Identifying DEGs in practice

Practical for FOS course Molecular Data Science: from disease mechanisms to personalized medicine, Tuesday October 23, 2018

Rodrigo Coutinho de Almeida and Yolande Ramos (Molecular Epidemiology)

In this practical you will analyze a microarray dataset to identify differentially expressed genes (DEGs) in arthritic cartilage. For this purpose, gene expression profiles of articular cartilage samples were generated with Illumina HiSeq2000 and HiSeq4000. Cartilage samples were collected from patients undergoing total joint replacement surgery due to primary hip or knee osteoarthritis (OA), both, from macroscopically unaffected cartilage regions (**Preserved**) and from affected cartilage regions (**OA**).
Analyses will be performed using **R**.

First we need to load the appropriate analysis package. For differential gene expression analysis and normalization we will use the **DESeq2 package**. Start **R** as administrator and copy/paste the following commands to make sure both packages are installed and loaded:

```
# load libraries
> library(DESeq2)
> library(biomaRt)
> library(pheatmap)
```

Next, we need to load the actual data:

```
> setwd("c:/bioinfo2018/")

> load(file = "data/RNASeq_PairedOA.Rdata")
> load(url("https://raw.githubusercontent.com/molepi/Molecular-Data-
Science/master/ExpressionData/RNASeq_PairedOA.Rdata")) # Load the expression data
```

This will load the expression matrix of the OA paired samples (PairsOA) and the phenotype data (pheno)
Inspect the loaded data by running the following commands:

```
> head(pheno)                    # show phenotype data
```

Question 1.

a) What information is provided?

b) How many samples are in the data set? How many unique individuals?

   R tips: `> nrow(pheno)`
   R tips: `> length(unique(pheno$Patient))`

c) How many Lesioned and how many preserved OA?

   R tips: `> table(pheno$Status)`

Question 2.
The next steps consist of data and normalization.

a) Plot the non-normalized with the commands bellow and explain why is important to normalize your RNA-seq data?:

```
> boxplot(PairsOA_all)
> hist(as.matrix(PairsOA_all))
```

b) Now normalize the data and plot. Compare with the previous plot and explain what happened to your dataset. What is considered to normalize RNA-seq data?

```
> rldGenes <- rlogTransformation(ddsGenes, fitType='local', bind = F)
MatrldGenes <- assay(rldGenes)

hist(rldGenes, main = "Distribution of genes in OA cartilage", col = "salmon",
xlab = "Log2 Normalized DESEQ2", prob = F)

hist(MatrldGenes, main = "Distribution of genes in OA cartilage", col = "salmon",
     xlab = "Log2 Normalized DESEQ2", prob = T, ylim = c(0,0.45))
lines(density(rldGenes),lwd = 2, col = "blue")

boxplot(MatrldGenes) #plot normalize data
```

c) Inspect you data using Principle Component Analysis (PCA), run the following command and report what did you see.

```
> plotPCA(rldGenes, intgroup=c("Batch"))
```

d) In case you identified a batch effect, run the following command and plot.

```
> NoBatch <- limma::removeBatchEffect(assay(rldGenes), rldGenes$Batch)

> pc <- prcomp(t(NoBatch))

percenVar <- pc$sdev^2/sum(pcv$sdev^2)

plotData <- data.frame(pheno, pc$x[,1:3])

p <- ggplot(plotData, aes(x = PC1, y = PC2, colour = Batch))

p + geom_point(size = 3) +
  scale_color_manual(values = c("salmon", "blue"))+
  xlab(paste0("PC1: ", round(percenVar[1] * 100), "% variance")) +
  ylab(paste0("PC2: ", round(percenVar[2] * 100), "% variance")) +
  ggtitle("PCA batch effect removed") +
  theme(plot.title = element_text(family = "Helvetica", color="#666666",
face="bold", size=16, hjust=0.5))
```

Now that we have normalized the dataset we are ready to perform the actual differential expression analysis. We're particularly interested in generic gene expression differences between pairs of preserved and OA lesioned cartilage. To analyse this we will perform a pairwise test analysis.

Which is done by the following commands:

```
> ddsGenes <- DESeqDataSetFromMatrix(countData=PairsOA_all,
                                    colData=pheno,
                                    design = ~Patient + Status) #Here you build the
model take the pairs of patients into account
ddsGenes <- DESeq(ddsGenes)
```

```
resGenes <- results(ddsGenes)
resGenes <- results(ddsGenes, contrast=c("Status", "Lesion", "Preserved")) #Fold
change will be set for Lesion
resGenes <- as.data.frame(resGenes)
resGenes <- resGenes[complete.cases(resGenes),] #take out NA
resGenes <- resGenes[order(resGenes$padj),] #Order according  with the P-values
(FDR)
resGenes$FC <- 2^resGenes$log2FoldChange #Include an extra column with the fold
changes

#Subset the signifcant genes
sigGen <- resGenes[resGenes$padj < 0.05,]

#Change Transcripts ID to Genes Symbols
ensembl=useMart("ensembl")
ensembl = useDataset("hsapiens_gene_ensembl",mart=ensembl)
attributes <- listAttributes(ensembl)
transcriptsids= as.character(rownames(sigGen))

geneIDs <- getBM(attributes = c('ensembl_gene_id', 'external_gene_name'),
                 filters = 'ensembl_gene_id',
                 values = transcriptsids,
                 mart = ensembl)
#Merge and preprocess the data
SignGeneNames <- merge(sigGen,geneIDs,by.x = "row.names",by.y = ,"ensembl_gene_id")
colnames(SignGeneNames)
SignGeneNames <- SignGeneNames[,c(1,9,8,3,6,7)]
colnames(SignGeneNames)[1:2] <- c("Emsembl_ID", "GeneNames")
SignGeneNames <- SignGeneNames[order(SignGeneNames$padj),]
```

Question 3. Inspect the output:

a) Run `head(resGenes, n)` to inspect the top *n* genes in your differential expression analysis. What strikes you about the adjusted p-values? How do you think this could have happened?

b) What is the range of differential expression?

R tips: `> range(resGenes$FC)`

c) What is the most downregulated gene? What is the most upregulated gene? Are they significant?

R tips: `> result[which(resGenes$FC %in% range(resGenes$FC)), ]`

d) Do you think these changes are biologically relevant?

e) How many significantly different genes have you identified?

f) How many remain if you only consider those with a Fold-change of at least 2?

R tips:

`> signGenesFC <- SignGeneNames[SignGeneNames$FC >2 | SignGeneNames$FC < 0.5,]`

g) Are the top differentially expressed genes known to be involved in OA? If so, does the direction of differential expression support the reported role in OA?

h) How about genes not known to be involved in OA, does their in- or decrease in expression make sense in relation to OA?

Question 4.

To add biological context to your results you can perform a pathway enrichment analysis. Using the following command you will check how your differentially expressed genes are enriched in one branch of the Biological Process from the Gene Ontology database (GO).

```
>  library(enrichR) #Load the library enrichR

dbs <- c("GO_Biological_Process_2017") #Subset a database for the exercise

enriched <- enrichr(SignGeneNames$GeneNames, dbs) #Perform the enrichment analysis

head(enriched[["GO_Biological_Process_2017"]]) #Check the results

#Here you will preprocess the results to visualize
GOBioProc <- as.data.frame(enriched$"GO_Biological_Process_2017")
GOBioProc$log.Adjpvalue <- -log10(as.numeric(GOBioProc$Adjusted.P.value))
GOBioProc$log.pvalue <- -log10(as.numeric(GOBioProc$P.value))
GOBioProc$Term <- factor(GOBioProc$Term, levels =
GOBioProc$Term[order(GOBioProc$P.value, decreasing = T)])

#Subset only pathways significant after correction for multiple testing
GOBioProc_FDR <- GOBioProc[GOBioProc$Adjusted.P.value <0.05,]

#Plot the top 10 pathways
ggplot(data=GOBioProc_FDR[1:10,], aes(x=Term, y=log.pvalue)) +
  geom_bar(stat="identity", position="identity", width = 0.5) + coord_flip() +
  theme(axis.text.x = element_text(size = 10, angle = 45,hjust = 1, vjust = 1)) +
  labs( x="GO Biological process", y="-log10(Adj P-value)")
```

a) How many pathways are significant after correction for multiple testing?

b) Which pathway is the most significant? Is this a known OA pathway?

c) How many genes is overlapping in the most significant pathway?