query <- GDCquery(project = "TCGA-LGG",

data.category = "Transcriptome Profiling",

data.type = "miRNA Expression Quantification")

query <- GDCquery(project = "TCGA-LGG",

data.category = "Transcriptome Profiling",

data.type = "Gene Expression Quantification",

workflow.type = "HTSeq - Counts")

colnames <- c()

i <- 2

while(i < length(colnames(mirna\_data))){

colnames <- c(colnames,substr(colnames(mirna\_data)[i],12,26))

i <- i + 3

}

cols <- (length(colnames(mirna\_data)) - 1)/3

rows <- length(rownames(mirna\_data))

dd <- matrix(rep(NA,cols\*rows),nrow = rows,

dimnames = list(mirna\_data[,1],colnames))

i <- 2

j <- 1

while(i < length(colnames(mirna\_data))){

dd[,j] <- mirna\_data[,i]

j <- j + 1

i <- i + 3

}

metaMatrix.RNA <- gdcParseMetadata(project.id = 'TCGA-LGG',

data.type = 'RNAseq',

write.meta = FALSE)

ii <- which(substr(rownames(metaMatrix.RNA),1,12)

%!in% substr(colnames(mirna\_data),1,12))

ii <- which(substr(colnames(mirna\_data),1,12)

%!in% substr(rownames(metaMatrix.RNA),1,12))

metaMatrix.RNAA <- gdcParseMetadata(project.id = 'TCGA-CHOL',

data.type = 'RNAseq',

write.meta = FALSE)

DEGAll <- gdcDEAnalysis(counts = rnaCounts,

group = metaMatrix.RNAA$sample\_type,

comparison = 'PrimaryTumor-SolidTissueNormal',

method = 'limma')

DEGAll0 <- gdcDEAnalysis(counts = matrixRNA,

group = metaMatrix$sample\_type,

comparison = 'Oligodendroglioma-Astrocytoma',

method = 'DESeq2')

DEGAll <- TCGAanalyze\_DEA(mat1 = dataFilt[,oligo\_type],

mat2 = dataFilt[,astro\_type],

Cond1type = "Oligodendroglioma",

Cond2type = "Astrocytoma",

fdr.cut = 0.01,

logFC.cut = 1,

method = "exactTest")

ceOutput <- gdcCEAnalysis(lnc = rownames(deLNC),

pc = rownames(dePC),

lnc.targets = 'starBase',

pc.targets = 'starBase',

rna.expr = rnaExpr,

mir.expr = mirExpr)

project <- 'TCGA-LGG'

rnadir <- paste(project, 'RNAseq', sep='/')

mirdir <- paste(project, 'miRNAs', sep='/')

####### Download RNAseq data #######

gdcRNADownload(project.id = 'TCGA-LGG',

data.type = 'RNAseq',

write.manifest = FALSE,

method = 'gdc-client',

directory = rnadir)

####### Download mature miRNA data #######

gdcRNADownload(project.id = 'TCGA-LGG',

data.type = 'miRNAs',

write.manifest = FALSE,

method = 'gdc-client',

directory = mirdir)

clinicaldir <- paste(project, 'Clinical', sep='/')

gdcClinicalDownload(project.id = 'TCGA-LGG',

write.manifest = FALSE,

method = 'gdc-client',

directory = clinicaldir)

metaMatrix.RNA <- gdcParseMetadata(project.id = 'TCGA-LGG',

data.type = 'RNAseq',

write.meta = FALSE)

####### Filter duplicated samples in RNAseq metadata #######

metaMatrix.RNA <- gdcFilterDuplicate(metaMatrix.RNA)

####### Filter non-Primary Tumor and non-Solid Tissue Normal samples in RNAseq metadata #######

metaMatrix.RNA <- gdcFilterSampleType(metaMatrix.RNA)

metaMatrix.MIR <- gdcParseMetadata(project.id = 'TCGA-LGG',

data.type = 'miRNAs',

write.meta = FALSE)

####### Filter duplicated samples in miRNAs metadata #######

metaMatrix.MIR <- gdcFilterDuplicate(metaMatrix.MIR)

####### Filter non-Primary Tumor and non-Solid Tissue Normal samples in miRNAs metadata #######

metaMatrix.MIR <- gdcFilterSampleType(metaMatrix.MIR)

rSeq\_dir <- "/Users/mac/Desktop/DATA\_LGG/GDCdata/TCGA-LGG/harmonized/Transcriptome\_Profiling/Gene\_Expression\_Quantification/"

mir\_dir <- "/Users/mac/Desktop/DATA\_LGG/GDCdata/TCGA-LGG/harmonized/Transcriptome\_Profiling/miRNA\_Expression\_Quantification/"

####### Merge RNAseq data #######

rSeqCounts <- gdcRNAMerge(metadata = metaMatrix.RNA,

path = rSeq\_dir, # the folder in which the data stored

organized = FALSE, # if the data are in separate folders

data.type = 'RNAseq')

####### Merge miRNAs data #######

mSeqCounts <- gdcRNAMerge(metadata = metaMatrix.MIR,

path = mir\_dir, # the folder in which the data stored

organized = FALSE, # if the data are in separate folders

data.type = 'miRNAs')

clin\_all <- read.csv("/Users/mac/Desktop/DATA\_LGG/clin\_all.csv",sep = "\t")

aaa <- which(clin\_all$WHO2016 == "N.O.S")

clin\_all <- clin\_all[-aaa,]

clin\_all <- clin\_all[,match(colnames(rSeqCounts),substr(clin\_all$Tumor\_Sample\_Barcode,1,15))]

rnaExpr <- gdcVoomNormalization(counts = rSeqCounts, filter = FALSE)

mirExpr <- gdcVoomNormalization(counts = mSeqCounts, filter = FALSE)

DEGAll00 <- gdcDEAnalysis0(counts = rSeqCounts,

group = metaMatrix.RNA$WHO2016,

comparison = 'Astrocytoma-Oligodendroglioma',

method = 'limma')

deALL <- gdcDEReport(deg = DEGAll, gene.type = 'all')

### DE long-noncoding

deLNC <- gdcDEReport(deg = DEGAll, gene.type = 'long\_non\_coding')

### DE protein coding genes

dePC <- gdcDEReport(deg = DEGAll, gene.type = 'protein\_coding')

ceOutput <- gdcCEAnalysis0(lnc = rownames(deLNC),

pc = rownames(dePC),

lnc.targets = lncsTarg,

pc.targets = pcTarg,

rna.expr = rnaExpr,

mir.expr = mirExpr00)

ceOutput2 <- ceOutput[ceOutput$hyperPValue<0.01 &

ceOutput$corPValue<0.01 & ceOutput$regSim != 0,]

edges <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'edges')

nodes <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'nodes')

write.table(edges, file='edges.txt', sep='\t', quote=F)

write.table(nodes, file='nodes.txt', sep='\t', quote=F)

ceLNC <- rownames(deLNC)[rownames(deLNC) %in% names(lncTarget)]

survOutput <- gdcSurvivalAnalysis(gene = rownames(deALL),

method = 'KM',

rna.expr = rnaExpr00,

metadata = metaMatrix.RNA0,

sep = 'median')

####################################

rnaCounts\_sub <- rnaCounts[match(colnames(rnaCounts),substr(clin\_data$sample,1,15)),]

rnaCounts\_sub0 <- rnaCounts\_sub[,which(colnames(rnaCounts\_sub) %in% substr(ccc,1,15))]

metaMatrix.RNA0 <- metaMatrix.RNA[which(colnames(rnaCounts\_sub) %in% substr(ccc,1,15)),]

clin\_data0 <- clin\_data[which(substr(clin\_data$sample,1,15) %in%colnames(rnaCounts\_sub)),]

metaMatrix.RNA0$WHO2016 <- clin\_data0$WHO2016

> DEGAll <- gdcDEAnalysis(counts = rnaCounts\_sub0,

+ group = metaMatrix.RNA0$WHO2016,

+ comparison = 'Astrocytoma-Oligodendroglioma',

+ method = 'limma')

Erreur : NA counts not allowed

aaa <- which(is.na(rownames(rnn))

+ )

> aaa

[1] 52 82 106 172 447 449

> rnn <- rnn[-aaa,]

> aaa <- which(is.na(rownames(rnn)))

> aaa

integer(0)

> DEGAll <- gdcDEAnalysis(counts = rnn,

+ group = metaMatrix.RNA0$WHO2016,

+ comparison = 'Astrocytoma-Oligodendroglioma',

+ method = 'limma')

symbol group logFC AveExpr t

ENSG00000026025 VIM protein\_coding 2.022192 14.049314 17.348102

ENSG00000006071 ABCC8 protein\_coding -2.084977 9.644588 -15.123639

ENSG00000029534 ANK1 protein\_coding -1.528081 8.720947 -14.753284

ENSG00000002587 HS3ST1 protein\_coding 2.235958 8.517953 14.412425

ENSG00000023892 DEF6 protein\_coding 1.318421 8.054333 13.666726

ENSG00000011600 TYROBP protein\_coding 1.277700 10.200646 12.683493

ENSG00000008300 CELSR3 protein\_coding -1.325491 10.660100 -12.513434

ENSG00000012779 ALOX5 protein\_coding 1.435513 8.256042 12.449704

ENSG00000009790 TRAF3IP3 protein\_coding 1.129309 6.116057 11.788639

ENSG00000011677 GABRA3 protein\_coding -1.559554 9.631813 -11.465233

ENSG00000019582 CD74 protein\_coding 1.385767 14.068042 11.128556

ENSG00000007129 CEACAM21 protein\_coding 1.106478 5.520098 11.080556

ENSG00000005844 ITGAL protein\_coding 1.347374 7.671231 11.071282

ENSG00000002586 CD99 protein\_coding 1.228890 12.959680 10.561370

ENSG00000003436 TFPI protein\_coding 1.809799 8.018655 10.534950

ENSG00000006747 SCIN protein\_coding 2.010267 8.940576 10.243346

ENSG00000008226 DLEC1 protein\_coding 1.434095 6.989737 10.182012

ENSG00000006283 CACNA1G protein\_coding -1.299943 8.125366 -9.891082

ENSG00000026508 CD44 protein\_coding 1.269659 12.298676 9.769961

ENSG00000007350 TKTL1 protein\_coding 1.509492 5.523180 9.510818

ENSG00000006740 ARHGAP44 protein\_coding -1.095506 8.677119 -9.446838

ENSG00000005020 SKAP2 protein\_coding 1.067934 9.160337 9.377365

ENSG00000009950 MLXIPL protein\_coding 1.209825 7.903775 9.361804

ENSG00000014257 ACPP protein\_coding 1.440963 3.438757 9.311729

ENSG00000021488 SLC7A9 protein\_coding 1.009613 3.464679 9.258286

ENSG00000006432 MAP3K9 protein\_coding -1.156165 8.040412 -9.186100

ENSG00000019186 CYP24A1 protein\_coding -1.085143 3.501435 -9.058355

ENSG00000019505 SYT13 protein\_coding -1.873573 8.599696 -8.952456

ENSG00000018280 SLC11A1 protein\_coding 1.476167 8.645577 8.504792

ENSG00000002746 HECW1 protein\_coding -1.168290 8.503463 -8.419466

ENSG00000007968 E2F2 protein\_coding 1.391333 6.027695 8.350419

ENSG00000019991 HGF protein\_coding 1.154795 6.611219 8.212267

ENSG00000021645 NRXN3 protein\_coding -1.093485 9.120658 -8.110309

ENSG00000006327 TNFRSF12A protein\_coding 1.336245 7.752770 7.683296

ENSG00000003137 CYP26B1 protein\_coding -1.181944 7.213652 -7.650578

ENSG00000025708 TYMP protein\_coding 1.068289 7.917529 6.835426

ENSG00000010438 PRSS3 protein\_coding -1.015972 5.789196 -6.720900

ENSG00000008118 CAMK1G protein\_coding -1.589325 6.458758 -6.594047

ENSG00000022355 GABRA1 protein\_coding -1.599751 7.768403 -6.161436

ENSG00000024526 DEPDC1 protein\_coding 1.056587 4.661326 6.078016

ENSG00000006116 CACNG3 protein\_coding -1.526336 6.464034 -5.888424

ENSG00000012223 LTF protein\_coding 1.431839 5.705463 4.687292

ENSG00000011083 SLC6A7 protein\_coding -1.222135 5.251244 -4.446431

ENSG00000006128 TAC1 protein\_coding -1.086536 5.812466 -4.207158

a <- c()

for(mir in rownames(mSeqCounts)){

b <- unlist(strsplit(mir,"-"))

z <- unlist(strsplit(b[length(b)],""))

m <- unlist(strsplit(b[2],""))

if(m[3] == "R"){

m[3] <- "r"

b[2] <- paste(m,collapse = '')

}

if(z[length(z)] == 'p'){

b <- paste(b[1:length(b)-1],collapse = '-')

a <- c(a,b)

} else {

b <- paste(b,collapse = '-')

a <- c(a,b)

}

}

#Read mirCode file

genesIn <- unlist(lapply(mircodes$gene\_id,as.character))

genesIn <- unlist(lapply(genesIn,function(x) unlist(strsplit(x,".",fixed = T))[1]))

#micrornas

gg <- factor(genesIn)

gg <- unlist(lapply(levels(gg),as.character))

matching <- rep(list(NA),length(gg))

names(matching) <- gg

h <- ""

for(r in seq(length(genesIn))){

prefix <- str\_sub(as.character(mircodes$microrna[r]),1,4)

str\_sub(prefix,3,3) <- str\_to\_lower(str\_sub(prefix,3,3))

prefix <- str\_c("hsa-",prefix)

suffix <- str\_sub(as.character(mircodes$microrna[r]),5,)

a <- unlist(strsplit(suffix,"/"))

for(i in seq(length(a))){

b <- unlist(strsplit(a[i],"-"))

if(length(b) == 2){

a[i] <- b[1]

}

}

a <- str\_c(prefix,a)

if(a %in% rownames(mirna\_data)){

if(h != genesIn[r]){

h <- genesIn[r]

matching[[genesIn[r]]] <- a

} else {

matching[[genesIn[r]]] <- c(matching[[genesIn[r]]],a)

}

} else {

h <- genesIn[r]

}

}

ff <- function(li){

ze <- c()

for(i in seq(length(li)))

if(length(li[[i]]) == 0)

ze <- c(ze,i)

return(ze)

}

deLNC00 <- c('ENSG00000260920','ENSG00000242125','ENSG00000261211')

dePC00 <- c('ENSG00000043355','ENSG00000109586','ENSG00000144355')

genes00 <- c(deLNC00, dePC00)

samples00 <- c('TCGA-2F-A9KO-01', 'TCGA-2F-A9KP-01',

'TCGA-2F-A9KQ-01', 'TCGA-2F-A9KR-01',

'TCGA-2F-A9KT-01', 'TCGA-2F-A9KW-01')

rnaExpr00 <- data.frame(matrix(c(2.7,7.0,4.9,6.9,4.6,2.5,

0.5,2.5,5.7,6.5,4.9,3.8,

2.1,2.9,5.9,5.7,4.5,3.5,

2.7,5.9,4.5,5.8,5.2,3.0,

2.5,2.2,5.3,4.4,4.4,2.9,

2.4,3.8,6.2,3.8,3.8,4.2),6,6),

stringsAsFactors=FALSE)

rownames(rnaExpr00) <- genes00

colnames(rnaExpr00) <- samples00

mirExpr00 <- data.frame(matrix(c(7.7,7.4,7.9,8.9,8.6,9.5,

5.1,4.4,5.5,8.5,4.4,3.5,

4.9,5.5,6.9,6.1,5.5,4.1,

12.4,13.5,15.1,15.4,13.0,12.8,

2.5,2.2,5.3,4.4,4.4,2.9,

2.4,2.7,6.2,1.5,4.4,4.2),6,6),

stringsAsFactors=FALSE)

colnames(mirExpr00) <- samples00

rownames(mirExpr00) <- c('hsa-miR-340-5p','hsa-miR-181b-5p',

'hsa-miR-181a-5p', 'hsa-miR-181c-5p',

'hsa-miR-199b-5p','hsa-miR-182-5p')

ceOutput00 <- gdcCEAnalysis(lnc = deLNC00,

pc = dePC00,

lnc.targets = 'starBase',

pc.targets = 'starBase',

rna.expr = rnaExpr00,

mir.expr = mirExpr00)

ceOutput2 <- ceOutput00[ceOutput00$hyperPValue<0.01 &

ceOutput00$corPValue<0.01 & ceOutput00$regSim != 0,]

edges <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'edges')

nodes <- gdcExportNetwork(ceNetwork = ceOutput2, net = 'nodes')

write.table(edges, file='edges.txt', sep='\t', quote=F)

write.table(nodes, file='nodes.txt', sep='\t', quote=F)

oligo\_indexs <- which(metaMatrix.RNA$WHO2016 == "Oligodendroglioma")

astro\_indexs <- which(metaMatrix.RNA$WHO2016 == "Astrocytoma")

oligo\_metaMatrix <- metaMatrix.RNA[oligo\_indexs,]

astro\_metaMatrix <- metaMatrix.RNA[astro\_indexs,]

oligoSeqCounts <- rSeqCounts[,oligo\_indexs]

astroSeqCounts <- rSeqCounts[,astro\_indexs]

oligoM <- mSeqCounts[,oligo\_indexs]

a <- which(is.na(astro\_metaMatrix$gender))

astroDEGAll <- gdcDEAnalysis(counts = astroSeqCounts,

group = astro\_metaMatrix$gender,

comparison = 'male-female',

method = 'limma')

oligoDEGAll <- gdcDEAnalysis(counts = oligoSeqCounts,

group = oligo\_metaMatrix$gender,

comparison = 'male-female',

method = 'limma')

oligo\_deALL <- gdcDEReport(deg = oligoDEGAll, gene.type = 'all')

### DE long-noncoding

oligo\_deLNC <- gdcDEReport(deg = oligoDEGAll, gene.type = 'long\_non\_coding')

### DE protein coding genes

oligo\_dePC <- gdcDEReport(deg = oligoDEGAll, gene.type = 'protein\_coding')

astro\_deALL <- gdcDEReport(deg = astroDEGAll, gene.type = 'all')

### DE long-noncoding

astro\_deLNC <- gdcDEReport(deg = astroDEGAll, gene.type = 'long\_non\_coding')

### DE protein coding genes

astro\_dePC <- gdcDEReport(deg = astroDEGAll, gene.type = 'protein\_coding')

gdcBarPlot(deg = astro\_deALL, angle = 45, data.type = 'RNAseq')

astro\_enrichOutput <- gdcEnrichAnalysis(gene = rownames(astro\_deALL), simplify = TRUE)

oligo\_enrichOutput <- gdcEnrichAnalysis(gene = rownames(oligo\_deALL), simplify = TRUE)

gdcEnrichPlot(astro\_enrichOutput, type = 'bar', category = 'GO', num.terms = 10)

gdcHeatmap00 <- function(deg.id, metadata, rna.expr) {

degDa <- rna.expr[deg.id,]

sampleCol <- ifelse(metadata$WHO2016=='Oligodendroglioma',

'green', 'orange')

#col=colorpanel(75,"darkblue","white","orangered")

lmat = rbind(c(4,3),c(2,1))

lwid = c(2,4)

lhei = c(1,5)

color.palette <- colorRampPalette(c("#F8F8F8","yellow","orange","red"))(n=599)

col\_breaks <- c(seq(-7,-4,length = 100),

seq(-3.99,-0.01,length = 100),

seq(0,2,length = 100),

seq(2.01,4,length = 100),

seq(4.01,8,length = 100),

seq(8.01,12,length = 100),

)

heatmap.2(as.matrix(degDa[,colnames(degDa)]), Colv=FALSE,

col=color.palette,breaks = col\_breaks, trace='none',

cexCol=1.5, cexRow=1,dendrogram='none', srtCol=90,

adjCol=c(0.8,0.15), density.info="none", labRow=NA,

key.title=NA,na.color="red",lwid=lwid, lhei=lhei,

margins =c(5,28), labCol=NA, key.xlab='Log10 Values',

scale='row', ColSideColors = sampleCol)

legend(y=1.14, x=.5, xpd=TRUE,

legend = unique(metadata$WHO2016),

col = c("orange","green"),

lty= 1,

lwd = 5,

cex=.7

)

}

gdcHeatmap00(deg.id = rownames(data),metadata = clin\_all,rna.expr=data)

gdcHeatmap00 <- function(deg.id, metadata, rna.expr) {

degDa <- rna.expr[deg.id,]

sampleCol <- ifelse(metadata$WHO2016=='Oligodendroglioma',

'green', 'orange')

#col=colorpanel(75,"darkblue","white","orangered")

lmat = rbind(c(4,3),c(2,1))

lwid = c(2,4)

lhei = c(1,5)

color.palette <- c("#F8F8F8",colorRampPalette(c("yellow","orange","red")))(n=599)

col\_breaks <- c(seq(0,0.1,length = 1),

seq(0.1,2,length = 100),

seq(2.01,4,length = 100),

seq(4.01,8,length = 100),

seq(8.01,12,length = 100),

seq(-7,-4,length = 100),

seq(-3.99,-0.01,length = 100)

)

heatmap.2(as.matrix(degDa[,colnames(degDa)]), Colv=FALSE,

col=matlab::jet.colors(200), trace='none',

cexCol=0.32, cexRow=0.1,dendrogram='none', srtCol=90,

adjCol=c(0.8,0.15), density.info="none", labRow=NA,

key.title=NA,na.color="red",lwid=lwid, lhei=lhei,

margins =c(3,3), labCol=NA, key.xlab='Normalized intensity',

scale='row', ColSideColors = sampleCol)

legend(y=1.14, x=.5, xpd=TRUE,

legend = unique(metadata$WHO2016),

col = c("orange","green"),

lty= 1,

lwd = 5,

cex=.7

)

}

starburst <- TCGAvisualize\_starburst(met, # DNA methylation with results

DEGAll, # DEG results

group1 = "Astrocytoma",

group2 = "Oligodendroglioma",

filename = "starburst.png",

genome = "hg38",

met.platform = "450K",

met.p.cut = 0.05,

exp.p.cut = 10^-2,

diffmean.cut = 0.15,

logFC.cut = 1,

width = 15,height = 10,

names = TRUE)

DEGAll00 <- gdcDEAnalysis0(counts = rSeqCounts,

group = metaMatrix.RNA$WHO2016,

comparison = 'Astrocytoma-Oligodendroglioma',

method = 'DESeq2',

n.corse = 4)

dd <- mirna\_data[-a,]

dd <- -log10(dd + 0.000000000001)

#########

#200 min pval

sorted\_rows <- rownames(dd\_order[1:50,])

dd\_200 <- dd[sorted\_rows,]

#########

pdf("Heatmap\_miRNAs\_cluster\_50.pdf",height = 12,width = 17)

sampleCol <- ifelse(clin\_all$WHO2016 == 'Astrocytoma','red','blue')

heatmap.2(dd\_200[,colnames(dd\_200)],Colv=FALSE,col = greenred(75),

scale = "column",ColSideColors=sampleCol,dendrogram='row',

key = TRUE,symkey = FALSE,density.info="none",trace = "none",

cexRow=1,key.xlab = 'Log10 values',

main = "Top 50 differentially expressed miRNAs",

labRow = rownames(dd\_200),labCol=NA,margins = c(7,10))

legend(y=1, x=.5, xpd=TRUE,

legend = unique(clin\_all$WHO2016),

col = c("red","blue"),

lty= 1,

lwd = 5,

cex=.7

)

dev.off()

Feature\_o$fraction <- Feature\_o$count/sum(Feature\_o$count)

Feature\_o <- Feature\_o[order(Feature\_o$fraction),]

Feature\_o$ymax <- cumsum(Feature\_o$fraction)

Feature\_o$ymin <- c(0,head(Feature\_o$ymax,n=1))

p1 <- ggplot(Feature\_o,

aes(fill=category,ymax=ymax,ymin=ymin,xmax=4,xmin=3)) +

geom\_rect()+ coord\_polar(theta="y")+xlim(c(0,4))+

theme(panel.grid=element\_blank())+

theme(axis.text=element\_blank())+

theme(axis.ticks=element\_blank())+

annotate("text",x=0,y=0,label="Feature Type")+

labs(title="")

gdcCEAnalysis0 <- function(lnc, pc, deMIR=NULL, lnc.targets='starBase',

pc.targets='starBase', rna.expr, mir.expr) {

hyperOutput <- hyperTestFun(lnc, pc, deMIR,

lnc.targets=lnc.targets, pc.targets=pc.targets)

message ('Step 1/3: Hypergenometric test done !')

regOutput <- multiRegTestFun(hyperOutput, rna.expr=rna.expr,

mir.expr=mir.expr)

message ('Step 2/3: Correlation analysis done !\n',

'Step 3/3: Regulation pattern analysis done !')

ceOutput <- data.frame(hyperOutput, regOutput, row.names=NULL)

return(ceOutput)

}

#### hypergeometric test

hyperTestFun <- function(lnc, pc, deMIR,

lnc.targets='starBase', pc.targets='starBase') {

if (length(lnc.targets) > 1) {

lnc.targets <- lnc.targets

} else {

lnc.targets <- lncTargets[[lnc.targets]]

}

if (length(pc.targets) > 1) {

pc.targets <- pc.targets

} else {

pc.targets <- pcTargets[[pc.targets]]

}

mir1 <- unique(unlist(lnc.targets))

mir2 <- unique(unlist(pc.targets))

mirs <- union(mir1,mir2)

popTotal <- length(mirs)

ceLNC <- lnc[lnc %in% names(lnc.targets)]

cePC <- pc[pc %in% names(pc.targets)]

#ceMIR <- mir[mir %in% mirs]

hyperOutput <- list()

i <- 0

for (lncID in ceLNC) {

listTotal <- length(lnc.targets[[lncID]])

for (gene in cePC) {

i = i + 1

ovlp <- intersect(lnc.targets[[lncID]], pc.targets[[gene]])

popHits <- length(pc.targets[[gene]])

Counts <- length(ovlp)

ovlpMIRs <- paste(ovlp, collapse = ',')

foldEnrichment <- Counts/listTotal\*popTotal/popHits

pValue <- phyper(Counts-1, popHits, popTotal-popHits,

listTotal, lower.tail=FALSE, log.p=FALSE)

ceMIR <- Reduce(intersect, list(ovlp, deMIR))

deMIRs <- paste(ceMIR, collapse = ',')

deMIRCounts <- length(ceMIR)

hyperOutput[[i]] <- c(lncID, gene, Counts, listTotal,

popHits,popTotal,foldEnrichment,pValue,ovlpMIRs,

deMIRCounts, deMIRs)

print(hyperOutput[[i]])

}

}

#hyperOutput <- Reduce(rbind, hyperOutput) ## slower

hyperOutput <- do.call(rbind, hyperOutput)

#hyperOutput <- rbind\_list(hyperOutput) ## not test

colnames(hyperOutput) <- c('lncRNAs','Genes','Counts','listTotal',

'popHits','popTotal','foldEnrichment','hyperPValue','miRNAs',

'deMIRCounts','deMIRs')

hyperOutput <- as.data.frame(as.matrix(hyperOutput),

stringsAsFactors=FALSE)

hyperOutput <- hyperOutput[as.numeric(hyperOutput$Counts)>0,]

#hyperOutput$FDR <- p.adjust(as.numeric(as.character(hyperOutput$pValue)),

#method = 'fdr')

#hyperOutput <- hyperOutput[hyperOutput$Counts>0,]

#hyperOutput$lncRNAs <- ensembl2symbolFun(hyperOutput$lncRNAs)

#hyperOutput$gene <- ensembl2symbolFun(hyperOutput$gene)

if (is.null(deMIR)) {

hyperOutput <- hyperOutput[,! colnames(hyperOutput) %in%

c('deMIRCounts','deMIRs')]

}

print(head(hyperOutput))

return (hyperOutput)

}

########################################################

######## other scores

multiRegFun <- function(lnc, pc, mirs, rna.expr, mir.expr) {

lncDa <- unlist(rna.expr[lnc,])

pcDa <- unlist(rna.expr[pc,])

corpl <- cor.test(pcDa, lncDa, alternative='greater')

ppl <- corpl$p.value

regpl <- corpl$estimate

mirs <- as.character(mirs)

if (mirs == '') {

reg <- NA

lncACT <- NA

partialSen <- NA

cosCol <- NA

} else {

mirs <- unlist(strsplit(mirs, ',', fixed=TRUE))

mirCor <- vapply(mirs, function(mir)

mirCorTestFun(lncDa, pcDa, mir, mir.expr),

numeric(2))

reglm <- mirCor[1,]

regpm <- mirCor[2,]

regSim <- 1-mean((abs(reglm - regpm)/(abs(reglm) + abs(regpm)))

^length(mirs))

#lncACT <- mean((abs(regpl)+abs(reglm)+abs(regpm))/3)

sppc <- mean(regpl-(regpl-reglm\*regpm)/(sqrt(1-reglm^2)\*

sqrt(1-regpm^2)))

#cos <- sum(reglm\*regpm)/(sqrt(sum(reglm^2))\*sqrt(sum(regpm^2)))

#col <- sum(abs(reglm)\*abs(regpm))/(sqrt(sum(abs(reglm)))\*

#sqrt(sum(abs(regpm))))

#cosCol <- (cos+col)/2

}

#scores <- c(regpl, ppl, reg, lncACT, partialSen, cosCol)

scores <- c(cor=regpl, corPValue=ppl, regSim, sppc)

return (scores)

}

multiRegTestFun <- function(hyperOutput, rna.expr, mir.expr) {

samples <- intersect(colnames(rna.expr), colnames(mir.expr))

rna.expr <- rna.expr[,samples]

mir.expr <- mir.expr[,samples]

lncID <- hyperOutput$lncRNAs

pcID <- hyperOutput$Genes

mirID <- hyperOutput$miRNAs

reg <- vapply(seq\_len(nrow(hyperOutput)), function(i)

multiRegFun(lncID[i], pcID[i], mirID[i], rna.expr, mir.expr),

numeric(4))

reg <- t(reg)

colnames(reg) <- c('cor','corPValue','regSim', 'sppc')

return (data.frame(reg))

}

mirCorTestFun <- function(lncDa,pcDa, mir,mir.expr) {

if(mir %in% rownames(mir.expr)){

mirDa <- unlist(mir.expr[mir,])

corlm <- cor.test(lncDa, mirDa, alternative='less')

corpm <- cor.test(pcDa, mirDa, alternative='less')

reglm <- corlm$estimate

regpm <- corpm$estimate

} else {

reglm <- NA

regpm <- NA

}

return (c(reglm, regpm))

}

#mirCorTestFun <- function(lncDa, pcDa, mir, mir.expr) {

# mirDa <- unlist(mir.expr[mir,])

# corlm <- cor.test(lncDa, mirDa, alternative='less')

# corpm <- cor.test(pcDa, mirDa, alternative='less')

# reglm <- corlm$estimate

# regpm <- corpm$estimate

# return (c(reglm, regpm)) ## lnc then pc

#}