Differential Gene Expression Between Ectopic and Eutopic Endometrium Highlights Cancer-Linked Pathways in Endometriomas

M. Gage

2025-05-18

Document Description

This document includes the applicable code, plots, and written assessments for differential expression analysis. The samples selected include a cohort of 10 ectopic ovarian endometrioma biopsies (Stage III-IV; ENDO group) and 10 eutopic endometrial tissue biopsies (CTL group). All samples were collected at Tartu University Hospital, Estonia, and disease states were confirmed by a pathologist according to ASRM guidelines. Criteria for exclusion included infections, endocrine or metabolic disorders, anatomic abnormalities, autoimmune diseases, and carcinomas. Women included in the study were not taking hormonal medications within three months of biopsy collection. Sample reads (FASTA files) were QC'd, trimmed, aligned to a human reference genome, and used to generate counts in Galaxy before differential expression analysis.

Loading in Data/Merging Counts Files

```r
# Set the working directory via setwd() or via Session->Set Working Directory->Choose Directory

# load libraries
library(DESeq2)
library(pheatmap)
library(EnhancedVolcano)
library(org.Hs.eg.db)

# List all .tabular files
files <- list.files(pattern = "\\.tabular$")

# Read and name each file
count_list <- lapply(files, function(file) {
  sample_name <- tools::file_path_sans_ext(file)
  read.table(file, sep="\t", skip=0, nrows=78932, quote="\"", header=FALSE,
             col.names = c("Gene_ID", sample_name))
})

# Merge all data frames by "Gene_ID"
merged_counts <- Reduce(function(x, y) merge(x, y, by="Gene_ID"), count_list)

# Reformat the data so that the rownames are Ensembl Gene IDs
```

```
rnaseqMatrix <- merged_counts[,c(2:21)]
rownames(rnaseqMatrix) <- merged_counts[,1]
head(rnaseqMatrix)

last_two <- substr(colnames(rnaseqMatrix), nchar(colnames(rnaseqMatrix))-1,
                   nchar(colnames(rnaseqMatrix)))
colnames(rnaseqMatrix)[1:10] <- paste0("ENDO_", last_two[1:10])
colnames(rnaseqMatrix)[11:20] <- paste0("CTL_", last_two[11:20])

head(rnaseqMatrix)
```

DESeq2 Data Preparation

```
#Extract sample group from the column names.
groups <- sub("_.*", "", colnames(rnaseqMatrix))

# Create colData (samples metadata)
samples <- data.frame(ID = colnames(rnaseqMatrix),
                      Group = factor(groups))

# Set rownames to match colnames of count matrix
rownames(samples) <- samples$ID

# Create the DEseq2DataSet object
deseq2Data <- DESeqDataSetFromMatrix(countData = rnaseqMatrix,
                                     colData = samples,
                                     design = ~ Group)

# Determine how many genes were lowly expressed and decide on a threshold
dim(deseq2Data)
dim(deseq2Data[rowSums(counts(deseq2Data)) > 10, ])

# Filter out lowly expressed genes
deseq2Data <- deseq2Data[rowSums(counts(deseq2Data)) > 10, ]
```

DESeq2 Diagnostic Analysis

```
pdf("pairwise_scatter_ENDO.pdf")
pairs(log2(counts(deseq2Data)+1), pch = '.', xlim=c(0,18), ylim=c(0,18))
dev.off()

# Run pipeline for differential expression steps
deseq2Data <- DESeq(deseq2Data)

# rlog transform counts (can use rlog instead of vst)
rld <- vst(deseq2Data, blind=FALSE)
rlogcounts <- data.frame(assay(deseq2Data))
rownames(rlogcounts) <- rownames(deseq2Data)
```

```
# PCA plot
pdf("ENDO_CTRL_PCA_plot.pdf")
plotPCA(rld, intgroup=c("Group"))
dev.off()

# PCA plot data
plotPCA(rld, intgroup=c("Group"), returnData=TRUE)

# Inspect rlog data using heatmap of pairwise correlation
# (THIS CAN TAKE A LONG TIME TO RUN)
pheatmap(cor(assay(rld)),cluster_cols = FALSE, cluster_rows = FALSE, fontsize = 15)
```

Developing the DEG List and Associated Statistics

```
# Pairwise contrast (developing DEG list).
res_ENDO_CTL <- results(deseq2Data, contrast=c("Group", "ENDO", "CTL"), alpha = 0.05)
#alpha is FDR threshold
resOrdered_ENDO_CTL <- res_ENDO_CTL[order(res_ENDO_CTL$pvalue),]

# Get number of differentially expressed data at different thresholds
summary(res_ENDO_CTL)
sum(res_ENDO_CTL$padj < 0.05, na.rm=TRUE) #FDR threshold
sum(res_ENDO_CTL$pvalue < 0.05, na.rm=TRUE)

# MA plot
pdf("ENDO_CTL_MA_plot.pdf")
plotMA(res_ENDO_CTL)
dev.off()

# Save the results as a data frame
results <- data.frame(resOrdered_ENDO_CTL)
head(results)

# Reading in the annotation file
annotation <- read.table("human_genome_annot.txt",header=TRUE,sep="\t",quote="\"")

# Annotate results
results.annot <- merge(results,annotation,by.x=0,by.y=1)
results.annot <- results.annot[order(results.annot$padj),]
names(results.annot) <- c("GeneID","baseMean","log2FoldChange","lfcSE","stat","pvalue",
                          "padj","Symbol","Description","Biotype","Chr","Begin","End",
                          "Strand")

head(results.annot)

write.table(results.annot,"annot_gene_list_ENDO_CTL.txt",sep="\t",row.names=FALSE)
```

Plots

```r
#*******************************************
# Volcano Plot
#*******************************************

# You will have to edit labels and file name for the treatment group you are working with

ens <- rownames(deseq2Data)
symbols <- mapIds(org.Hs.eg.db, keys = ens,
                  column = c('SYMBOL'), keytype = 'ENSEMBL')
symbols <- symbols[!is.na(symbols)]
symbols <- symbols[match(rownames(deseq2Data), names(symbols))]
rownames(deseq2Data) <- symbols
keep <- !is.na(rownames(deseq2Data))
deseq2Data <- deseq2Data[keep,]

res <- results(deseq2Data,
               contrast = c("Group", "ENDO", "CTL"))

res <- lfcShrink(deseq2Data,
                 contrast = c("Group", "ENDO", "CTL"), res=res, type = 'normal')

pdf("ENDO_CTL_Volcano_plot.pdf")
EnhancedVolcano(res,
                lab = rownames(res),
                x = 'log2FoldChange',
                y = 'pvalue',
                pCutoff = 0.0000000000000000001,
                drawConnectors = TRUE,
                widthConnectors = 0.5,
                FCcutoff = 4.0,
                pointSize = 2.0,
                labSize = 3.0,
                title = 'ENDO vs CTL Volcano Plot')



#*******************************************
# Heatmap
#*******************************************


# Candidate genes (you can select any genes you want on BioMart.)
# The current selected genes are MT1X, MT1F, MT1G, MT1E, and MT1M)
candidates <- read.delim("Major_Genes_of_Interest_ENDO.txt",header=TRUE)
names(candidates) <- c("ID","Symbol", "Biotype")

# Subset results with data just for candidate genes
table_hits <- rlogcounts[candidates$ID,]

# Replace any NA's with 0
table_hits[is.na(table_hits)] <- 0
```

```r
# Replace Ensembl Gene IDs with symbols
row.names(table_hits) <- candidates$Symbol

cal_z_score <- function(x){
  (x - mean(x)) / sd(x)
}

table_hits_norm <- t(apply(table_hits, 1, cal_z_score))

pheatmap(table_hits_norm,name = "Row Z-Score",cluster_cols = FALSE, fontsize = 12,
         main = "Genes Mapped to KEGG 'Cell Cycle' Pathway")
```

```r
pdf("heatmap_Mapped_Genes_KEGGCellCyclePathway.pdf")
pheatmap(table_hits_norm,name = "Row Z-Score",cluster_cols = FALSE, fontsize = 8,
         main = "Genes Mapped to KEGG 'Cell Cycle' Pathway")
```

```r
dev.off()




#********************************************
# Boxplot
#********************************************


library(ggplot2)
library(reshape2)

# Extract rlog-transformed counts for candidate genes
mt_genes <- candidates$ID
mt_rlog <- assay(rld)[mt_genes, ]
rownames(mt_rlog) <- candidates$Symbol

# Convert to long format for ggplot2
mt_df <- as.data.frame(t(mt_rlog))
mt_df$SampleID <- rownames(mt_df)
mt_df$Group <- samples[mt_df$SampleID, "Group"]

mt_long <- melt(mt_df, id.vars = c("SampleID", "Group"), variable.name = "Gene",
                value.name = "Expression")

# Plot
ggplot(mt_long, aes(x = Group, y = Expression, fill = Group)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  geom_jitter(width = 0.2, size = 1.2, alpha = 0.5) +
  facet_wrap(~ Gene, scales = "free_y") +
  theme_minimal(base_size = 14) +
  scale_fill_manual(
    values = c("CTL" = "skyblue", "ENDO" = "tomato"),
    name = "Group"  # Legend title
  ) +
  labs(
    x = NULL,
    y = "rlog Expression"
  ) +
  theme(
    axis.text.x = element_blank(),     # Remove axis labels
    axis.ticks.x = element_blank(),    # Remove tick marks
    strip.text = element_text(size = 12, face = "bold"),
    legend.position = "right"          # Show legend on the right
  )
```