#!/Usr/bin/env Rscript

# load library

library(DESeq2)

#Step 0. Set path--------------------------------------------

BasePATH=setwd("~/RNAseqProject")

print(BasePATH)

tmpGSE= list.files(path = BasePATH, pattern = "^GSE")

GseDir= tmpGSE[ !grepl(".R", tmpGSE)]

for (j in GseDir)

{

print(j)

SETPATH=paste0(BasePATH,"/",j)

print(SETPATH)

setwd(SETPATH)

getwd()

#Getting GSEaccession list fronm the directory

GseNum=gsub("GSE","",j)

designName=paste0("design\_",GseNum,".csv")

print(designName)

#Create design matrix-----------------------------------------------

sampleInfo <-read.table(designName, sep=',',as.is=T, header=T)

print(sampleInfo)

head(sampleInfo)

rownames(sampleInfo)<-sampleInfo[,1]

#Condition\_GSE<-paste0(j,sampleInfo[,2])

#print(Condition\_GSE)

#GSESampleInfo<-cbind(sampleInfo[,1],Condition\_GSE )

#print(GSESampleInfo)

#Load Data and generate readcountMAtrix---------------------------------

GSEName=paste0("",j,".csv")

readcount\_table<-(read.table(paste(sampleInfo[1,1]),sep='\t',as.is=T, header=T))

res <- readcount\_table[1];

colnames(res)[1] <- "GENE ID"

Readcountlist<-read.csv(GSEName,header=FALSE)

for (i in 1:nrow(Readcountlist))

{

print(paste(Readcountlist[i,]))

readcount\_table<-(read.table(paste(Readcountlist[i,]),sep='\t',as.is=T, header=T))

res<- cbind(res, readcount\_table[7])

colnames(res)[i+1] <- paste(Readcountlist[i,])

}

ReadcountMatrixName<-paste0("",j,"countmatrix.csv")

print(ReadcountMatrixName)

write.csv(as.data.frame(res),file="results/ReadcountMatrixName.csv")

InputDF<-data.frame(res)

print(InputDF)

InputDF<-data.frame(res)

InputDF2<-InputDF[,-1]

rownames(InputDF2)<-InputDF[,1]

print(InputDF2)

#Differential expression analysis

dds = DESeqDataSetFromMatrix(countData = InputDF2, colData=sampleInfo, design = ~Treatment)

dds<-DESeq(dds)

res2<-results(dds)

summary(res2)

#Generating Normalized readcount matrix-----------------------

countData <- data.frame(InputDF2)

dim(countData)

dds.EstSizFac <- estimateSizeFactors(dds)

norm.count=counts(dds.EstSizFac, normalized=TRUE)

head(norm.count)

head(countData)

norm.count.round = round(norm.count, 3)

head(norm.count.round)

#------

#ConditinIndex<-paste0("",j,(sampleInfo[,2]))

#print(ConditionIndex)

NormcountMatrixName<-paste0(j,"Norm.count.csv")

write.csv(as.data.frame(norm.count.round),file="results/NormcountMatrixName.csv", row.names=T)

#Getting FPKM and load library edgeR--------------------

library(edgeR)

head(norm.count)

GeneidLengthFile="SRR.ID.Length"

AnnoData <- read.table(GeneidLengthFile, sep='\t',as.is=T, header=T)

head(AnnoData)

head(AnnoData)

norm.count.DGEList <- DGEList(counts=norm.count, genes=AnnoData[,c("Geneid","Length")])

print(norm.count.DGEList)

norm.rpkm <- rpkm(norm.count.DGEList, norm.count.DGEList$genes$Length)

head(norm.rpkm)

norm.rpkm.round = round(norm.rpkm, 3)

head(norm.rpkm.round)

NormFPKMName<-paste0("",j,"Norm.FPKM.csv")

write.csv(as.data.frame(norm.rpkm.round),file="results/NormFPKMName.csv", row.names=T)

# Getting average FPKM value for each GSE------------------------

FPKM.j<-read.csv("results/NormFPKMName.csv",header=T)

WholeAveFPKM=data.frame(GeneID=res[,1])

head(WholeAveFPKM)

for (i in unique(sampleInfo[,2]))

{

print(i)

SRR<-sampleInfo[(which(sampleInfo[,2]==i)),]

print(SRR)

FPKM.condition<-FPKM.j[,colnames(FPKM.j) %in% SRR[,1]]

head(FPKM.condition)

MeanFPKMCondition<-as.matrix(rowMeans(FPKM.condition))

head(MeanFPKMCondition)

colnames(MeanFPKMCondition)<-i

head(MeanFPKMCondition)

WholeAveFPKM=cbind(WholeAveFPKM,MeanFPKMCondition)

}

head(WholeAveFPKM)

MeanFPKMName<-paste0("",j,"Mean.FPKM.csv")

write.csv(as.data.frame(WholeAveFPKM),file="results/MeanFPKMName.csv", row.names=T)