

DEG Analysis Male Tumor vs. Tumor Adjacent

Annika Jorgensen

1-07/2024

R Markdown

Title: "DEG_Analysis Male Tumor vs. Tumor"

Author: Annika Jorgensen

Date: 01/07/2024

Purpose: This document is for the author to parse out the Male tumor vs. tumor adjacent code from the "DEG_changed_comparison" file as well as demonstrate understanding of the code and theory.

Libraries

The first chunk of code is dedicated to installing the libraries. These libraries are to help execute the differential analysis and helps visualize the data. The code was not included for concision.

Defining Colors

This chunk defines color palette variables that are going to be used in plots later on the script. These variables are defined by converting BrewerCode palettes into palettes that can be used in R.

```
viralPalette <- brewer.pal(8, "Set1")
hbvColor <- viralPalette[1]
hcvColor <- viralPalette[2]
bothColor <- viralPalette[3]
neitherColor <- viralPalette[4]

sexTissuePalette <- brewer.pal(12, "Paired")
maleTumorColor <- sexTissuePalette[4]
maleAdjacentColor <- sexTissuePalette[3]
femaleTumorColor <- sexTissuePalette[6]
femaleAdjacentColor <- sexTissuePalette[5]
```

Read in data

This code is where you read in all the data files that are going to be used in the script. The data is also converted into a variety of variables that makes the data easier to handle. The data is also cleaned up to make sure the analysis done later is accurate and precise.

```
metadata <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/metadata_for_de.csv", row.names = 1)
tumorAdjacentExp <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/japan_all_samples_salmon.csv")
colnames(tumorAdjacentExp) <- gsub("\\.", "-", colnames(tumorAdjacentExp)) #changing the column names

# Importing gene annotations
```

```

#genes <- read.table("gencode.v25.chr_patch_hapl_scaff.annotation.bed", header=FALSE, sep="\t")
genes <- read.table("~/Desktop/ResearchProjects/LiverCancer/Metadata/gencodeTranscripts.txt", header=TRUE)
genes <- data.frame(genes)
tumorAdjacentExp <- tumorAdjacentExp[rownames(tumorAdjacentExp) %in% genes$GENEID ,]
genes <- genes[match(rownames(tumorAdjacentExp), genes$GENEID),]
# Calculating gene length, this is needed for calculating the FPKM values
genes$length <- with(genes, end - start)

# Removing Samples due to low quality
metadata <- metadata[!(metadata$ID == "RK023") ,]
metadata <- metadata[!(metadata$ID == "RK106") ,]
metadata <- metadata[!(metadata$ID == "RK113") ,]
metadata <- metadata[!(metadata$ID == "RK135") ,]
metadata <- metadata[!(metadata$ID == "RK105") ,]
metadata <- metadata[!(metadata$ID == "RK116") ,]
metadata <- metadata[!(metadata$ID == "RK066") ,]
metadata <- metadata[!(metadata$ID == "RK096") ,]

#Removing both and NBNC samples
metadata <- metadata[!(metadata$Virus_infection == "NBNC"),]
metadata <- metadata[!(metadata$Virus_infection == "both"),]

# Subsetting and ordering metadata to match the count matrix
tumorAdjacentExpSubset <- tumorAdjacentExp[, colnames(tumorAdjacentExp) %in% metadata$sampleid]
metadataSubset <- metadata[metadata$sampleid %in% colnames(tumorAdjacentExpSubset),]
metadataSubset <- metadataSubset[match(colnames(tumorAdjacentExpSubset), metadataSubset$sampleid),]
rownames(metadataSubset) <- metadataSubset$sampleid

# Adding tissue type, converting categorical variables to factors
metadataSubset$tumor <- as.numeric(grepl('tumor', metadataSubset$sampleid, ignore.case=T))

#Swapping lesion type for sample RK169
metadataSubset["RK169-tumor-XY","tumor"] <- 0
metadataSubset["RK169-adjacent-XY","tumor"] <- 1

#Changing rownames to match swapped lesion type
rownames(metadataSubset)[rownames(metadataSubset)==="RK169-tumor-XY"] <- "RK169_adjacent-XY"
rownames(metadataSubset)[rownames(metadataSubset)==="RK169-adjacent-XY"] <- "RK169_tumor-XY"
rownames(metadataSubset)[rownames(metadataSubset)==="RK169_adjacent-XY"] <- "RK169-adjacent-XY"

rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)==="RK169-tumor-XY"] <- "RK169_adjacent"
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)==="RK169-adjacent-XY"] <- "RK169_tumor"
rownames(tumorAdjacentExpSubset)[rownames(tumorAdjacentExpSubset)==="RK169_adjacent-XY"] <- "RK169-adjacent"

#Swapping lesion type for sample RK065
metadataSubset["RK065-tumor-XX","tumor"] <- 0
metadataSubset["RK065-adjacent-XX","tumor"] <- 1

#Changing rownames in metadata to match swapped lesion type
rownames(metadataSubset)[rownames(metadataSubset)==="RK065-tumor-XY"] <- "RK065_adjacent-XY"
rownames(metadataSubset)[rownames(metadataSubset)==="RK065-adjacent-XY"] <- "RK065_tumor-XY"
rownames(metadataSubset)[rownames(metadataSubset)==="RK065_adjacent-XY"] <- "RK065-adjacent-XY"

```

```

rownames(tumorAdjacentExpSubset) [rownames(tumorAdjacentExpSubset)=="RK065-tumor-XY"] <- "RK065_adjacent"
rownames(tumorAdjacentExpSubset) [rownames(tumorAdjacentExpSubset)=="RK065-adjacent-XY"] <- "RK065_tumor"
rownames(tumorAdjacentExpSubset) [rownames(tumorAdjacentExpSubset)=="RK065_adjacent-XY"] <- "RK065-adjacent"

metadataSubset$gender_tissue <- paste(metadataSubset$Gender, metadataSubset$tumor, sep="_")
metadataSubset$gender_tissue_viral <- paste(metadataSubset$gender_tissue, metadataSubset$Virus_infection)
metadataSubset$library_type <- metadataSubset$strandedness
metadataSubset$library_type <- factor(metadataSubset$library_type)
metadataSubset$tumor <- factor(metadataSubset$tumor)
metadataSubset$Ta <- factor(metadataSubset$Ta)
metadataSubset$Portal_vein_invasion <- factor(metadataSubset$Portal_vein_invasion)
metadataSubset$Hepatic_vein_invasion <- factor(metadataSubset$Hepatic_vein_invasion)
metadataSubset$Bile_duct_invasion <- factor(metadataSubset$Bile_duct_invasion)
metadataSubset$Liver_fibrosis <- factor(metadataSubset$Liver_fibrosis)
metadataSubset$Prognosis <- factor(metadataSubset$Prognosis)

# Creating the DGEList object
dge <- DGEList(counts=tumorAdjacentExpSubset, genes=genes)
colnames(dge) <- colnames(tumorAdjacentExpSubset)
dge$samples$sex <- metadataSubset$Gender
dge$samples$viral <- factor(metadataSubset$Virus_infection)
dge$samples>ID <- metadataSubset$ID
dge$samples$tumor <- metadataSubset$tumor
dge$samples$gender_tissue <- metadataSubset$gender_tissue
dge$samples$gender_tissue_viral <- metadataSubset$gender_tissue_viral
dge$samples$library_type <- metadataSubset$library_type
dge$samples$edmonson_grade <- metadataSubset$Edmonson_grade
dge$samples$Ta <- metadataSubset$Ta
dge$samples$survival <- metadataSubset$Overall_survival_month
dge$samples$smoking <- factor(metadataSubset$Smoking)
dge$samples$alcohol <- factor(metadataSubset$Alcohol_intake)
dge$samples$fibrosis <- factor(metadataSubset$Liver_fibrosis)

# Inspecting the N of samples in each group
table(dge$samples$gender_tissue_viral)

## 
## F_0_HBV F_0_HCV F_1_HBV F_1_HCV M_0_HBV M_0_HCV M_1_HBV M_1_HCV
##      7       32      8      33      31      59      37      71

# =====
# Filtering expression data
# =====

# Keeping genes that have a mean FPKM of at least 0.5 in at least one of the
# groups under investigation and at least 6 reads in at least 10 samples
fpkm <- rpkm(dge, gene.length=dge$genes$length)

M_1_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_1_HBV")], 1, mean, na.rm=TRUE)
M_0_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_0_HBV")], 1, mean, na.rm=TRUE)
M_1_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_1_HCV")], 1, mean, na.rm=TRUE)
M_0_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="M_0_HCV")], 1, mean, na.rm=TRUE)

```

```

F_1_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_1_HBV")], 1,mean,na.rm=TRUE)
F_0_HBV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_0_HBV")], 1,mean,na.rm=TRUE)
F_1_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_1_HCV")], 1,mean,na.rm=TRUE)
F_0_HCV_mean_fpkm <- apply(as.data.frame(fpkm)[(dge$samples$gender_tissue_viral=="F_0_HCV")], 1,mean,na.rm=TRUE)

keep <- (M_1_HBV_mean_fpkm > 0.5 | M_0_HBV_mean_fpkm > 0.5 |
           M_1_HCV_mean_fpkm > 0.5 | M_0_HCV_mean_fpkm > 0.5 |
           F_1_HBV_mean_fpkm > 0.5 | F_0_HBV_mean_fpkm > 0.5 |
           F_1_HCV_mean_fpkm > 0.5 | F_0_HCV_mean_fpkm > 0.5 )

dge <- dge[keep,,keep.lib.sizes=FALSE]
dge <- calcNormFactors(dge, method="TMM")
keep <- rowSums(dge$counts > 6) >= 10
dge <- dge[keep,,keep.lib.size=FALSE]
dge <- calcNormFactors(dge, method="TMM")

# N of genes retained after filtering
dim(dge$genes)

```

```
## [1] 12465      7
```

DGE Analysis tumor vs. tumor- adjacent

This section is doing voom/limma and Go/KEGG analysis on all of the tumor vs. tumor adjacent samples

Design matrix

This code chunk is creating a design matrix to analyze tumor vs. tumor adjacent regardless of sex. This design matrix creates a model matrix that takes the inputs that we want to consider for later limma/voom analysis.

In this specific design matrix we are considering all tumor samples and all tumor adjacent samples.

The colnames code is just renaming columns of the design matrix to identify what they are for easier readability.

```

# =====
# =====
# Analysis of all tumor vs. tumor-adjacent regardless of sex
# =====
# =====

# Creating a new design model matrix with the variable of interest and the
# library type
design <- model.matrix(~0+dge$samples$tumor+dge$samples$library_type)

#I just put design matrix in twice because I didn't remember what inputs to put in.
identical(design,design,num.eq=TRUE)

## [1] TRUE

colnames(design) <- gsub("dge\\\$samples\\\$tumor", "tumor", colnames(design))
colnames(design) <- gsub("dge\\\$samples\\\$library_typeunstranded", "library_type", colnames(design))
head(design)

```

```

##   tumor0 tumor1 library_type
## 1     1     0             1
## 2     0     1             1
## 3     1     0             1
## 4     0     1             1
## 5     1     0             1
## 6     0     1             1

```

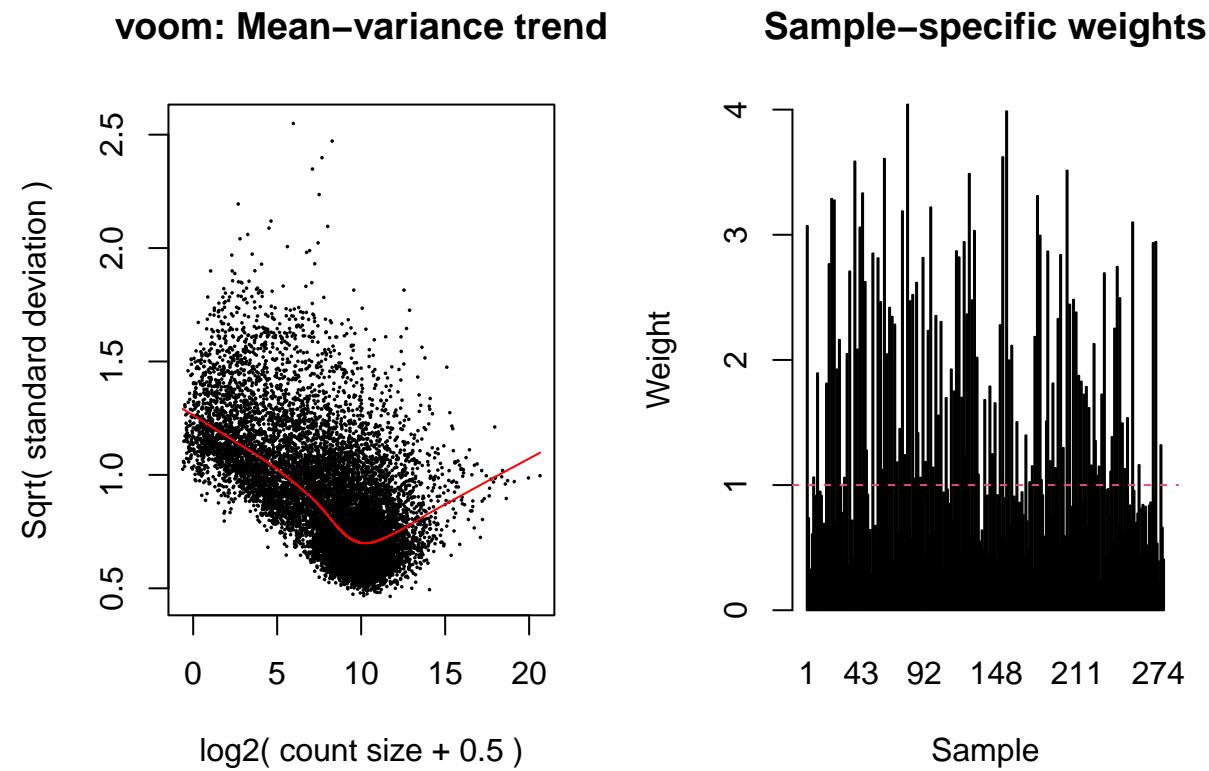
voom

voom is a function that lies within a package called limma. limma/voom is used in DGE analysis. voom is a function that takes the counts in a metadata set and transforms them into log2 of TMM values calculated above in the normalization factors. A linear model is then fitted to the TMM for each gene and residuals are calculated. A smoothed curve is then fitted to the square root of the residual standard deviation by the average expression (this is the red line). This smooth curve is then used to obtain weights for each gene and sample that are passed into limma along with the TMM values.

```

# Running voom again with the new design matrix.
v <- voomWithQualityWeights(dge, design, plot=TRUE)

```



limma

This section marks the beginning of running limma. limma creates a linear fit to the data, makes comparisons of the fitted data, and applies Bayes smoothing. limma starts by creating a variable that has all of the duplicate correlation values on v and design. These correlation values will be used later in a linear fit.

```

# =====
# Differential expression analysis with limma - all male tumor adjacent vs. non-tumor-adjacent
# =====

# Block design for individual. This is used in tumor-normal comparisons with
# paired samples.
corfit <- duplicateCorrelation(v, design, block = v$targets$ID)
# This should give a positive correlation value. It represents the
# correlation between measurements made on the same person.
corfit$consensus

## [1] 0.17723

```

limma graph

This is the linear model with limma, notice that the correlation values were the duplicate correlations used earlier

```

# Fitting the linear model with limma.
# If using paired samples, the within-patient correlation and a block design
# for patient is used to account for pairwise samples
fit <- lmFit(v, design, block = v$targets$ID, correlation = corfit$consensus)

```

Coefficient vector

This code chunk involves extracting coefficients from the linear fit model and storing them in a vector for later use.

```

# Contrast design for differential expression
# Defining pairwise comparisons
contrasts <- makeContrasts(Adjacent_vs_Tumor = tumor1 - tumor0,
                            levels=colnames(design))
head(contrasts)

##          Contrasts
## Levels      Adjacent_vs_Tumor
## tumor0           -1
## tumor1            1
## library_type      0

# Assigning all comparisons to a vector for later
allComparisons <- colnames(contrasts)

```

Contrast Analysis

This next code chunk reorients the linear model obtained earlier and obtains the coefficients and standard errors from the model. This step also sets us up to apply Empirical Bayes smoothing.

```

# Running contrast analysis
vfit <- contrasts.fit(fit, contrasts = contrasts)
# Look at N of DEGs with adj. p <0.01 and log2FC>2
summary(decideTests(vfit, adjust.method = "BH", p.value = 0.05, lfc = 2))

```

```

##          Adjacent_vs_Tumor
## Down                  499
## NotSig                11808
## Up                   158

```

Bayes smoothing

This code chunk uses Empirical Bayes smoothing to plot the final model after doing the limma and voom analysis.

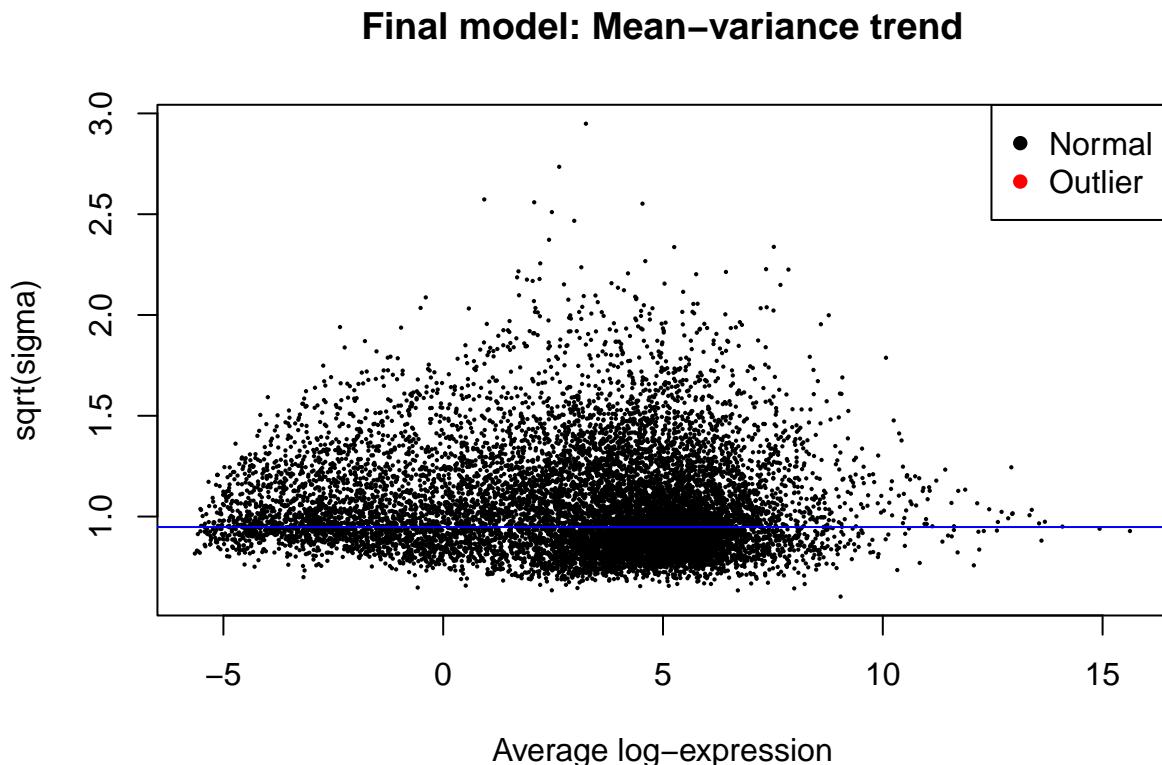
Empirical Bayes smoothing is a way to account for uncertainty. The technique uses the population in a region as a measure of confidence. Meaning that areas with low margin of error are left untouched while estimates with higher margin of error are moved closer to the global average.

The write.csv is commented out because it is not needed for this file

```

# Computing differential expression based on the empirical Bayes moderation of
# the standard errors towards a common value. Robust = should the estimation of
# the empirical Bayes prior parameters be robustified against outlier sample
# variances?
veBayesFit <- eBayes(vfit, robust=TRUE)
plotSA(veBayesFit, main = "Final model: Mean-variance trend")

```



```

vTopTable <- topTable(veBayesFit, n=Inf, p.value=1, lfc=0)
DEGs <- topTable(veBayesFit, n=Inf, p.value=0.05, lfc=2)
#DEGs_print <- data.frame(DEGs$GENEID, DEGs$gene_name, DEGs$adj.P.Val, DEGs$logFC)
#write.csv(DEGs_print, "~/R/gene_list_tumor_vs_tumor_adjacent.csv")

```

DGE Analysis Male tumor vs. tumor- adjacent

This section is doing voom/limma and Go/KEGG analysis on all of the Male tumor vs. tumor adjacent samples

Design matrix

This code chunk is setting up a second design matrix for a voom analysis of tumor vs. non-tumor differentiated by sex.

```
# =====
# =====
# Analysis of tumor vs. non-tumor differentiated by sex
# =====
# =====

# =====
# voom transformation
# =====

# Creating a design model matrix with the variable of interest

#Added library type and Targets to the design matrix
#design <- model.matrix(~0+dge$samples$gender_tissue_viral)

design <- model.matrix(~0+v$targets$gender_tissue+v$targets$library_type)

identical(design,design,num.eq=TRUE)

## [1] TRUE

#Added target and library type column names
colnames(design) <- gsub("v\\$targets\\$gender_tissue", "", colnames(design))
colnames(design) <- gsub("v\\$targets\\$library_typeunstranded", "library_type", colnames(design))

head(design)

##   F_0 F_1 M_0 M_1 library_type
## 1  0  0  1  0          1
## 2  0  0  0  1          1
## 3  0  0  1  0          1
## 4  0  0  0  1          1
## 5  0  0  1  0          1
## 6  0  0  0  1          1
```

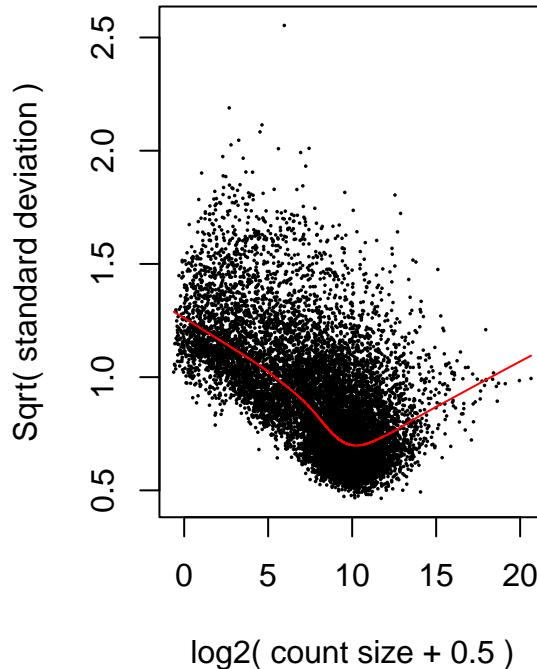
Voom for Male Tumor vs. Tumor adjacent

This code chunk is using the voom function and outputting the graph.

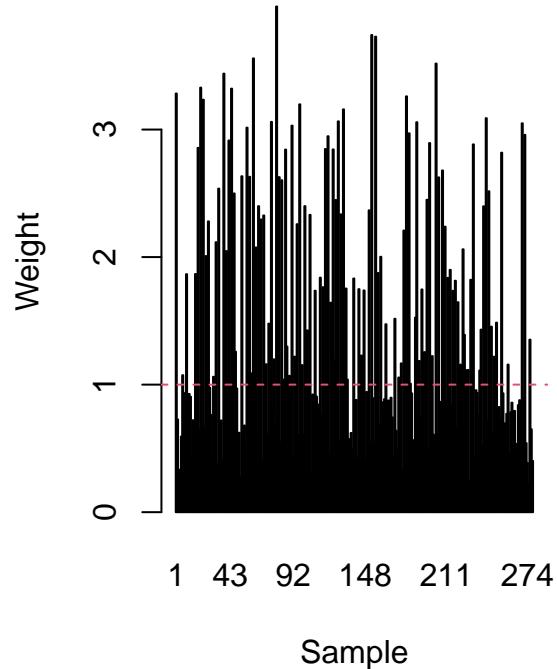
```
# Running voom with quality weights. Normalizes expression intensities so that
# the log-ratios have similar distributions across a set of samples.
# To quantile normalize, add normalize.method="quantile"
# Running parallel
```

```
#pdf("~/Desktop/Research Projects/Liver Cancer Project/Figures/ male tumor tumor adjacent voom.pdf", width=10, height=8)
v <- voomWithQualityWeights(dge, design, plot=TRUE)
```

voom: Mean–variance trend



Sample-specific weights



```
dev.off()
```

```
## null device
## 1
```

limma

This code chunk is for doing paired samples correlation tests. The value of the correlation should be positive.

```
# Block design for individual. This is used in tumor-normal comparisons with
# paired samples.
corfit <- duplicateCorrelation(v, design, block = v$targets$ID)
# This should give a positive correlation value. It represents the
# correlation between measurements made on the same person.
corfit$consensus
```

```
## [1] 0.1758979
```

limma graph This code chuck is doing a linear fit model with limma for male tumor vs tumor-adjacent.

```
# Fitting the linear model with limma.
# If using paired samples, the within-patient correlation and a block design
# for patient is used to account for pairwise samples
fit <- lmFit(v, design, block = v$targets$ID, correlation = corfit$consensus)
```

Coefficient vector

This code chunk does pairwise comparisons of the male tumor vs. tumor adjacent sample for contrast design for differential expressions and stores the comparisons into a vector for later.

```
# Contrast design for differential expression
# Defining pairwise comparisons

contrasts <- makeContrasts(Adjacent_M_vs_Tumor_M = M_1 - M_0,
                            Adjacent_F_vs_Tumor_F = F_1 - F_0,
                            levels=colnames(design))

#Couldn't make the pair wise comparisions would so I just picked some random column names.
#contrasts <- makeContrasts(Adjacent_F_vs_Tumor_F = F_1_HBV - F_0_HBV,
#                           #Adjacement_M_vs_Tumor_M=M_1_both- M_0_both,
#                           #levels=colnames(design))

head(contrasts)

##          Contrasts
## Levels      Adjacent_M_vs_Tumor_M Adjacent_F_vs_Tumor_F
##   F_0                  0                   -1
##   F_1                  0                    1
##   M_0                 -1                   0
##   M_1                  1                   0
##   library_type          0                   0

# Assigning all comparisons to a vector for later
allComparisons <- colnames(contrasts)
```

Contrast Analysis

This code is running a contrast analysis using the data from the limma model.

```
# Running contrast analysis
vfit <- contrasts.fit(fit, contrasts = contrasts)
# Looking at N of DEGs with adj. p <0.01 and log2FC>2
summary(decideTests(vfit, adjust.method = "BH", p.value = 0.05, lfc = 2))
```

	Adjacent_M_vs_Tumor_M	Adjacent_F_vs_Tumor_F
## Down	524	471
## NotSig	11790	11836
## Up	151	158

Bayes smoothing

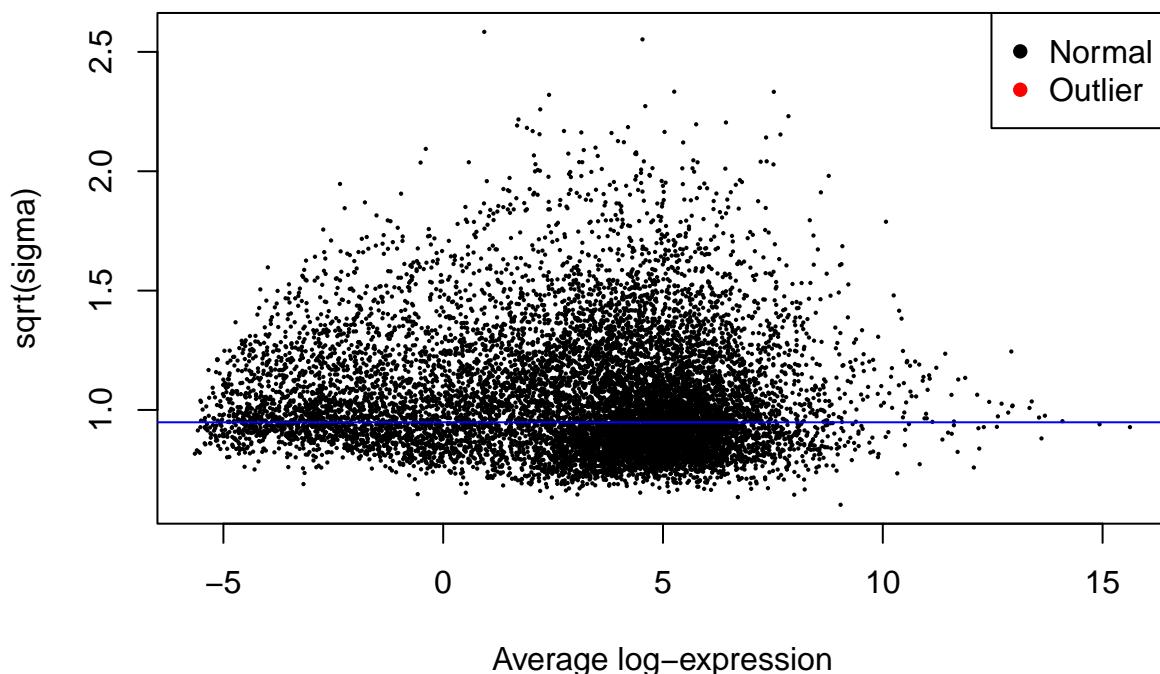
Bayes smoothing on data from the limma model. Empirical Bayes smoothing increases the power of the limma model.

```

# Computing differential expression based on the empirical Bayes moderation of
# the standard errors towards a common value. Robust = should the estimation of
# the empirical Bayes prior parameters be robustified against outlier sample
# variances?
veBayesFit <- eBayes(vfit, robust=TRUE)
#pdf("~/Desktop/Research Projects/Liver Cancer Project/Figures/male tumor tumor adjacent final model.pdf")
plotSA(veBayesFit, main = "Final model: Mean-variance trend")

```

Final model: Mean–variance trend



```
dev.off()
```

```

## null device
##          1

vTopTable_M <- topTable(veBayesFit, coef=1, n=Inf, p.value=1, lfc=0)
write.csv(vTopTable_M, "~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpression_M_topTable.csv")

vTopTable_F <- topTable(veBayesFit, coef=2, n=Inf, p.value=1, lfc=0)
write.csv(vTopTable_F, "~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpression_F_topTable.csv")

DEGs_M <- topTable(veBayesFit, coef=1, n=Inf, p.value=0.05, lfc=2)
DEGs_F <- topTable(veBayesFit, coef=2, n=Inf, p.value=0.05, lfc=2)
DEGs_F_relax_p <- topTable(veBayesFit, coef=2, n=Inf, p.value=0.05, lfc=2)

```

Data Visualization

This section creates Volcano plots for easy visualization of the gene lists.

Volcano Plot Object

```
# =====
#Volcano plot of female tumor vs tumor-adjacent
# =====

df <- data.frame(vTopTable_F$adj.P.Val, vTopTable_F$logFC, vTopTable_F$chr, vTopTable_F$GENEID, vTopTable_F$name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.01),]$id

dfAnons <- subset(df, chr != "chrX" & chr != "chrY" & !(id %in% dfSig))
dfAnons <- cbind(dfAnons, rep(1, nrow(dfAnons)))
colnames(dfAnons)[6] <- "Color"

dfXnons <- subset(df, chr == "chrX" & !(id %in% dfSig))
dfXnons <- cbind(dfXnons, rep(2, nrow(dfXnons)))
colnames(dfXnons)[6] <- "Color"

dfYnons <- subset(df, chr == "chrY" & !(id %in% dfSig))
dfYnons <- cbind(dfYnons, rep(3, nrow(dfYnons)))
colnames(dfYnons)[6] <- "Color"

dfa <- subset(df, chr != "chrX" & chr != "chrY" & id %in% dfSig)
dfa <- cbind(dfa, rep(4, nrow(dfa)))
colnames(dfa)[6] <- "Color"

dfX <- subset(df, chr == "chrX" & id %in% dfSig)
dfX <- cbind(dfX, rep(5, nrow(dfX)))
colnames(dfX)[6] <- "Color"

dfY <- subset(df, chr == "chrY" & id %in% dfSig)
dfY <- cbind(dfY, rep(6, nrow(dfY)))
colnames(dfY)[6] <- "Color"

dfPlot <- rbind(dfAnons, dfXnons, dfYnons, dfa, dfX, dfY)
dfPlot$Color <- as.factor(dfPlot$Color)
```

Volcano Plot

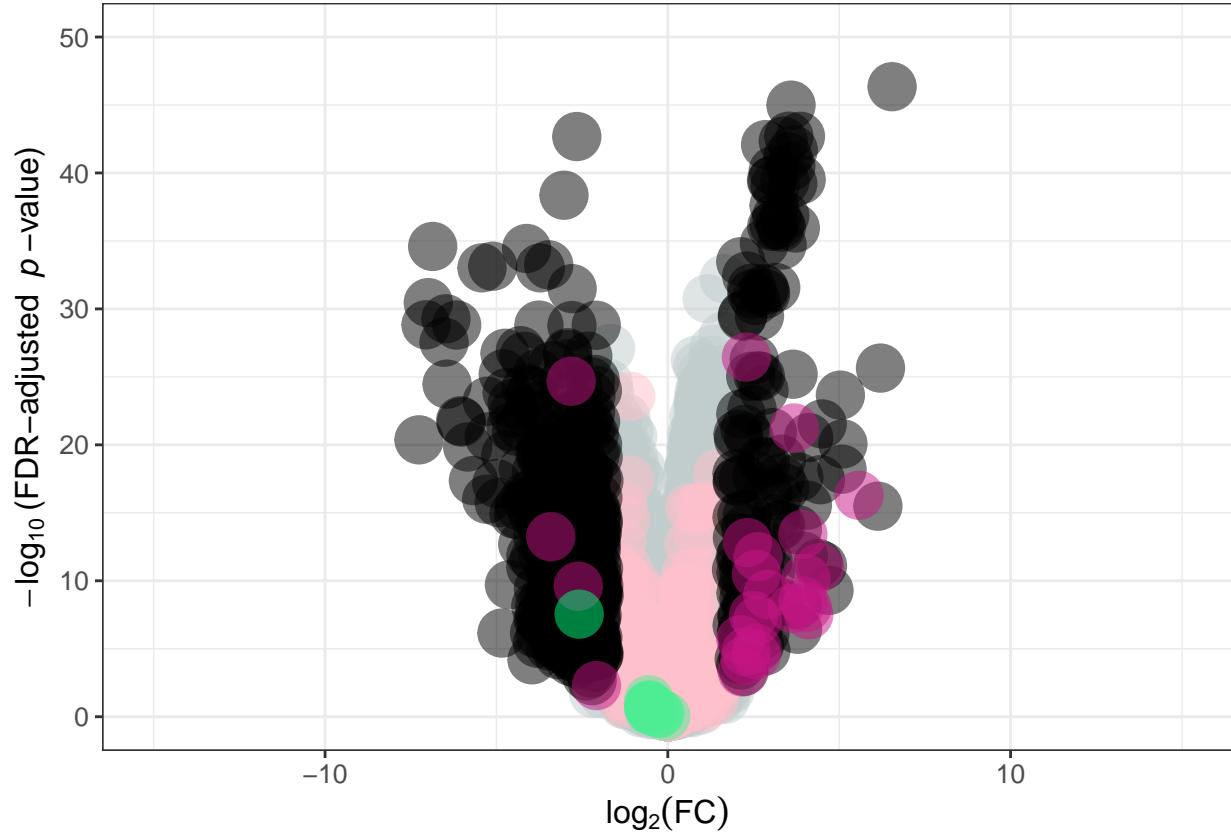
This code chunk plots the female tumor vs. tumor adjacent volcano plot.

```
p <- ggplot(data = dfPlot, aes(x = logFC, y = -log10(adj.P.Val), color=Color )) +
  geom_point(alpha = 0.5, size = 8) +
  theme_bw() +
  theme(legend.position = "none") +
  xlim(c(-15, 15)) + ylim(c(0, 50)) +
  scale_color_manual(values = c("azure3", "pink", "seagreen2", "black", "mediumvioletred", "springgreen"))
  labs(x=expression(log[2](FC)),
       y=expression(-log[10] ~ "(FDR-adjusted " ~ italic("p") ~ "-value)")) +
  theme(axis.title.x=element_text(size=12),
```

```

    axis.text.x=element_text(size=10)) +
  theme(axis.title.y=element_text(size=12),
        axis.text.y=element_text(size=10))
p

```



```
forLabel <- subset(dfPlot, adj.P.Val<=0.01 & abs(logFC)>=2)
```

Significance Thresholds This code chunk is setting the significance thresholds for the volcano plot. The thresholds can be seen via dotted lines on the graph.

```

# Adding lines for significance thresholds
pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumour")
p + geom_hline(yintercept = 2, colour="#000000", linetype="dashed")
) + geom_vline(xintercept = 2, colour="#000000", linetype="dashed")
) + geom_vline(xintercept = -2, colour="#000000", linetype="dashed")
) + geom_text_repel(data=forLabel, max.iter=1000, box.padding = 0.25, force=1, aes(x = logFC, y = -log10(padj)))
## Warning: ggrepel: 589 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
#dev.off()

```

```

df <- data.frame(vTopTable_F$adj.P.Val, vTopTable_F$logFC, vTopTable_F$chr, vTopTable_F$GENEID, vTopTable_F$name)
colnames(df) <- c("adj.P.Val", "logFC", "chr", "id", "name")
dfSig <- df[(abs(df$logFC) >= 2 & df$adj.P.Val <= 0.05),]$id

p <- ggplot(data=df, aes(x=logFC, y=-log10(adj.P.Val))) + geom_point(alpha=0.5)+theme_bw()

p2 <- p+ geom_vline(xintercept=c(-2, 2), linetype= "dashed", size=1.5) + geom_hline(yintercept= 2, linetype= "dashed", size=1.5)

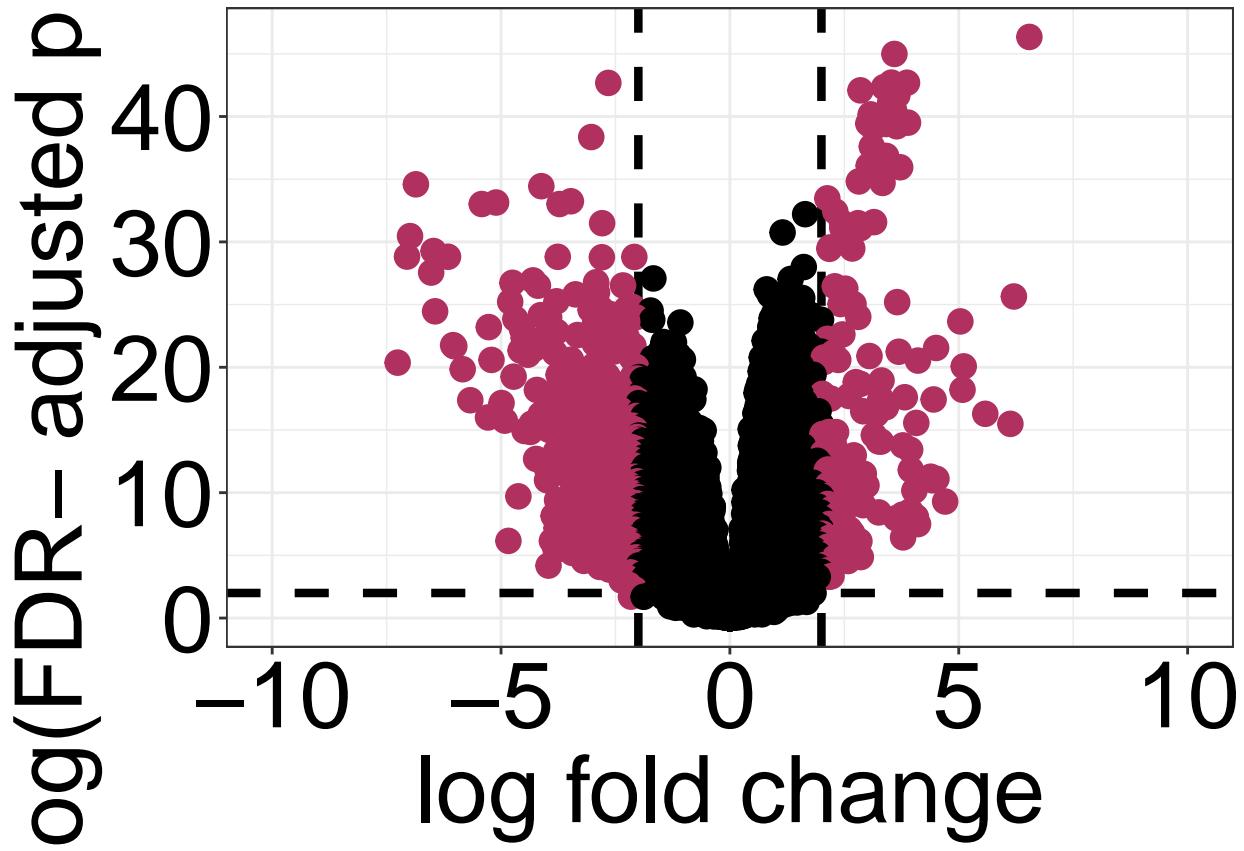
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

p3 <- p2+ theme(legend.position = c(0.925,0.75)) + geom_point(aes(color=abs(logFC)>=2 & adj.P.Val<=0.05), size=35)

p4 <- p3+theme(axis.title = element_text(size=35), axis.text = element_text(size=35, color="black"), legend.position = c(0.925,0.75))

p4

```



```

pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumor_DEA.pdf")
p4
dev.off()

```

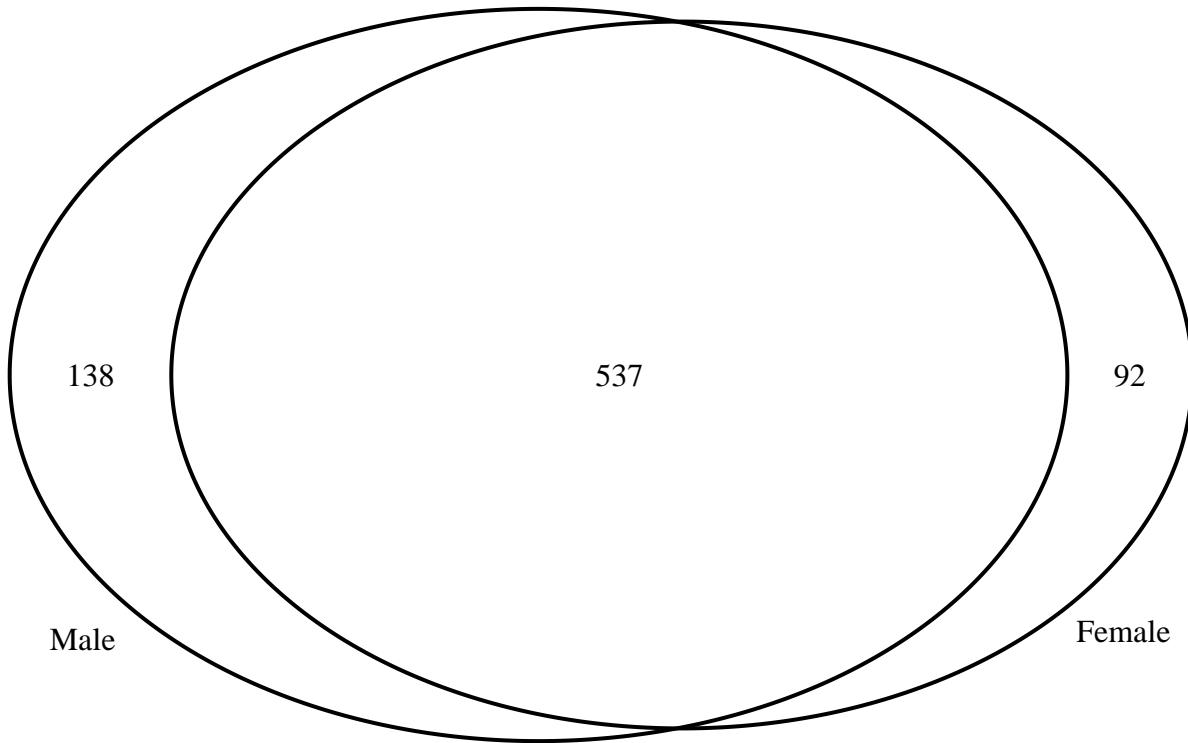
```
## pdf  
## 2
```

Venn Diagram

In this code chunk two Venn Diagrams are made. The commented out sections are the original code sections. The first two sections are creating the Venn Diagrams. Given that I have a different working directory and am doing this is an RMD file not an R file. I rewrote the code to create the Venn Diagrams and output them into the RMD file which you see below.

The third section is inputting data from a CSV file. I do not have the said CSV files so I have commented these sections out. These samples show the overlap between male and female samples.

```
library(VennDiagram)  
library(grDevices)  
#venn.diagram(List("Female"=DEGs_F$gene_name, "Male"=DEGs_M$gene_name), filename = "~/3.0 Hasting Rese  
  
#venn.diagram(List("Female"=DEGs_F_relax_p$gene_name, "Male"=DEGs_M$gene_name), filename = "~/3.0 Haste  
  
#write.csv(DEGs_F,("~/3.0 Hasting Research/Liver Cancer Analysis/DEG Figures/Female_all_DEGs.csv")  
#write.csv(DEGs_M, "~/3.0 Hasting Research/Liver Cancer Analysis/DEG Figures/Male_all_DEGs.csv")  
  
write.csv(DEGs_F, "~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnal  
write.csv(DEGs_M, "~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnal  
  
venn1<- venn.diagram(List("Female"=DEGs_F$gene_name, "Male"=DEGs_M$gene_name),filename = NULL)  
grid.newpage()  
  
grid.draw(venn1)
```



```

pdf(file="~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/Fema
venn2<-venn.diagram(List("Female"=DEGs_F_relax_p$gene_name, "Male"=DEGs_M$gene_name),  filename = NULL)
grid.newpage()
grid.draw(venn2)
pdf(file="~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/Fema
dev.off()

## pdf
## 2

```

Results

This section shows the results of the DEG and Go/KEGG analysis

DEG results

Kept this just in case but it is not being used since we are doing pathway analysis with Reactome

```

# =====
# Enriched pathways males vs. females
# =====

#path <- "~/3.0 Hasting Research/Liver Cancer Analysis/DEGs/DEGs_newcounts_Salmon_batchMD1cov_fpkm05_HB

```

```

#Do not have appropriate files using old files for this

#degResult_genes <- DEGs_M

#degResult_genes$hgnc_symbol <- degResult_genes$name
#ensembl <- useEnsembl(biomart="ensembl", dataset="hsapiens_gene_ensembl")

#biomart <- getBM(attributes=c('ensembl_gene_id', 'hgnc_symbol', 'entrezgene_id', 'kegg_enzyme'), filter="hgnc_symbol")

#Deleted "kegg_enzyme" label because it does not work.

#biomart <- getBM(attributes=c('ensembl_gene_id', 'hgnc_symbol', 'entrezgene_id'), filters = 'hgnc_symbol')
#colnames(degResult_genes) <- c("chr", "start", "end", "TXNAME", "GENEID", "hgnc_symbol", "length", "logFC")

```

GO/KEGG Analysis Female tumor vs. tumor-adjacent

```

#degResult <- DEGs_F
#degResult$hgnc_symbol <- degResult$gene_name
#degResult_biomart <- merge(degResult, biomart, by="hgnc_symbol")
#degResult_biomart <- degResult_biomart[complete.cases(degResult_biomart), ]
#geneList <- degResult_biomart$entrezgene_id
#geneList_quant <- degResult_biomart$logFC
#ego <- enrichGO(geneList, OrgDb = "org.Hs.eg.db", ont="BP", readable=TRUE)
#pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumor_vs_Tumor-adjacent/GO_BP.pdf")
#goplot(ego)
#dev.off()
#pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumor_vs_Tumor-adjacent/GO_BP_dotplot.pdf")
#dotplot(ego, showCategory=30)
#dev.off()

```

```

#degResult <- DEGs_F_relax_p
#degResult$hgnc_symbol <- degResult$gene_name
#degResult_biomart <- merge(degResult, biomart, by="hgnc_symbol")
#degResult_biomart <- degResult_biomart[complete.cases(degResult_biomart), ]
#geneList <- degResult_biomart$entrezgene_id
#geneList_quant <- degResult_biomart$logFC
#ego <- enrichGO(geneList, OrgDb = "org.Hs.eg.db", ont="BP", readable=TRUE)
#pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumor_vs_Tumor-adjacent/GO_BP_relaxed.pdf")
#goplot(ego)
#dev.off()
#pdf("~/Desktop/ResearchProjects/LiverCancer/DEA_removed_samples/DifferentialExpressionAnalysis/FemaleTumor_vs_Tumor-adjacent/GO_BP_dotplot_relaxed.pdf")
#dotplot(ego, showCategory=30)

```