

A

```
###Filter out genes with negative ComBat normalized counts an round to integers###
combat_cts <- combat_cts[-which(combat_cts < 0, arr.ind=TRUE)[,1],]
```

B

```
combat_countData <- round(as.matrix(combat_cts,sep="\t"))
```

```
###Define your experimental design for DESeq2###
```

```
combat_totDesign = data.frame(
  row.names = colnames( combat_countData ),
  condition = c( "iluc", "iluc", "iluc", "ief", "ief", "ief", "wtef", "wtef", "wtef",
                 "del122", "del122", "del122", "daf", "daf", "daf" ),
  batch = c( "A", "B", "C", "A", "B", "C", "A", "B", "C", "A", "B", "C", "A", "B", "C" ) )
```

```
combat_totDesign
```

```
###Create a DESeq DataSet with count data and design###
```

```
combat_dds <- DESeqDataSetFromMatrix( countData = combat_countData,
                                     colData = combat_totDesign,
                                     design = ~ batch + condition)
```

```
###Estimate size factors and run DESeq2###
```

```
combat_dds <- estimateSizeFactors(combat_dds)
combat_dds.analysis <- DESeq(combat_dds)
```

```
###Extract rlog-normalized counts###
```

```
combat_dds.rlog <- rlog(combat_dds, blind = FALSE)
head( assay( combat_dds.rlog ), 5 )
```

C

```
###Extract pairwise comparison results###
```

```
results.1 <- results(combat_dds.analysis, contrast = c("condition", "ief", "iluc"))
summary(results.1)
results.2 <- results(combat_dds.analysis, contrast = c("condition", "ief", "wtef"))
summary(results.2)
results.3 <- results(combat_dds.analysis, contrast = c("condition", "ief", "del122"))
summary(results.3)
results.4 <- results(combat_dds.analysis, contrast = c("condition", "ief", "daf"))
summary(results.4)
```

D

```
###Add gene symbols to data###
```

```
results.1$symbol <- mapIds(org.Hs.eg.db,
                          keys=row.names(results.1),
                          column="SYMBOL",
                          keytype="ENSEMBL",
                          multiVals="first")
```

```
head(results.1)
```

```
results.1.DF <- as.data.frame(results.1)
```

E

```
###Export differential expression data###
```

```
de_data <- data.frame(ENSID = rownames(results.1.DF), baseMean = results.1.DF$baseMean, symbol = results.1.DF$symbol,
                    iluc_log2FC = results.1.DF$log2FoldChange, iluc_pvalue = results.1.DF$pvalue, iluc_padj = results.1.DF$padj,
                    wtef_log2FC = results.2.DF$log2FoldChange, wtef_pvalue = results.2.DF$pvalue, wtef_padj = results.2.DF$padj,
                    del122_log2FC = results.3.DF$log2FoldChange, del122_pvalue = results.3.DF$pvalue, del122_padj = results.3.DF$padj,
                    daf_log2FC = results.4.DF$log2FoldChange, daf_pvalue = results.4.DF$pvalue, daf_padj = results.4.DF$padj)
```

```
write.table(de_data, "differential_expression.txt", sep = "\t", quote = FALSE, row.names = FALSE)
```