meta.data[1:3,]

####Doublets identification#

library(DoubletFinder)

###完整的分析：前面和建立单细胞Seurat对象一样，因为前面跑过了，这里都省略

if (TRUE){

 SRR783<-Read10X(data.dir = "SRR80\_outs//filtered\_feature\_bc\_matrix/")

 SRR783[1:5,1:5]

 SRR783\_object<- CreateSeuratObject(counts = SRR783, project = "SRR783", min.cells = 0, min.features = 200)

 SRR783\_object[["percent.mt"]] <- PercentageFeatureSet(SRR783\_object, pattern = "^MT-")

 hist(SRR783\_object[["percent.mt"]]$percent.mt)

 VlnPlot(SRR783\_object, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

 plot1 <- FeatureScatter(SRR783\_object, feature1 = "nCount\_RNA", feature2 = "percent.mt")

 plot2 <- FeatureScatter(SRR783\_object, feature1 = "nCount\_RNA", feature2 = "nFeature\_RNA")

 CombinePlots(plots = list(plot1,plot2))

 SRR783\_object

 ##select the cells with 300 genes at least and 4000 at most, the percent of mitochondrion genes is less than 10%

 SRR783\_val<- subset(SRR783\_object, subset = nFeature\_RNA > 200 & nFeature\_RNA < 7000 & percent.mt < 20)

 SRR783\_val@meta.data[1:5,]

 colname<-paste("SRR783\_",colnames(SRR783\_val),sep="")

 colname

 SRR783\_val<-RenameCells(object = SRR783\_val,colname)

 #rm(list = c("BC2","bc2\_object"))

 SRR783\_val@meta.data[1:5,]

 SRR783\_val$tissue\_type<-"Normal"

 SRR783\_val <- NormalizeData(SRR783\_val, normalization.method = "LogNormalize", scale.factor = 10000)

 SRR783\_val <- FindVariableFeatures(SRR783\_val, selection.method = "vst", nfeatures = 2000)

 ###scaling the data###

 all.genes <- rownames(SRR783\_val)

 SRR783\_val <- ScaleData(SRR783\_val,vars.to.regress = c("percent.mt"))

 SRR783\_val

 ###perform linear dimensional reduction###

 SRR783\_val <- RunPCA(SRR783\_val, features = VariableFeatures(object = SRR783\_val))

 print(SRR783\_val[["pca"]], dims = 1:5, nfeatures = 5)

 #dev.off()

 VizDimLoadings(SRR783\_val, dims = 1:2, reduction = "pca")

 ElbowPlot(SRR783\_val,ndims = 50)

 ####cluster the cells###

 SRR783\_val <- FindNeighbors(SRR783\_val, dims = 1:30)

 ####Run non-linear dimensional reduction (UMAP/tSNE)###

 SRR783\_val <- RunUMAP(SRR783\_val, dims = 1:30)

 SRR783\_val<-RunTSNE(SRR783\_val,dims=1:30)

 SRR783\_val <- FindClusters(SRR783\_val, resolution = 0.6)

}

SRR783\_val@meta.data[1:5,]

####因为前面跑过，所以直接从这步开始计算###

SRR783\_val\_db<-SRR783\_val

sweep.res.list\_SRR783\_val <- paramSweep\_v3(SRR783\_val\_db, PCs = 1:30)

sweep.stats\_SRR783\_val <- summarizeSweep(sweep.res.list\_SRR783\_val, GT = FALSE)

bcmvn\_SRR783\_val<-find.pK(sweep.stats\_SRR783\_val)

pK\_value <- as.numeric(as.character(bcmvn\_SRR783\_val$pK[bcmvn\_SRR783\_val$BCmetric == max(bcmvn\_SRR783\_val$BCmetric)]))

annotations <- SRR783\_val\_db@meta.data$RNA\_snn\_res.0.6

homotypic.prop <- modelHomotypic(annotations) ####get the estimated number

nExp\_poi <- round(homotypic.prop\*length(SRR783\_val\_db@meta.data$orig.ident))

nExp\_poi.adj <- round(nExp\_poi\*(1-homotypic.prop))

pN\_value <- 0.25

pANN\_value <- paste0("pANN\_",pN\_value,"\_",pK\_value,'\_',nExp\_poi)

SRR783\_val\_db <- doubletFinder\_v3(SRR783\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi, reuse.pANN = FALSE)

SRR783\_val\_db <- doubletFinder(SRR783\_val\_db, PCs = 1:30, pN = pN\_value, pK = pK\_value, nExp = nExp\_poi.adj, reuse.pANN = pANN\_value)

SRR783\_val\_db@meta.data[1:5,]

SRR783\_val$doublets<-SRR783\_val\_db$DF.classifications\_0.25\_0.08\_330

SRR783\_val@meta.data[1:5,]

####saveRDS###

saveRDS(SRR783\_val,file = "./SRR83\_outs/SRR783\_val.rds")

DimPlot(SRR783\_val,reduction = "tsne",group.by = "doublets")

####merge the scRNA-data####

rm(list=ls())

library(Seurat)

library(dplyr)

library(future)

library(future.apply)

library(dplyr)

library(msigdbr)

library(clusterProfiler)

plan("multiprocess", workers = 8) ###set the compute core

options(future.globals.maxSize = 12000 \* 1024^2)

getwd()

setwd("~/Documents\_PC/scRNA-seq/Data/10X")

SRR780<-readRDS("./CC4/CC4\_val.rds")

SRR781<-readRDS("./CC1/CC1\_val.rds")

SRR782<-readRDS("./CC2/CC2\_val.rds")

SRR783<-readRDS("./CC3/CC3\_val.rds")

SRR783@meta.data[1:5,]

#####取小的对象###如果电脑内存较小的话只用1000个细胞，这里我们不用

SRR780<-SRR780[,1:1000]

SRR781<-SRR781[,1:1000]

SRR782<-SRR782[,1:1000]

SRR783<-SRR783[,1:1000]

##########

sce.mergeTEN<- merge(SRR780,y=c(SRR781,SRR782,SRR783),project = "scTEN")

sce.mergeTEN@meta.data[1:5,]

rm(SRR780,SRR781,SRR782,SRR783)

gc()

#saveRDS(sce.mergeTEN,file="./sce.mergeTEN\_2.rds")

sce.mergeTEN<-readRDS(file="./sce.mergeTEN.rds")

####integration with harmony####

library(devtools)

#install\_github("immunogenomics/harmony")

library(harmony)

gc()

sce.mergeTEN@meta.data[1:5,]

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

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

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

sce.mergeTEN@meta.data[1:5,]

sce.mergeTEN<-NormalizeData(sce.mergeTEN,verbose = T)

sce.mergeTEN<-FindVariableFeatures(sce.mergeTEN,selection.method = "vst", nfeatures = 2000)

sce.mergeTEN<-ScaleData(sce.mergeTEN,vars.to.regress = c("percent.mt","S.Score","G2M.Score"),verbose = FALSE)

sce.mergeTEN<-RunPCA(sce.mergeTEN,verbose = T,npcs = 50)

ElbowPlot(sce.mergeTEN,ndims = 10)

p1 <- DimPlot(object = sce.mergeTEN, reduction = "pca", pt.size = .1, group.by = "orig.ident")

p2 <- VlnPlot(object = sce.mergeTEN, features = "PC\_1", group.by = "orig.ident", pt.size = .1)

CombinePlots(plots=list(p1,p2))

sce.mergeTEN<-RunHarmony(sce.mergeTEN,group.by.vars = c("orig.ident"), plot\_convergence = TRUE)

harmony\_embeddings <- Embeddings(sce.mergeTEN, 'harmony')

dim(harmony\_embeddings)

p3 <- DimPlot(object = sce.mergeTEN, reduction = "harmony", pt.size = .1, group.by = "orig.ident")

p4 <- VlnPlot(object = sce.mergeTEN, features = "harmony\_1", group.by = "orig.ident", pt.size = .1)

CombinePlots(plots=list(p3,p4))

p1+p3

sce.mergeTEN <- sce.mergeTEN %>%

 RunUMAP(reduction = "harmony", dims = 1:50) %>%

 RunTSNE(reduction = "harmony", dims = 1:50) %>%

 FindNeighbors(reduction = "harmony", dims = 1:50)

sce.mergeTEN<-FindClusters(sce.mergeTEN,resolution = 0.1)

#saveRDS(sce.mergeTEN,file="./sce.mergeTEN\_2.rds")

#sce.mergeTEN<-readRDS(file="./sce.mergeTEN\_2.rds")

table(Idents(sce.mergeTEN))

#Idents(sce\_inter)<-sce\_inter$seurat\_clusters

sce.mergeTEN@meta.data[1:5,]

p1<-DimPlot(sce.mergeTEN,reduction = "tsne",label = T)

p1

DimPlot(sce.mergeTEN,reduction = "umap",label = T)

DimPlot(sce.mergeTEN,reduction = "tsne",label = T,split.by = "tissue\_type")

DimPlot(sce.mergeTEN,reduction = "umap",label = T,split.by = "tissue\_type")

table(sce.mergeTEN@meta.data$cell\_cluster\_origin)

sce.mergeTEN@meta.data[1:5,]

p2<-DimPlot(sce.mergeTEN,reduction = "tsne",group.by = "cluster",label = T)

p2

DimPlot(sce.mergeTEN,reduction = "tsne",group.by = "cluster",label = T)

DimPlot(sce.mergeTEN,group.by = "cell\_predict",reduction = "tsne",label = T)

table(Idents(sce.mergeTEN),sce.mergeTEN$cell\_cluster\_origin)

DimPlot(sce.mergeTEN,reduction = "tsne",label = T)

####FPI

#### LPCAT3、ACSL4、NCOA4、ALOX15、GPX4

 FeaturePlot(sce.mergeTEN,features = c("LPCAT3","ACSL4","NCOA4","ALOX15","GPX4"),reduction = "tsne",label = T)

###epithelial cell

FeaturePlot(sce.mergeTEN,features = c("EPCAM","CDH1","KRT7","KRT5","UPK1A","KRT20"),reduction = "tsne",label = T)

##TNK cell

FeaturePlot(sce.mergeTEN,features = c("CD3D","CD8A","CD4","NCAM1","IL7R"),reduction = "tsne",label = T)

##endothelial cell

FeaturePlot(sce.mergeTEN,features = c("VWF","PECAM1"),reduction = "tsne",label = T)

####fibroblasts and pericytes

FeaturePlot(sce.mergeTEN,features = c("RGS5","COL1A1","PDGFRA","PDGFRB","DES","FAP"),reduction = "tsne",label = T)

###B cell and plasma

FeaturePlot(sce.mergeTEN,features = c("CD19","JCHAIN","CD79A"),reduction = "tsne",label = T)

###myeloid cell

FeaturePlot(sce.mergeTEN,features = c("CD74","CD14","FCGR3A","S100A8"),reduction = "tsne",label = T)

###mast cell

FeaturePlot(sce.mergeTEN,features = c("CPA3"),reduction = "tsne",label = T)

####差异基因

marker <- FindMarkers(sce.mergeTEN, ident.1 = "Tumor",ident.2 = "Normal",group.by="tissue\_type")

sig\_dge.cluster <- subset(marker, p\_val\_adj<0.05&abs(avg\_log2FC)>1)

####marker gene

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

write.csv(sce.mergeTEN.markers,"./sce.mergeTEN.markers.csv")

marker <- FindMarkers(sce.mergeTEN, ident.1 = "21",ident.2 = "8",only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)

c<-table(Idents(sce.mergeTEN),sce.mergeTEN$cell\_predict)

c

write.csv(c,file="../10X/cc.csv")

p1+p2

sce.mergeTEN<-RenameIdents(sce.mergeTEN,"0"="T cell","2"="Epithelial","1"="Smooth muscle cell","6"="Epithelial","3"="Fibroblasts","4"="Endothelial cell","7"="Macrophage","8"="Epithelial",

 "9"="B cell","10"="Lymphocyte","11"="Lymphocyte","5"="Plasma cell")

sce.mergeTEN<-RenameIdents(sce.mergeTEN,"Fibroblasts"="Fibroblasts")

DimPlot(sce.mergeTEN,reduction = "tsne",label = T)

DimPlot(sce.mergeTEN,reduction = "umap",label = T)

#table(Idents(sce.mergeTEN))

DimPlot(sce.mergeTEN,reduction = "umap",label = T,group.by = "celltype")

sce.mergeTEN$Seurat\_harmony<-Idents(sce.mergeTEN)

sce.mergeTEN@meta.data[1:5,]

markers<-FindAllMarkers(sce.mergeTEN,only.pos = T,min.pct = 0.25,logfc.threshold = 0.25)

write.csv(markers,file="./markers.csv")

library(ggsci)

##define the color

??ggsci

#pal\_npg()(11)

cors<-pal\_npg()(7)

cors<-c("#029149","#E0367A","#5D90BA","#431A3D","#FFD121","#D8D155","#223D6C","#D20A13","#f27024")

##Reduction plot##

DimPlot(sce.mergeTEN,reduction="tsne",cols=cors,label = T,pt.size = 0.8)

DimPlot(sce.mergeTEN,reduction="umap",cols=cors)

###By sample

DimPlot(sce.mergeTEN,reduction="tsne",cols = cors,split.by = "orig.ident",ncol = 4)

DimPlot(sce.mergeTEN,reduction="umap",cols = cors,split.by = "orig.ident",ncol = 4)

###By type

sce.mergeTEN$tissue\_type

DimPlot(sce.mergeTEN,reduction="umap",cols = cors,split.by = "tissue\_type")

DimPlot(sce.mergeTEN,reduction="tsne",cols = cors,group.by = "tissue\_type",pt.size = 0.8)

#DimPlot(sce\_inter,reduction="umap",cols = cors,label = F)

#DimPlot(sce\_inter,reduction="umap",cols = cors,split.by = "orig.ident",ncol = 3)

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

write.csv(markers,file="./markers.csv")

####vlnplot and dotplot#####

features = c("CD3G","CD8A","KRT19","KRT7","KRT18","ACTA2","COL1A1","PDGFRA","FAP","VWF","PECAM1","CD14","CD68","CD37","CD74","KIT","JCHAIN")

length(features)

VlnPlot(sce.mergeTEN,features = features[1:6],cols = cors,pt.size = 0)

VlnPlot(sce.mergeTEN,features = features[7:12],cols = cors,pt.size = 0)

VlnPlot(sce.mergeTEN,features = features[12:17],cols = cors,pt.size = 0)

#VlnPlot(sce.mergeTEN,features = features[18:20],cols = cors,pt.size = 0)

VlnPlot(sce.mergeTEN,features = c("CD14","VEGFA"),cols = cors,pt.size = 0)

DotPlot(sce.mergeTEN,features = features,cols = c("grey","darkblue"))

#####distribution plot

library(ggplot2)

table(sce.mergeTEN$tissue\_type,sce.mergeTEN$orig.ident)

tab<-table(Idents(sce.mergeTEN),sce.mergeTEN$orig.ident)

tab

tab<-as.data.frame(tab)

tab

ggplot(tab,aes(x=Var2,y=Freq,fill=Var1))+

 geom\_bar(stat='identity',position='stack',alpha=.5)+

 labs(title='',x='group',y='Cell Number')+

 theme(legend.justification = 'right',

 legend.position = 'right',

 legend.key.height = unit(0.1,'cm'),

 panel.background = element\_blank(),

 axis.line=element\_line(size=0.5,colour="black")

 )+ scale\_fill\_manual(values=cors)

tab<-table(Idents(sce.mergeTEN),sce.mergeTEN$orig.ident)

tab

tab<-prop.table(tab,2)\*100

tab

tab<-as.data.frame(tab)

ggplot(tab,aes(x=Var2,y=Freq,fill=Var1))+

 geom\_bar(stat='identity',position='stack',alpha=.5)+

 labs(title='',x='group',y='Cell Proportion (%)')+

 theme(legend.justification = 'right',

 legend.position = 'right',

 legend.key.height = unit(0.1,'cm'),

 panel.background = element\_blank(),

 axis.line=element\_line(size=0.5,colour="black")

 )+ scale\_fill\_manual(values=cors)

####compare the distribution of CD45+

tab<-table(sce.mergeTEN$orig.ident,Idents(sce.mergeTEN))

tab

tab<-tab[,2:6]

tab

tab<-prop.table(tab,1)\*100

tab

tab<-as.data.frame(tab)

tab

group<-c(rep(c("Normal","Tumor"),5,each=2))

group

tab$group=group

tab

tab[tab$Var2=="T/NK",]

library(ggpubr)

p <- ggboxplot(tab[tab$Var2=="T/NK",], x = "group", y = "Freq",

 color = "group", palette =c("#00AFBB", "#E7B800"),###定义颜色

 add = "jitter", shape = "group")

p

my\_comparisons <- list( c("Tumor","Normal") )

p + stat\_compare\_means(comparisons = my\_comparisons,method = "t.test",paired = F)+ # Add pairwise comparisons p-value

 stat\_compare\_means(label.y = 70,method = "t.test",paired = F)

tab

tab[tab$Var2=="B",]

p <- ggboxplot(tab[tab$Var2=="B",], x = "group", y = "Freq",

 color = "group", palette ="npg",###定义颜色

 add = "jitter", shape = "group")

p

my\_comparisons <- list( c("Tumor","Normal") )

p + stat\_compare\_means(comparisons = my\_comparisons,method = "t.test",paired = F)+ # Add pairwise comparisons p-value

 stat\_compare\_means(label.y = 25,method = "t.test",paired = F)

sce.mergeTEN@meta.data[1:5,]

saveRDS(sce.mergeTEN,file = "./sce.mergeTEN.rds")

save.image(file="./SCE\_TEN.RData")###保存镜像

load(file="./SCE\_TEN.RData")

####第一天课程结束点，请大家联系其余几个datasets的分析流程####

#### 第二天课程开始####

markers<-read.csv(file="./sce.mergeTEN.markers.csv",row.names=1)

#Doheatmap#####

features<-unique(markers$gene[markers$p\_val\_adj<0.0001&markers$avg\_log2FC>1.5]) ####通常p\_val\_adj<0.05,avg\_log2FC>0.25

features

####end####

getwd()

#sce.mergeTEN<-ScaleData(sce.mergeTEN, vars.to.regress = c("percent.mt","S.Score","G2M.Score"))

sce.mergeTEN\_DR<- ScaleData(sce.mergeTEN, vars.to.regress = c("percent.mt","S.Score","G2M.Score"),features = features)

pdf(file="DO\_heatmap.pdf",width = 8,height = 10)

DoHeatmap(sce.mergeTEN\_DR,features = features,group.colors = cors)###It take several mins####

dev.off()

#cors

####matrix

genemeanMatrix<-AverageExpression(sce.mergeTEN)

genemeanMatrix<-genemeanMatrix$RNA

features\_to\_anno = c("CD3G","CD8A","KRT19","KRT7","KRT18","ACTA2","COL1A1","PDGFRA","FAP","VWF","PECAM1","CD14","CD68","CD37","CD74","KIT","JCHAIN")

length(features\_to\_anno)

geneMatrix<-genemeanMatrix[features\_to\_anno,]

###rownames

cc<-rownames(geneMatrix)

cc

#cc[which(!(cc%in%features\_to\_anno))]<-""

#cc

table(Idents(sce.mergeTEN))

levels(Idents(sce.mergeTEN))##知道细胞亚群包含哪些

####information for columns

annotation\_col = data.frame(

 CellType = c("T cell","Epithelial","Smooth muscle cell","Fibroblasts","Endothelial cell","Macrophage","B cell","Lymphocyte","Plasma cell")

)

rownames(annotation\_col) = colnames(geneMatrix)

####define the seq

annotation\_col$CellType<-factor(annotation\_col$CellType,levels= c("T cell","Epithelial","Smooth muscle cell","Fibroblasts","Endothelial cell","Macrophage","B cell","Lymphocyte","Plasma cell"))

levels(Idents(sce.mergeTEN))

cors

paste(c("T cell","Epithelial","Smooth muscle cell","Fibroblasts","Endothelial cell","Macrophage","B cell","Lymphocyte","Plasma cell"),cors,sep='"="')

####color to use in heatmap

ann\_colors = list(

 CellType = c("T cell"="#029149" , "Epithelial"="#E0367A" ,

"Smooth muscle cell"="#5D90BA" ,"Fibroblasts"="#431A3D",

"Endothelial cell"="#FFD121" , "Macrophage"="#D8D155" ,

"B cell"="#223D6C" , "Lymphocyte"="#D20A13",

"Plasma cell"="#f27024" )

 #GeneClass = c("CD4+" = "#7570B3", "CD8+" = "#E7298A","NK" = "#66A61E")

)

?pheatmap

library(RColorBrewer)

#dev.off()

#dev.new()

ComplexHeatmap::pheatmap(as.matrix(geneMatrix), annotation\_col = annotation\_col,cluster\_rows = F,labels\_row = cc,

 annotation\_colors = ann\_colors,cluster\_cols = F,scale = "row",color = colorRampPalette(rev(brewer.pal(n = 11, name ="RdYlBu")))(50))

#gaps\_row = c(8, 14) )

#####GSva analysis######

#####GSEA analysis####

library(devtools)

install\_github("arc85/singleseqgset")##https://arc85.github.io/singleseqgset/articles/singleseqgset.html

#library(singleseqgset)

library(singleseqgset)

?msigdbr

h.human <- msigdbr(species="Homo sapiens",category="H")

h.names <- unique(h.human$gs\_name)

h.sets <- vector("list",length=length(h.names))

h.names

names(h.sets) <- h.names

for (i in names(h.sets)) {

 h.sets[[i]] <- pull(h.human[h.human$gs\_name==i,"gene\_symbol"])

}

h.sets

#expr.mat=sce.mergeTEN@assays$RNA@data

expr.mat[1:5,1:5]

sce.mergeTEN@meta.data[1:5,]

###内存小的话，跑小样本

expr.mat=sce.mergeTEN@assays$RNA@data#[,c(1:1000)]###如果用全部细胞胞，则不要取前面1000个细胞

expr.mat[1:5,1:5]

group=sce.mergeTEN$Seurat\_harmony#[1:1000]##如果用全部细胞胞，则不要取前面1000个细胞

group

####计算每组相对于其他组的显著差异基因

logfc.data <- logFC(cluster.ids=sce.mergeTEN$Seurat\_harmony,expr.mat=expr.mat)

#logfc.data <- logFC(cluster.ids=sce.mergeTEN$Seurat\_harmony[1:1000],expr.mat=expr.mat[,c(1:1000)])

names(logfc.data)

gse.res <- wmw\_gsea(expr.mat=expr.mat,cluster.cells=logfc.data[[1]],log.fc.cluster=logfc.data[[2]],gene.sets=h.sets)

names(gse.res)

res.stats <- gse.res[["GSEA\_statistics"]]

res.stats

ac<-data.frame(cluster=factor(colnames(res.stats)))

ac

rownames(ac)=colnames(res.stats)

ann\_colors = list(

 cluster = c("T cell"="#029149" , "Epithelial"="#E0367A" ,

"Smooth muscle cell"="#5D90BA" ,"Fibroblasts"="#431A3D",

"Endothelial cell"="#FFD121" , "Macrophage"="#D8D155" ,

"B cell"="#223D6C" , "Lymphocyte"="#D20A13",

"Plasma cell"="#f27024" )

 #GeneClass = c("CD4+" = "#7570B3", "CD8+" = "#E7298A","NK" = "#66A61E")

)

#cors

#dev.off()

library(stringr)

rownames(res.stats)<-str\_replace\_all(rownames(res.stats),pattern = "HALLMARK\_",replacement ="")

pheatmap::pheatmap(res.stats,fontsize\_row = 8,annotation\_col = ac,annotation\_legend = F,annotation\_colors = ann\_colors,cluster\_rows = T,

 cluster\_cols = F,scale = "row")

#ggsave("GSEAheatmap.pdf",width = 22,height = 18,units = "cm")

#####compare the differentially expressed genes between the sunclusters#####

table(Idents(sce.mergeTEN))

Idents(sce.mergeTEN)

Bcells<-Idents(sce.mergeTEN)%in%c("B cell","Plasma cell")

table(Bcells)

Bcells<-sce.mergeTEN[,Bcells]

table(Idents(Bcells))

#####compare the gene expression between the cells####

bcmarkers<-FindAllMarkers(Bcells,only.pos = T,min.pct = 0.1,logfc.threshold = 0)

library(dplyr)

library(msigdbr)

library(clusterProfiler)

#m\_t2g <- msigdbr(species = "Homo sapiens", category = "C5",subcategory = "BP")%>%dplyr::select(gs\_name, human\_gene\_symbol)

marker<-bcmarkers[bcmarkers$p\_val\_adj<0.05,]

library(stringr)

#BiocManager::install("org.Hs.eg.db")

#marker$gene<-str\_replace\_all(marker$gene,pattern = "[.]",replacement = "-")

cellID<-bitr(marker$gene,fromType = "SYMBOL",toType = "ENTREZID",OrgDb = "org.Hs.eg.db")

cellID

tmp<-marker$gene%in%cellID$SYMBOL

table(tmp)

markers\_2<-marker[tmp,]

table(markers\_2$gene%in%cellID$SYMBOL)

?left\_join

library(dplyr)

mmm<-left\_join(markers\_2,cellID,by=c("gene"="SYMBOL"))

?compareCluster

xx.formula <- compareCluster(ENTREZID~cluster, data=mmm,

 fun="enrichKEGG")

xx.formula\_GO <- compareCluster(ENTREZID~cluster, data=mmm,OrgDb='org.Hs.eg.db',

 fun="enrichGO")

dotplot(xx.formula,showCategory =20)

dotplot(xx.formula\_GO,showCategory = 10)

dd<-as.data.frame(xx.formula)

dd<-dd[c(2,5,6,15,17,21,22,30,42:48),]

dd$logp<- -1\*log10(dd$pvalue)

#cors[4:6]

library(ggpubr)

#head(dd)

#dd$Description<-paste0(1:15,dd$Description)

dd$cluster<-factor(dd$cluster,levels=c("B","Plasma\_B"))

ggbarplot(dd, x = "Description", y = "logp",

 fill = "cluster", # change fill color by mpg\_level

 color = "white", # Set bar border colors to white

 palette = "jco",#c("#D62728FF", "#8C564BFF"), # jco journal color palett. see ?ggpar

 sort.val = "asc", # Sort the value in descending order

 sort.by.groups = T, # Don't sort inside each group

 x.text.angle = 0, # Rotate vertically x axis texts

 ylab = "-log10(pvalue)",

 legend.title = "KEGG terms",

 rotate = TRUE,

 ggtheme = theme\_classic()

)

save.image(file="./SCE\_TEN.RData")

load(file="./SCE\_TEN.RData")

####merge the scRNA-data####

rm(list=ls())

library(Seurat)

library(dplyr)

library(future)

library(future.apply)

library(monocle)

library(ggsci)

plan("multicore", workers = 4) ###set the compute core

options(future.globals.maxSize = 15000 \* 1024^2)

#getwd()

Myeloid<-readRDS("./CAFsMyeloid.rds")

table(Idents(sce.mergeTEN))

Myeloid<-sce.mergeTEN[,Idents(sce.mergeTEN)%in%c("Fibroblasts")]#Idents(sce.mergeTEN)=="B"｜Idents(sce.mergeTEN)=="Plasma\_B"

rm(sce.mergeTEN)

gc()

###check the

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

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

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

Myeloid@meta.data[1:5,]

Myeloid<-NormalizeData(Myeloid,verbose = T)

Myeloid<-FindVariableFeatures(Myeloid,selection.method = "vst", nfeatures = 2000)

Myeloid<-ScaleData(Myeloid,vars.to.regress = c("percent.mt","S.Score","G2M.Score"),verbose = FALSE)

Myeloid<-RunPCA(Myeloid,verbose = T,npcs = 50)

ElbowPlot(Myeloid,ndims = 50)

p1 <- DimPlot(object = Myeloid, reduction = "pca", pt.size = .1, group.by = "orig.ident")

p2 <- VlnPlot(object = Myeloid, features = "PC\_1", group.by = "orig.ident", pt.size = .1)

CombinePlots(plots=list(p1,p2))

library(harmony)

Myeloid<-RunHarmony(Myeloid,"orig.ident", plot\_convergence = TRUE)

harmony\_embeddings <- Embeddings(Myeloid, 'harmony')

dim(harmony\_embeddings)

p3 <- DimPlot(object = Myeloid, reduction = "harmony", pt.size = .1, group.by = "orig.ident")

p4 <- VlnPlot(object = Myeloid, features = "harmony\_1", group.by = "orig.ident", pt.size = .1)

CombinePlots(plots=list(p3,p4))

p1+p3

Myeloid <- Myeloid %>%

 RunUMAP(reduction = "harmony", dims = 1:50) %>%

 RunTSNE(reduction = "harmony", dims = 1:50) %>%

 FindNeighbors(reduction = "harmony", dims = 1:50)

Myeloid<-FindClusters(Myeloid,resolution = 0.1)

DimPlot(Myeloid,reduction = "tsne",label = T)

DimPlot(Myeloid,reduction = "umap",label = T)

cors<-pal\_npg()(10)

DimPlot(Myeloid,reduction = "tsne",label = T,split.by = "tissue\_type")

###Monocytes

#iCAFs

cors<-c("grey","red")

FeaturePlot(Myeloid,features = c("C7","CNN1"),reduction = "tsne",label = F,cols=cors)

###vCAFs

FeaturePlot(Myeloid,features = c("TNXB","CTGF"),reduction = "tsne",label = F,cols=cors)

##mCAFs

FeaturePlot(Myeloid,features = c("SELENOM","RACK1"),reduction = "tsne",label = F,cols=cors)

Myeloid<-RenameIdents(Myeloid,"1"="mCAFs","3"="mCAFs","5"="iCAFs","7"="mCAFs","8"="vCAFs","4"="iCAFs",

 "9"="mCAFs","0"="vCAFs","6"="iCAFs","2"="mCAFs","10"="mCAFs")

library(ggsci)

cors<-pal\_npg()(2)

cors<-c("#029149","#E0367A","#5D90BA","#431A3D","#FFD121","#D8D155","#223D6C","#D20A13","#f27024",cors)

##Reduction plot##

DimPlot(Myeloid,reduction="tsne",cols=cors)

DimPlot(Myeloid,reduction="umap",group.by="RNA\_snn\_res.0.1",cols=cors)

#Myeloid<-readRDS("./Myeloid.rds")

mmt<-Myeloid[,Idents(Myeloid)%in%c("mCAFs","iCAFs","vCAFs")]

###monocle analysis of the trajectory###

library(monocle)

##get the monoand macro###BDonly

###更多的信息查看http://cole-trapnell-lab.github.io/monocle-release/docs/#installing-monocle

mmt$cell\_type\_val<-Idents(mmt)

mmt@meta.data[1:5,]

mmt<-FindVariableFeatures(mmt)

?FindVariableFeatures

matrix<-as.matrix(mmt@assays$RNA@counts)

dim(matrix)

matrix[1:500,1:10]

gene\_ann <- data.frame(

 gene\_short\_name = row.names(matrix),

 row.names = row.names(matrix)

)##基因注释

#oct[["cell\_group"]]<-Idents(oct)

mmt@meta.data[1:5,]

sample\_ann <- mmt@meta.data###细胞注释

fd <- new("AnnotatedDataFrame",data=gene\_ann)

pd<-new("AnnotatedDataFrame",data=sample\_ann)

#?newCellDataSet

sc\_cds\_2 <- newCellDataSet(matrix, phenoData = pd,featureData =fd,expressionFamily = negbinomial.size(),lowerDetectionLimit=0.01)

sc\_cds\_2 <- estimateSizeFactors(sc\_cds\_2)

sc\_cds\_2 <- estimateDispersions(sc\_cds\_2)

####QC#####

?detectGenes

sc\_cds\_2 <- detectGenes(sc\_cds\_2, min\_expr = 1)

expressed\_genes <- row.names(subset(fData(sc\_cds\_2), num\_cells\_expressed >= 5))

fData(sc\_cds\_2)[1:5,]

#####cluster analysis######

###以高变基因为ordering

#ordering\_genes<-mmt@assays$RNA@var.features

###以两组之间差异基因为ordering gene

diff\_test\_res <- differentialGeneTest(sc\_cds\_2[expressed\_genes,],

 fullModelFormulaStr = "~cell\_type\_val",cores = 4,

 verbose = T) #+num\_genes\_expressed+orig.ident

ordering\_genes <- row.names(subset(diff\_test\_res, qval < 1e-2)) #1e-200

#ordering\_genes<-markers %>% group\_by(cluster) %>% top\_n(n=200,wt=avg\_logFC)

sc\_cds2 <- setOrderingFilter(sc\_cds\_2, ordering\_genes)

plot\_ordering\_genes(sc\_cds2)

sc\_cds2<- reduceDimension(sc\_cds2, max\_components = 2, num\_dim=6,reduction\_method = "DDRTree")#,residualModelFormulaStr = "~orig.ident")

sc\_cds2 <- orderCells(sc\_cds2)

plot\_cell\_trajectory(sc\_cds2, color\_by = "cell\_type\_val",show\_branch\_points = T)

#sc\_cds2 <- orderCells(sc\_cds2,root\_state = 8)

#pData(sc\_cds\_2)[1:5,]

#p1<-plot\_cell\_trajectory(sc\_cds2, color\_by = "RNA\_snn\_res.0.8",show\_branch\_points = F)#+facet\_wrap(~RNA\_snn\_res.0.8)

plot\_cell\_trajectory(sc\_cds2, color\_by = "State",show\_branch\_points = T)#+facet\_wrap(~Site)

plot\_cell\_trajectory(sc\_cds2, color\_by = "Pseudotime",show\_branch\_points = T)

plot\_cell\_trajectory(sc\_cds2, markers = c("COL6A3","C3","IL6"),use\_color\_gradient = T,show\_branch\_points = F)

to\_be\_tested <- row.names(subset(fData(sc\_cds2), gene\_short\_name %in% c("COL6A3","C3","IL6")))

cds\_subset <- sc\_cds2[to\_be\_tested,]

plot\_genes\_in\_pseudotime(cds\_subset, color\_by = "cell\_type\_val")

####get the genes change along with the Pseudotime###

diff\_test\_res <- differentialGeneTest(sc\_cds2[expressed\_genes,],

 fullModelFormulaStr = "~sm.ns(Pseudotime)",cores = 5)

sig\_gene\_names <- row.names(subset(diff\_test\_res, qval < 0.05))

#dev.off()

dev.new()

pseudoplot<-plot\_pseudotime\_heatmap(sc\_cds2[sig\_gene\_names,],

 num\_clusters = 5,###进行调节分群

 cores = 5,

 show\_rownames = F,return\_heatmap = T)

####get the gene names in each cluster##

clusters <- cutree(pseudoplot$tree\_row, k = 4)

clustering <- data.frame(clusters)

clustering[,1] <- as.character(clustering[,1])

colnames(clustering) <- "Gene\_Clusters"

table(clustering)

library(msigdbr)

library(clusterProfiler)

msigdbr\_show\_species()

m\_df <- msigdbr(species = "Homo sapiens")

m\_t2g\_C5 <- msigdbr(species = "Homo sapiens", category = "C5",subcategory = "BP") %>%

 dplyr::select(gs\_name, gene\_symbol)

m\_t2g\_C7 <- msigdbr(species = "Homo sapiens", category = "C7") %>%

 dplyr::select(gs\_name, gene\_symbol)

m\_t2g\_C2 <- msigdbr(species = "Homo sapiens", category = "C2",subcategory = c("CP:KEGG")) %>%

 dplyr::select(gs\_name, gene\_symbol) ###signating pathways

###group1##

gene1<-rownames(clustering)[clustering[,1]==1]

em\_group1 <- enricher(gene1, TERM2GENE=m\_t2g\_C5)

em\_group\_pathway\_1 <- enricher(gene1, TERM2GENE=m\_t2g\_C2)

###simplify(em\_group1)

head(em\_group1)

head(em\_group\_pathway\_1)

barplot(em\_group1,showCategory = 20)

barplot(em\_group\_pathway\_1)

write.csv(em\_group1,file="./heatmap\_cluster1\_enrichment.csv")

gene2<-rownames(clustering)[clustering[,1]==2]

em\_group2<- enricher(gene2, TERM2GENE=m\_t2g\_C5)

head(em\_group2)

barplot(em\_group2,showCategory = 20)

write.csv(em\_group2,file="./heatmap\_cluster2\_enrichment.csv")

gene3<-rownames(clustering)[clustering[,1]==3]

em\_group3<- enricher(gene3, TERM2GENE=m\_t2g\_C5)

head(em\_group3)

barplot(em\_group3,showCategory = 20)

write.csv(em\_group3,file="./heatmap\_cluster3\_enrichment.csv")

gene4<-rownames(clustering)[clustering[,1]==4]

em\_group4<- enricher(gene4, TERM2GENE=m\_t2g\_C5)

head(em\_group4)

barplot(em\_group4,showCategory = 20)

write.csv(em\_group4,file="./heatmap\_cluster4\_enrichment.csv")

saveRDS(Myeloid,file="./CAFsMyeloid.rds")

rm(list=ls())

library(Seurat)

library(dplyr)

library(future)

library(future.apply)

library(stringr)

library(infercnv)

plan("multiprocess", workers = 6) ###set the compute core

options(future.globals.maxSize = 60000 \* 1024^2)

getwd()

rm(list=ls())

sce.mergeTEN<-readRDS("./sce.mergeTEN\_2.rds")

table(Idents(sce.mergeTEN))

epith<-sce.mergeTEN[,Idents(sce.mergeTEN)=="Epithelial"]

rm(sce.mergeTEN)

#####选择原来1/5的细胞数进行分析###

se<-seq(from=1,to=3491,by=1)

epith<-epith[,se]###缩小了的epithel数据，正式分析时不要取这个，直接用epith

####

matrix<-as.matrix(epith@assays$RNA@counts)

table(scenicOptions$features)

epith@meta.data[1:5,]

ann<-as.data.frame(ifelse(epith@meta.data[,"tissue\_type"]=="Tumor","maglignant\_Epithelial","Normal"),row.names=rownames(epith@meta.data))

ann<-as.matrix(ann)

colnames(ann)<-"cell\_type"

gene.order<-read.table(file="./gene\_position\_updated\_hg19.txt",header = F,sep = "\t")

rownames(gene.order)<-gene.order$V1

gene.order<-gene.order[,-1]

inferobj<-CreateInfercnvObject(raw\_counts\_matrix=matrix,

 annotations\_file=ann,

 gene\_order\_file=gene.order,

 ref\_group\_names=c("Normal"))

outdir\_1<-"./epi\_3"####输出的文件夹位置

#####会花比较长的时间，课上不跑，留在下课后跑

infercnv\_obj = infercnv::run(inferobj,

 cutoff=0.1, # cutoff=1 works well for Smart-seq2, and cutoff=0.1 works well for 10x Genomics

 out\_dir=outdir\_1,

 cluster\_by\_groups=TRUE,

 denoise=TRUE,plot\_steps = T,

 HMM=F)

####完整命令行

inferobj <- infercnv::run(inferobj,

 cutoff=0.1,

 out\_dir=outdir\_1,

 window\_length = 101,

 max\_centered\_threshold = 3,

 plot\_steps = F,denoise = TRUE,

 cluster\_by\_groups=T,

 no\_prelim\_plot = T,

 HMM=FALSE,

 sd\_amplifier = 1.3,

 analysis\_mode = "samples",

 num\_threads=3,

 png\_res = 200)

####check the malignant cells#####

####load the package###

library(cluster)

library(scales)

library(Seurat)

library(reshape2)

library(ggplot2)

library(cowplot)

library(fpc)

library(dplyr)

library(cluster)

#install.packages("fpc")

outdir<-"./epi\_3"##改为保存infercnv.22\_denoised.observations.txt文件的位置

infercnv\_output <- as.data.frame(t(read.table(paste0(outdir,

 "/infercnv.observations.txt"))))

library(stringr)

rownames(infercnv\_output)<-str\_replace\_all(rownames(infercnv\_output),pattern = "[.]",replacement = "-")##因为inferCNV将-转变为.了

infercnv\_output\_ref <- as.data.frame(t(read.table(paste0(outdir,

 "/infercnv.references.txt"))))

rownames(infercnv\_output\_ref)<-str\_replace\_all(rownames(infercnv\_output\_ref),pattern = "[.]",replacement = "-")

saveRDS(ann,file="ann.rds")

ann<-readRDS(file="ann.rds")

ann[1:5,]

metadata\_df<-data.frame(ann)

dim(metadata\_df)

metadata\_df$cell\_ids<-rownames(metadata\_df)

epithelial\_ids <- metadata\_df$cell\_ids[grep("magli", metadata\_df$cell\_type)]

epithelial\_heatmap <- infercnv\_output[rownames(infercnv\_output) %in% epithelial\_ids,]

tme\_ids <- metadata\_df$cell\_ids[grep("magli", metadata\_df$cell\_type,invert = T)]

tme\_heatmap<-infercnv\_output\_ref[rownames(infercnv\_output\_ref)%in% tme\_ids,]

print(paste0("Number of heatmap rows after non-epithelial thrown: ",

 nrow(epithelial\_heatmap)))

#

epithelial\_metadata <- metadata\_df[rownames(epithelial\_heatmap),]

tme\_metadata <- metadata\_df[rownames(tme\_heatmap),]

library(scales)

scaled\_df <- as.data.frame(rescale(as.matrix(epithelial\_heatmap), c(-1,1)))

CNA\_values <- apply(scaled\_df, 1, function(y) {

 return(mean(y^2))

})

tme\_scaled\_df <- as.data.frame(rescale(as.matrix(tme\_heatmap), c(-1,1)))

tme\_CNA\_values <- apply(tme\_scaled\_df, 1, function(y) {

 return(mean(y^2))

})

CNA\_value\_df <- data.frame(

 row.names = names(CNA\_values),

 CNA\_value = CNA\_values

)

tme\_CNA\_value\_df <- data.frame(

 row.names = names(tme\_CNA\_values),

 CNA\_value = tme\_CNA\_values

)

epithelial\_metadata <- cbind(epithelial\_metadata, CNA\_value\_df)

print(paste0(

 "Are epithelial\_metadata rownames still in the same order as epithelial\_heatmap?? ",

 identical(rownames(epithelial\_heatmap), rownames(epithelial\_metadata))

))

tme\_metadata <- cbind(tme\_metadata, tme\_CNA\_value\_df)

total<-rbind(epithelial\_heatmap,tme\_heatmap)

total\_df<-rbind(CNA\_value\_df,tme\_CNA\_value\_df)

#table(rownames(total)==rownames(total\_df))

###对肿瘤全部细胞CNV值进行排序 并取前面5%作为参照

CNA\_order <- order(CNA\_value\_df$CNA\_value, decreasing=T)

ordered\_CNA\_values <- data.frame(

 row.names = rownames(CNA\_value\_df)[CNA\_order],

 CNA\_value = CNA\_value\_df[CNA\_order,]

)

top\_cancer <- head(ordered\_CNA\_values, nrow(ordered\_CNA\_values)\*0.05)###top5%的细胞

top\_cancer

dim(top\_cancer)

top\_cancer\_CNV\_average <- apply(epithelial\_heatmap[rownames(top\_cancer),], 2, mean)

top\_cancer\_CNV\_average

total\_correlations <- apply(total, 1, function(x) {

 cor <- cor.test(as.numeric(x), top\_cancer\_CNV\_average, method = "pearson")

 cor\_result <- data.frame(cor$estimate, cor$p.value)

 return(cor\_result)

})

total\_correlation\_df <- do.call("rbind", total\_correlations)

total\_metadata<-rbind(epithelial\_metadata,tme\_metadata)

table(rownames(total\_metadata)==rownames(total\_correlation\_df))

total\_metadata <- cbind(total\_metadata, total\_correlation\_df)

print(paste0(

 "Are epithelial\_metadata rownames still in the same order as epithelial\_heatmap?? ",

 identical(rownames(epithelial\_heatmap), rownames(epithelial\_metadata))

))

total\_metadata<-data.frame(total\_metadata)

total\_metadata$normal\_cell\_call <- "Normal"

#####玉质需要自己设置

total\_metadata$normal\_cell\_call[

 total\_metadata$CNA\_value > 0.2|total\_metadata$cor.estimate > 0.2

] <- "Cancer"

table(total\_metadata$normal\_cell\_call,total\_metadata$cell\_type)###

#epithelial\_metadata<-epithelial\_metadata[,c(1:8)]

# create quad plot:

library(ggplot2)

p <- ggplot(total\_metadata,

 aes(x=CNA\_value, y=cor.estimate, color=as.factor(normal\_cell\_call)))+

 geom\_point()+

 scale\_color\_manual(values=c("black", "#74add1"),

 labels=c("Cancer", "Normal"))+

 xlab("Infercnv level")+

 ylab("Corr. with top 5% cancer (p<0.05)")+

 theme(legend.title = element\_blank())+

 geom\_vline(xintercept = 0.2)+

 geom\_hline(yintercept = 0.2)

p

####定义为unassign####

total\_metadata$normal\_cell\_call\_2<-total\_metadata$normal\_cell\_call

total\_metadata$normal\_cell\_call\_2[

 total\_metadata$cell\_type=="maglignant\_Epithelial"&total\_metadata$normal\_cell\_call=="Normal"

] <- "Unassigned"

table(total\_metadata$normal\_cell\_call\_2)

p <- ggplot(total\_metadata,

 aes(x=CNA\_value, y=cor.estimate, color=as.factor(normal\_cell\_call\_2)))+

 geom\_point()+

 scale\_color\_manual(values=c("black", "#74add1", "#b2182b"),

 labels=c("Cancer", "Normal", "Unassigned"))+

 xlab("Infercnv level")+

 ylab("Corr. with top 5% cancer (p<0.05)")+

 theme(legend.title = element\_blank())+

 geom\_vline(xintercept = 0.04)+

 geom\_hline(yintercept = 0.2)

p

write.csv(total\_metadata,file="./Epithelial\_tumor\_or\_normal.csv")

save.image(file="./infCNV.RData")

rm(list=ls())

#install.packages("Seurat")

library(Seurat)

library(dplyr)

library(future)

library(future.apply)

plan("multiprocess", workers = 6) ###set the compute core

options(future.globals.maxSize = 60000 \* 1024^2)

getwd()

library(stringr)

#memory.limit()

#BiocManager::install("igraph",force = T)

#setwd("~/Documents\_PC/scRNA-seq/Data/")

####prepare the data for cce.mergeTellchat analysis####

sce.mergeTEN<-readRDS("./sce.mergeTEN\_3.rds")

 #write.table(as.matrix(sce.mergeTEN@assays$RNA@data), 'cellphonedb\_count.txt', sep='\t', quote=F)

 #meta\_data <- cbind(rownames(sce.mergeTEN@meta.data), sce.mergeTEN@meta.data[,'Seurat\_harmony', drop=F])

#meta\_data <- as.matrix(meta\_data)

 #meta\_dat[is.na(meta\_data)] = "Unkown" # 细胞类型中不能有NA

 #write.table(meta\_data, 'cellphonedb\_meta.txt', sep='\t', quote=F, row.names=F)

table(Idents(sce.mergeTEN))

sce.mergeTEN<-sce.mergeTEN[,Idents(sce.mergeTEN)!="Doublets"]

sce.mergeTEN@meta.data[1:5,]

table(sce.mergeTEN$tissue\_type)

levels(Idents(sce.mergeTEN))

normaldata<-sce.mergeTEN[,sce.mergeTEN$tissue\_type=="Normal"]

tumordata<-sce.mergeTEN[,sce.mergeTEN$tissue\_type=="Tumor"]

##减少计算，只分析前面2000个细胞

normaldata<-normaldata[,1:2000]

table(Idents(normaldata))

tumordata<-tumordata[,1:2000]

table(Idents(tumordata))

###perform the cellchat analysis##

#devtools::install\_github("sqjin/CellChat")

library(CellChat)

normal.input <- GetAssayData(normaldata, assay = "RNA", slot = "data") # normalized data matrix

labels <- factor(normaldata$Seurat\_harmony,levels=levels(Idents(sce.mergeTEN)))

labels

meta <- data.frame(group = labels, row.names = rownames(normaldata@meta.data)) # create a dataframe of the cell labels

meta$group<-str\_replace\_all(meta$group,pattern = "/",replacement = "\_")

meta$group <- factor(meta$group,levels=c("T cell","Epithelial","Smooth muscle cell","Fibroblasts","Endothelial cell","Macrophage","B cell","Lymphocyte","Plasma cell"))

cellchat\_normal <- createCellChat(object = normal.input, meta = meta, group.by = "group")

saveRDS(cellchat\_normal,file="./Cellchat/normalcellchat\_1.rds")

tumor.input <- GetAssayData(tumordata, assay = "RNA", slot = "data") # normalized data matrix

labels <- factor(tumordata$Seurat\_harmony,levels=levels(Idents(sce.mergeTEN)))

meta <- data.frame(group = labels, row.names = rownames(tumordata@meta.data)) # create a dataframe of the cell labels

meta$group<-str\_replace\_all(meta$group,pattern = "/",replacement = "\_")

meta$group <- factor(meta$group,levels=c("T cell","Epithelial","Smooth muscle cell","Fibroblasts","Endothelial cell","Macrophage","B cell","Lymphocyte","Plasma cell"))

cellchat\_tumor <- createCellChat(object = tumor.input, meta = meta, group.by = "group")

saveRDS(cellchat\_tumor,file="./Cellchat/tumorcellchat\_1.rds")

####prepare the deconvolution analysis gene signature####

rm(list=ls())

setwd("~/Documents\_PC/scRNA-seq/Data")

normal\_cellchat<-readRDS(file="./Cellchat/normalcellchat\_1.rds")

tumor\_cellchat<-readRDS(file="./Cellchat/normalcellchat\_1.rds")

normal\_cellchat@DB <- CellChatDB.human

gc()##释放内存

normal\_cellchat <- subsetData(normal\_cellchat) # subset the expression data of signaling genes for saving computation cost

#future::plan("multiprocess", workers = 2) # do parallel

#> Warning: [ONE-TIME WARNING] Forked processing ('multicore') is disabled

#> in future (>= 1.13.0) when running R from RStudio, because it is

#> considered unstable. Because of this, plan("multicore") will fall

#> back to plan("sequential"), and plan("multiprocess") will fall back to

#> plan("multisession") - not plan("multicore") as in the past. For more details,

#> how to control forked processing or not, and how to silence this warning in

#> future R sessions, see ?future::supportsMulticore

normal\_cellchat <- identifyOverExpressedGenes(normal\_cellchat)

normal\_cellchat <- identifyOverExpressedInteractions(normal\_cellchat)

normal\_cellchat <- projectData(normal\_cellchat, PPI.human)

normal\_cellchat <- computeCommunProb(normal\_cellchat, raw.use = TRUE)

# Filter out the cell-cell communication if there are only few number of cells in certain cell groups

normal\_cellchat <- filterCommunication(normal\_cellchat, min.cells = 1)

normal\_cellchat <- computeCommunProbPathway(normal\_cellchat)

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

saveRDS(normal\_cellchat,file="./Cellchat/normalcellchat\_1.rds")

normal\_cellchat<-readRDS("./Cellchat/normalcellchat\_1.rds")

normal\_cellchat <- aggregateNet(normal\_cellchat)

groupSize <- as.numeric(table(normal\_cellchat@idents))

groupSize

table(normal\_cellchat@idents)

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

netVisual\_circle(normal\_cellchat@net$count,arrow.size = 0.01, vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Number of interactions")

netVisual\_circle(normal\_cellchat@net$weight, arrow.size = 0.01,vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Interaction weights/strength")

mat <- normal\_cellchat@net$count

#或者看weight

mat <- normal\_cellchat@net$weight

dev.off()

####下面针对其中一个细胞亚型分析其对其他细胞的interaction的Ligand-receptor数量

mat2 <- matrix(0, nrow = nrow(mat), ncol = ncol(mat), dimnames = dimnames(mat))

mat2[1, ] <- mat[1, ]

netVisual\_circle(mat2, vertex.weight = groupSize,arrow.size = 0.2, weight.scale = T, edge.weight.max = max(mat), title.name = rownames(mat)[1])

###查看具体的信号通路

df.net <- subsetCommunication(normal\_cellchat)

#df.net\_tumor <- subsetCommunication(tumor\_cellchat)

levels(df.net$source)

pathways.show <- c("PDGF")

#table(df.net$pathway\_name)

# Hierarchy plot

# Here we define `vertex.receive` so that the left portion of the hierarchy plot shows signaling to fibroblast and the right portion shows signaling to immune cells

#dev.new()

seq(1,4)

vertex.receiver = seq(1,4) # a numeric vector.

netVisual\_aggregate(normal\_cellchat, signaling = pathways.show, vertex.receiver = vertex.receiver,layout = "hierarchy")

netVisual\_aggregate(normal\_cellchat, signaling = pathways.show, vertex.receiver = vertex.receiver,layout = "chord")

#?netVisual\_aggregate

#par(mfrow=c(1,1))

#netVisual\_aggregate(normal\_cellchat, signaling = pathways.show,vertex.receiver = vertex.receiver, layout = "circle")

netVisual\_aggregate(normal\_cellchat, signaling = pathways.show, layout = "circle")

netAnalysis\_contribution(normal\_cellchat, signaling = pathways.show)

####get specific cell

levels(normal\_cellchat@idents)

netVisual\_bubble(normal\_cellchat, sources.use = c(1), targets.use = c(2:10), remove.isolate = FALSE)###epithelial

netVisual\_bubble(normal\_cellchat, sources.use = c(2), targets.use = c(8,9), remove.isolate = FALSE)##TNK

netVisual\_bubble(normal\_cellchat, sources.use = c(4), targets.use = c(1:3,5:9), remove.isolate = FALSE)##myeloid cell

#saveRDS(normal\_cellchat,file="./Cellchat/normalcellchat\_analysis.rds")

###We have run these in the workstation###

tumor\_cellchat<-readRDS(file="./Cellchat/tumorcellchat\_1.rds")

tumor\_cellchat@DB <- CellChatDB.human

#gc()

tumor\_cellchat <- subsetData(tumor\_cellchat) # subset the expression data of signaling genes for saving computation cost

future::plan("multiprocess", workers = 4) # do parallel

tumor\_cellchat <- identifyOverExpressedGenes(tumor\_cellchat)

tumor\_cellchat <- identifyOverExpressedInteractions(tumor\_cellchat)

tumor\_cellchat <- projectData(tumor\_cellchat, PPI.human)

tumor\_cellchat <- computeCommunProb(tumor\_cellchat, raw.use = TRUE)

# Filter out the cell-cell communication if there are only few number of cells in certain cell groups

tumor\_cellchat <- filterCommunication(tumor\_cellchat, min.cells = 1)

tumor\_cellchat <- computeCommunProbPathway(tumor\_cellchat)

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

saveRDS(tumor\_cellchat,file="./Cellchat/tumorcellchat\_1.rds")

tumor\_cellchat<-readRDS("./Cellchat/tumorcellchat\_1.rds")

tumor\_cellchat <- aggregateNet(tumor\_cellchat)

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

groupSize <- as.numeric(table(tumor\_cellchat@idents))

groupSize

table(tumor\_cellchat@idents)

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

netVisual\_circle(tumor\_cellchat@net$count,arrow.size = 0.01, vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Number of interactions")

netVisual\_circle(tumor\_cellchat@net$weight, arrow.size = 0.01,vertex.weight = groupSize, weight.scale = T, label.edge= F, title.name = "Interaction weights/strength")

mat <- tumor\_cellchat@net$count

##mat <- tumor\_cellchat@net$weight

#dev.off()

####下面针对其中一个细胞亚型分析其对其他细胞的interaction的Ligand-receptor数量

mat2 <- matrix(0, nrow = nrow(mat), ncol = ncol(mat), dimnames = dimnames(mat))

mat2[1, ] <- mat[1, ]

netVisual\_circle(mat2, vertex.weight = groupSize,arrow.size = 0.2, weight.scale = T, edge.weight.max = max(mat), title.name = rownames(mat)[1])

###查看具体的信号通路

df.net <- subsetCommunication(tumor\_cellchat)

#df.net\_tumor <- subsetCommunication(tumor\_cellchat)

levels(df.net$source)

pathways.show <- c("PDGF")

#table(df.net$pathway\_name)

# Hierarchy plot

# Here we define `vertex.receive` so that the left portion of the hierarchy plot shows signaling to fibroblast and the right portion shows signaling to immune cells

#dev.new()

vertex.receiver = seq(1,4) # a numeric vector.

netVisual\_aggregate(tumor\_cellchat, signaling = pathways.show, vertex.receiver = vertex.receiver,layout = "hierarchy")

?netVisual\_aggregate

par(mfrow=c(1,1))

netVisual\_aggregate(tumor\_cellchat, signaling = pathways.show, layout = "circle")

netAnalysis\_contribution(tumor\_cellchat, signaling = pathways.show)

####get specific cell

netVisual\_bubble(normal\_cellchat, sources.use = c(1), targets.use = c(2:10), remove.isolate = FALSE)###epithelial

netVisual\_bubble(tumor\_cellchat, sources.use = c(1), targets.use = c(2:10), remove.isolate = FALSE)###epithelial

netVisual\_bubble(normal\_cellchat, sources.use = c(2), targets.use = c(1,3:10), remove.isolate = FALSE)##TNK

netVisual\_bubble(tumor\_cellchat, sources.use = c(2), targets.use = c(1,3:10), remove.isolate = FALSE)##TNK

netVisual\_bubble(normal\_cellchat, sources.use = c(4), targets.use = c(1:3,5:9), remove.isolate = FALSE)##myeloid cell

netVisual\_bubble(tumor\_cellchat, sources.use = c(5), targets.use = c(1:4,6:10), remove.isolate = FALSE)##myeloid cell

#netAnalysis\_signalingRole\_network(normal\_cellchat, signaling = c("PDGF"), width = 8, height = 2.5, font.size = 10)#,"VEGF"

# the slot 'netP' means the inferred intercellular communication network of signaling pathways

# Visualize the computed centrality scores using heatmap, allowing ready identification of major signaling roles of cell groups

netAnalysis\_signalingRole\_network(tumor\_cellchat, signaling = c("PDGF"), width = 8, height = 2.5, font.size = 10)#,"VEGF"

netAnalysis\_signalingRole\_network(tumor\_cellchat, signaling = c("IFN-II"), width = 8, height = 2.5, font.size = 10)

pdf(file="tumor1.pdf",width = 12, height = 10)

ht1 <- netAnalysis\_signalingRole\_heatmap(tumor\_cellchat, pattern = "outgoing",height = 18)

ht2 <- netAnalysis\_signalingRole\_heatmap(tumor\_cellchat, pattern = "incoming",height = 18)

ht1 + ht2

dev.off()

#normal\_cellchat <- netAnalysis\_computeCentrality(normal\_cellchat, slot.name = "netP") # the slot 'netP' means the inferred intercellular communication network of signaling pathways

# Visualize the computed centrality scores using heatmap, allowing ready identification of major signaling roles of cell groups

netAnalysis\_signalingRole\_network(normal\_cellchat, signaling = c("IGF"), width = 8, height = 2.5, font.size = 10)

netAnalysis\_signalingRole\_network(normal\_cellchat, signaling = c("IFN-II"), width = 8, height = 2.5, font.size = 10)

pdf(file="normal1.pdf",width = 12, height = 10)

ht1 <- netAnalysis\_signalingRole\_heatmap(normal\_cellchat, pattern = "outgoing",height = 18)

ht2 <- netAnalysis\_signalingRole\_heatmap(normal\_cellchat, pattern = "incoming",height = 18)

ht1 + ht2

dev.off()

#saveRDS(normal\_cellchat,file="./Cellchat/normalcellchat\_analysis.rds")

#saveRDS(tumor\_cellchat,file="./Cellchat/tumorcellchat\_analysis.rds")

###we next compare the cell signaling pathways between the normal and tumor tissues##

data.dir <- './comparison'

dir.create(data.dir)

setwd(data.dir)

object.list <- list(normal = normal\_cellchat, tumor = tumor\_cellchat)

cellchat <- mergeCellChat(object.list, add.names = names(object.list))

#> Merge the following slots: 'data.signaling','net', 'netP','meta', 'idents', 'var.features' , 'DB', and 'LR'.

cellchat

gg1 <- compareInteractions(cellchat, show.legend = F, group = c(1,2))

gg2 <- compareInteractions(cellchat, show.legend = F, group = c(1,2), measure = "weight")

gg1 + gg2

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

netVisual\_diffInteraction(cellchat, weight.scale = T)###counts 差异

netVisual\_diffInteraction(cellchat, weight.scale = T, measure = "weight")###weight的差异

gg1 <- netVisual\_heatmap(cellchat)

#> Do heatmap based on a merged object

gg2 <- netVisual\_heatmap(cellchat, measure = "weight")

#> Do heatmap based on a merged object

gg1 + gg2

gg1 <- rankNet(cellchat, mode = "comparison", stacked = T, do.stat = TRUE)

gg2 <- rankNet(cellchat, mode = "comparison", stacked = F, do.stat = TRUE)

gg1 + gg2

library(ComplexHeatmap)

#> Loading required package: grid

#> ========================================

#> ComplexHeatmap version 2.7.1.1010

#> Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/

#> Github page: https://github.com/jokergoo/ComplexHeatmap

#> Documentation: http://jokergoo.github.io/ComplexHeatmap-reference

#>

#> If you use it in published research, please cite:

#> Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional

#> genomic data. Bioinformatics 2016.

#>

#> This message can be suppressed by:

#> suppressPackageStartupMessages(library(ComplexHeatmap))

#> ========================================

i = 1

# combining all the identified signaling pathways from different datasets

pathway.union <- union(object.list[[i]]@netP$pathways, object.list[[i+1]]@netP$pathways)

ht1 = netAnalysis\_signalingRole\_heatmap(object.list[[i]], pattern = "outgoing", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6)

ht2 = netAnalysis\_signalingRole\_heatmap(object.list[[i+1]], pattern = "outgoing", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6)

draw(ht1 + ht2, ht\_gap = unit(0.5, "cm"),height=unit(12, "cm"),width=unit(16, "cm"))

ht1 = netAnalysis\_signalingRole\_heatmap(object.list[[i]], pattern = "incoming", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6, color.heatmap = "GnBu")

ht2 = netAnalysis\_signalingRole\_heatmap(object.list[[i+1]], pattern = "incoming", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6, color.heatmap = "GnBu")

draw(ht1 + ht2, ht\_gap = unit(0.5, "cm"),height=unit(12, "cm"),width=unit(16, "cm"))

ht1 = netAnalysis\_signalingRole\_heatmap(object.list[[i]], pattern = "all", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6, color.heatmap = "OrRd")

ht2 = netAnalysis\_signalingRole\_heatmap(object.list[[i+1]], pattern = "all", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6, color.heatmap = "OrRd")

draw(ht1 + ht2, ht\_gap = unit(0.5, "cm"),height=unit(12, "cm"))

cellchat@idents

##Part III: Identify the upgulated and down-regulated signaling ligand-receptor pairs

netVisual\_bubble(cellchat, sources.use = c(1), targets.use = c(4,7), comparison = c(1, 2), angle.x = 45)

netVisual\_bubble(cellchat, sources.use = c(1), targets.use = c(2,3), comparison = c(1, 2), angle.x = 45)

###指定信号通路

netVisual\_bubble(cellchat, sources.use = c(1), targets.use = c(4,7), signaling = c("MIF"), comparison = c(1, 2), angle.x = 90)

colnames(mat2)

##compare specific gene clusters### vlnplot

cellchat@meta$datasets = factor(cellchat@meta$datasets, levels = c("normal", "tumor")) # set factor level

plotGeneExpression(cellchat, signaling = "CXCL", split.by = "datasets", colors.ggplot = T)

plotGeneExpression(cellchat, signaling = "VEGF", split.by = "datasets", colors.ggplot = T)

plotGeneExpression(cellchat, signaling = "IGF", split.by = "datasets", colors.ggplot = T)

save.image(file="./Cell\_chat.RData")