Capstone Project - Roberto Atallah

Integrated analysis of lncRNA-miRNA-mRNA ceRNA network in human renal cell carcinoma

Code

deg.R

```r
#   Differential expression analysis with limma
#Set the working directory to where the R scripts are found
setwd("~/Documents/capstone/rscripts/")
#Install BiocManager package to be able to then install the GEOquery and limma packages
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("GEOquery")
BiocManager::install("limma")
#load them
library(GEOquery)
library(limma)
library(dplyr)
# load series and platform data from GEO, this platform is for total RNA
gset <- getGEO("GSE16441", GSEMatrix =TRUE, AnnotGPL=TRUE)
if (length(gset) > 1) idx <- grep("GPL6480", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
pData(gset)$data_processing[1] #Data has already been LOWESS normalized and log2 transformed, it is indicated in
file.
# make proper column names to match toptable
fvarLabels(gset) <- make.names(fvarLabels(gset))
# group membership for all samples
gsms <- "0000000000000000011111111111111111"
sml <- strsplit(gsms, split="")[[1]]
# assign samples to groups and set up design matrix
gs <- factor(sml)
groups <- make.names(c("RCC","Normal"))
levels(gs) <- groups
gset$group <- gs
design <- model.matrix(~group + 0, gset)
colnames(design) <- levels(gs)
#fit linear model
fit <- lmFit(gset, design)
# set up contrasts of interest and recalculate model coefficients
cts <- paste(groups[1], groups[2], sep="-")
cont.matrix <- makeContrasts(contrasts=cts, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
# compute statistics and table of top significant genes
fit2 <- eBayes(fit2, 0.01)
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)
tT <- subset(tT, select=c("ID","adj.P.Val","P.Value","t","B","logFC","Gene.symbol","Gene.title", "Gene.ID"))
#Take table and mutate it based on cut off condition to extract DEGs, remove the genes with no name (undefined)
deg <- tT %>%
  mutate(condition = abs(logFC) > 2 & adj.P.Val < 0.05 & Gene.symbol != "")
# summarize test results
sum(deg$condition == TRUE) #differentially regulated
sum(deg$condition== TRUE & deg$logFC > 0) #upregulated
sum(deg$condition== TRUE & deg$logFC < 0) #downregulated
deg <- subset(deg, deg$condition == TRUE)
#having a file in the capstone folder, called files to store all necessary tables and txt files
write.csv(deg, file="../files/DEG_table.csv", row.names=FALSE)
write.table(deg$Gene.ID, file= "../files/DEG_IDs.txt", row.names = FALSE, col.names = FALSE)
write.table(deg$Gene.symbol, file= "../files/DEG_symbols.txt", row.names = FALSE, col.names = FALSE)
#REGEX to take long non coding RNA from the data, it should start by LINC
write.table(grep("LINC.*", deg$Gene.symbol, value = TRUE), "../files/LNC.txt", row.names = FALSE, col.names = FAL
SE)
#finding top 20 DEGs significant and top 6 LINC, to display it in the report
head_deg <- subset(head(deg, n=20), select= c("Gene.symbol","adj.P.Val","P.Value","logFC"))
lnc <- deg[grep("LINC.*", deg$Gene.symbol), ]
lnc_head <- subset(head(lnc, n=6), select= c("Gene.symbol","adj.P.Val","P.Value","logFC"))
#writing to file
write.table(head_deg, file="../files/DEG_table_head.csv", row.names=FALSE)
write.table(lnc_head, file="../files/LNC_table_head.csv", row.names=FALSE)

# volcano plot (log P-value vs log fold change)
library(ggplot2)
tT %>%
  mutate(Significant = adj.P.Val < 0.05 & abs(logFC) > 2) %>%
  ggplot(aes(x = logFC, y = -log10(adj.P.Val), col=Significant)) + geom_point()
ex <- exprs(gset)
# box-and-whisker plot
ord <- order(gs)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02","#66A61E", "#A6761D", "#B32424", "#B324B3", "#666
666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE16441", "/", annotation(gset), sep ="")
boxplot(ex[,ord], boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs[ord])
legend("topleft", groups, fill=palette(), bty="n")
```

dem.R

```r
#   Differential expression analysis with limma
setwd("~/Documents/capstone/rscripts/")
#load packages
library(GEOquery)
library(limma)
library(dplyr)
# load series and platform data from GEO, this platform is for miRNA
gset <- getGEO("GSE16441", GSEMatrix =TRUE, AnnotGPL=TRUE)
if (length(gset) > 1) idx <- grep("GPL8659", attr(gset, "names")) else idx <- 1
gset <- gset[[idx]]
# make proper column names to match toptable
fvarLabels(gset) <- make.names(fvarLabels(gset))
# group membership for all samples
gsms <- "0000000000000000011111111111111111"
sml <- strsplit(gsms, split="")[[1]]
#Data has already been LOWESS normalized and log2 transformed, it is indicated in file.
# assign samples to groups and set up design matrix
gs <- factor(sml)
groups <- make.names(c("RCC","Normal"))
levels(gs) <- groups
gset$group <- gs
design <- model.matrix(~group + 0, gset)
colnames(design) <- levels(gs)
# fit linear model
fit <- lmFit(gset, design)
# set up contrasts of interest and recalculate model coefficients
cts <- paste(groups[1], groups[2], sep="-")
cont.matrix <- makeContrasts(contrasts=cts, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
# compute statistics and table of top significant genes
fit2 <- eBayes(fit2, 0.01)
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)
tT <- subset(tT, select=c("ID","adj.P.Val","P.Value","t","B","logFC","miRNA_ID","SPOT_ID"))
dem <- tT %>%
  mutate(condition = abs(logFC) > 1 & adj.P.Val < 0.05)
# summarize test results
sum(dem$condition == TRUE)
sum(dem$condition== TRUE & dem$logFC > 0)
sum(dem$condition== TRUE & dem$logFC < 0)
dem <- subset(dem, dem$condition == TRUE)
write.csv(dem, file="../files/dem_table.csv", row.names=FALSE)
write.table(dem$miRNA_ID, file= "../files/dem_symbols.txt", row.names = FALSE, col.names = FALSE)
#finding top 20 DEMs significant
head_dem <- subset(head(dem, n=20), select= c("miRNA_ID","adj.P.Val","P.Value","logFC"))
write.table(head_dem, file="../files/dem_table_head.csv", row.names=FALSE)
# summarize test results
# volcano plot (log P-value vs log fold change)
library(ggplot2)

tT %>%
  mutate(Significant = adj.P.Val < 0.05 & abs(logFC) > 1) %>%
  ggplot(aes(x = logFC, y = -log10(adj.P.Val), col=Significant)) + geom_point()
ex <- exprs(gset)
# box-and-whisker plot
ord <- order(gs)
palette(c("#1B9E77", "#7570B3", "#E7298A", "#E6AB02", "#D95F02","#66A61E", "#A6761D", "#B32424", "#B324B3", "#666
666"))
par(mar=c(7,4,2,1))
title <- paste ("GSE16441", "/", annotation(gset), sep ="")
boxplot(ex[,ord], boxwex=0.6, notch=T, main=title, outline=FALSE, las=2, col=gs[ord])
legend("topleft", groups, fill=palette(), bty="n")
```

code.R

```r
library(dplyr)

setwd("~/Documents/capstone/rscripts/")
deg <- read.csv("../files/DEG_table.csv")
dem <- read.csv("../files/dem_table.csv")
#Reading file of lncRNA miRNA interactions that was taken from DIANA lnc Base
#then taking predicted miRNA ids without duplicates (because multiple lncRNA can interact with the same miRNA)
lncmiR <- read.csv("../files/lncRNA_miRNA.csv")

write.table(unique(lncmiR$miRNA), "../files/miRNA_predicted.txt", row.names = FALSE, col.names = FALSE)
miRNA_predicted <- read.table("../files/miRNA_predicted.txt")
#Finding intersections of miRNA in DEM file and predicted ones
miRNA_intersection <- intersect(dem$miRNA_ID, miRNA_predicted$V1)
write.table(miRNA_intersection, "../files/miRNA_intersection.txt", row.names = FALSE, col.names=FALSE)
#We got 3 miRNAs found in both DEM and predicted."hsa-miR-532-5p" "hsa-miR-140-5p"and "hsa-miR-28-5p" stored
#in the file miRNA_intersection
#in this code, the predictions are made by using only mirWalk database
#MIRWalk Target of miRNA prediction, after using the mirWalk website to do it
mirwalk_pred <- read.csv(file = "../files/miRWalk_miRNA_Targets-3.csv")
#taking only columns of miRNA and target
mirwalk_pred <- mirwalk_pred[, c(1,3)]
#taking the intersections of mRNA predicted with the DEmRNAs found befire
intersection_table <- subset(mirwalk_pred, mirwalk_pred[,2] %in% deg$Gene.symbol)
#removing duplicate rows because some miRNA mRNA interactions are written multiple times (multiple binding sequen
ces)
intersection_table <- distinct(intersection_table)
#writing to table
write.csv(intersection_table, "../files/intersection_table_miRNA_mRNA.csv", row.names = FALSE)
length(unique(mirwalk_pred$genesymbol)) #the number of predicted mRNA
length(unique(intersection_table$genesymbol)) #the number of final mRNA

miRNA_values <- c("hsa-miR-140-5p", "hsa-miR-532-5p", "hsa-miR-28-5p")
#the lncRNA interacting with these 3 miRNA
lncRNA_miRNA_filtered <- subset(lncmiR, miRNA %in% miRNA_values)
write.table(lncRNA_miRNA_filtered, "../files/lncRNA_miRNA_filtered.csv", row.names = FALSE)
colnames(lncRNA_miRNA_filtered)[1:2] <- c("source", "target")
colnames(intersection_table)[1:2] <- c("source", "target")
#adding column called regulation to see if up or down regulated for further color in cytoscape
deg_updated <- deg
deg_updated$regulation <- ifelse(deg_updated$logFC > 0, "up", "down")
dem_updated <- dem
dem_updated$regulation <- ifelse(dem_updated$logFC > 0, "up", "down")
#for loop to see if gene up or down regulated
for (i in 1:nrow(intersection_table)) {
  gene <- intersection_table[i, 2]
  match_row <- which(deg_updated$Gene.symbol == gene)
  if (length(match_row) > 0) {
    regulation <- deg_updated[match_row[1], "regulation"]
  } else {
    regulation <- "NA"
  }
  intersection_table[i, "regulation"] <- regulation
}
for (i in 1:nrow(lncRNA_miRNA_filtered)) {
  mirna <- lncRNA_miRNA_filtered[i, 2]
  match_row <- which(dem_updated$miRNA_ID == mirna)
  if (length(match_row) > 0) {
    regulation <- dem_updated[match_row[1], "regulation"]
  } else {
    regulation <- "NA"
  }
  lncRNA_miRNA_filtered[i, "regulation"] <- regulation
}
cytoscape <- rbind(intersection_table, lncRNA_miRNA_filtered)
write.table(cytoscape, "../files/cytoscape_to_import.csv", row.names = FALSE)
#this is the table to be imported to cytoscape in order to draw the network
#these are the final mRNA that will be used for further PPI network and enrichment analysis
write.table(unique(intersection_table$target), "../files/deg_intersection.txt", row.names = FALSE, col.names = FA
LSE)

#getting top 30 hub genes from both PPI and ceRNA network,
ppi_hub <- read.csv("../files/PPI_hub.csv", header = TRUE)#exported from cytoscape
cerna_hub <- read.csv("../files/ceRNA_hub.csv", header = TRUE)#exported from cytoscape
cerna_hub$name <- gsub('"', '', cerna_hub$name)
#The 3 key genes found (venn diagramm showed) by intersecting PPI and ceRNA hub genes
result <- merge(ppi_hub, cerna_hub, by.x = "display.name", by.y = "name")
result$display.name
#to use in venn diagram
write.csv(ppi_hub$display.name, "../files/ppi_hub_genes.txt", row.names = FALSE, col.names = FALSE)
write.csv(cerna_hub$name, "../files/cerna_hub_genes.txt", row.names = FALSE, col.names = FALSE)
```

code2.R

```r
library(dplyr)
setwd("~/Documents/capstone/rscripts/")
#This is the code that I tried to find the genes that are targets to the three miRNA using 3 DB
#the search was done using the websites of the databases and the tables were downloaded and used here
pred_mirDB <- read.csv("../files/mirDB.csv")
pred_mirTarBase <- read.csv("../files/search_result-2.csv")
target1<- read.csv("../files/Copy of TargetScan8.0__miR-140-5p.predicted_targets.csv")
target2 <- read.csv("../files/Copy of TargetScan8.0__miR-28-5p_708-5p.predicted_targets.csv")
target3 <- read.csv("../files/Copy of TargetScan8.0__miR-532-5p.predicted_targets.csv")
target2 <- subset(target2, target2$Representative.miRNA == "hsa-miR-28-5p")

target1 <- target1[, -9]
target2 <- target2[, -9]
target3 <- target3[, -9]

pred_targetScan1 <- union(target1, target2)
pred_targetScan <- union(pred_targetScan1, target3)
#the one of target scan I performed it one by one so I had to do the union
#here renaming the columns to be the same so I could later join them
colnames(pred_mirDB)[colnames(pred_mirDB)=="Gene.Symbol"] <- "Target"
colnames(pred_mirDB)[colnames(pred_mirDB)=="miRNA.Name"] <- "miRNA"


colnames(pred_targetScan)[colnames(pred_targetScan)=="Target.gene"] <- "Target"
colnames(pred_targetScan)[colnames(pred_targetScan)=="Representative.miRNA"] <- "miRNA"
#taking only miRNA and target from tables because i am only interested in them
pred_mirDB <- pred_mirDB[, c("miRNA", "Target")]
pred_mirTarBase <- pred_mirTarBase[, c("miRNA", "Target")]
pred_targetScan <- pred_targetScan[, c("miRNA", "Target")]
#Combining all three tables together, without duplicates
combined_table <- rbind(pred_mirDB, pred_mirTarBase, pred_targetScan)
combined_table<-unique(combined_table)
#adding rows to take count if one row is found in a table or not
combined_table$mirDB <- integer(nrow(combined_table))
combined_table$targetScan <- integer(nrow(combined_table))
combined_table$mirTarBase <- integer(nrow(combined_table))
#conditions, 1 found, 0 not found
combined_table$mirDB <- ifelse(paste(combined_table$Target, combined_table$miRNA) %in%
                              paste(pred_mirDB$Target, pred_mirDB$miRNA), 1, 0)

combined_table$targetScan <- ifelse(paste(combined_table$Target, combined_table$miRNA) %in%
                                   paste(pred_targetScan$Target, pred_targetScan$miRNA), 1, 0)

combined_table$mirTarBase <- ifelse(paste(combined_table$Target, combined_table$miRNA) %in%
                                   paste(pred_mirTarBase$Target, pred_mirTarBase$miRNA), 1, 0)

#summing, then taking only ones with sum greater or equal than 2
combined_table$Sum <- rowSums(combined_table[,3:5])
subset_table <- subset(combined_table, Sum >= 2)
temp <- subset(head(subset_table, n=20))
#genes intersection predicted vs DEGs, found 17, not enough
subset_table_intersection <- subset(subset_table, subset_table[,2] %in% deg$Gene.symbol)
array<- unique(subset_table_intersection$Target)
write.table(unique(subset_table_intersection)$Target, "../files/genes_intersection.txt", row.names = FALSE, col.n
ames = FALSE)

x <- subset_table_intersection
y <- lncRNA_miRNA_filtered[, 1:2]
merged <- merge(x, y, by.x = "miRNA", by.y = "target")
merged <- merged[, c("source", "miRNA","Target", "mirDB", "targetScan",  "mirTarBase", "Sum")]
colnames(merged)[1] <- "lncRNA"
write.table(merged, "../files/cytoscape_to_import2.csv", row.names = FALSE)
#table to view network
```

enrichment.R

```r
setwd("~/Documents/capstone/rscripts/")
load(dplyr)
#This table was downloaded from DAVID database, and taking top 5 of each category by P value
#PS the data of table is already ordered by P.value
chart<-read.table("../files/chart_4CBB66A493291681316993753.txt", sep="\t", header = TRUE)
cc <- chart[grep("^GOTERM_CC", chart$Category), ][1:5, ]
bp <- chart[grep("^GOTERM_BP", chart$Categorry), ][1:5, ]
mf <- chart[grep("^GOTERM_MF", chart$Category), ][1:5, ]
kegg <- chart[grep("^KEGG", chart$Category), ][1:5, ]

#Restructuring the tables
cc <- cc %>%
  mutate(Category = "CC") %>%
  select(Category, everything())

cc<-cc[,c(1:3, 5)]


bp <- bp %>%
  mutate(Category = "BP") %>%
  select(Category, everything())

bp<-bp[,c(1:3, 5)]


mf <- mf %>%
  mutate(Category = "MF") %>%
  select(Category, everything())

mf<-mf[,c(1:3, 5)]


kegg <- kegg %>%
  mutate(Category = "KEGG") %>%
  select(Category, everything())

kegg<-kegg[,c(1:3, 5)]
#Binding all rows together
enrichement<- rbind(cc, bp, mf, kegg)

temp <- enrichement
temp$Term <- substr(temp$Term, 1, 10)
library(ggplot2)
# create a bar plot of the enrichment data
ggplot(enrichement, aes(x = Count, y = reorder(Term, Count), fill = Category)) +
  geom_bar(stat = "identity") +
  xlab("Count") +
  ylab("Row") +
  ggtitle("Enrichement Data by Row Name")

write.csv(enrichement, "../files/enrichement.csv")
```

enrichment2.R

```r
setwd("~/Documents/capstone/rscripts/")
load(dplyr)
#This table was downloaded from DAVID database, and taking top 5 of each category by P value
#PS the data of table is already ordered by P.value
chart<-read.table("../files/chart_14FC45D11CE21683469320556.txt", sep="\t", header = TRUE)
cc <- chart[grep("^GOTERM_CC", chart$Category), ][1:5, ]
bp <- chart[grep("^GOTERM_BP", chart$Category), ][1:5, ]
mf <- chart[grep("^GOTERM_MF", chart$Category), ][1:5, ]
kegg <- chart[grep("^KEGG", chart$Category), ][1:5, ]

#Restructuring the tables
cc <- cc %>%
  mutate(Category = "CC") %>%
  select(Category, everything())

cc<-cc[,c(1:3, 5)]


bp <- bp %>%
  mutate(Category = "BP") %>%
  select(Category, everything())

bp<-bp[,c(1:3, 5)]


mf <- mf %>%
  mutate(Category = "MF") %>%
  select(Category, everything())

mf<-mf[,c(1:3, 5)]


kegg <- kegg %>%
  mutate(Category = "KEGG") %>%
  select(Category, everything())

kegg<-kegg[,c(1:3, 5)]
#Binding all rows together
enrichement<- rbind(cc, bp, mf, kegg)

temp <- enrichement
temp$Term <- substr(temp$Term, 1, 10)
library(ggplot2)
# create a bar plot of the enrichment data
ggplot(enrichement, aes(x = Count, y = reorder(Term, Count), fill = Category)) +
  geom_bar(stat = "identity") +
  xlab("Count") +
  ylab("Row") +
  ggtitle("Enrichement Data by Row Name")

write.csv(enrichement, "../files/enrichement2.csv")
```