# Salmon_DPYSL2B

*dbennett*

*11 May 2017*

---

## Background

Dihydropyrimidinase Like 2 (*DPYSL2*) encodes CRMP2, a member of the collapsin response mediator family located on the petite arm of chromosome 8. DPYSL2 plays a key role in microtubule assembly as well as synaptic signalling. *DPYSL2* has been associated with neurological disorders such as alzheimers disease and schizophrenia [1]. Chr 8p has been shown to be a schizophrenia susceptibility loci as far back as 1995 [2]. In 2009, *Holmans et al*, performed a large linkage study which found that the previously held assumption that *NRG1* portrayed schizophrenia risk at this loci [3]. *Fallin*, in 2005, furthered this work and ascertained that *DPYSL2* was also a schizophrenia candidate gene in this region [4]. Three transcripts are encoded by *DPYSL2*; A,B, and C.

The aim of this research is to identify the link between schizophrenia and *DPYSL2B* via mTOR signaling established by *liu et al* (same reasearch group) *liu* found that 13 dinucleotide repeats (DNR) located in the 5' UTR of *DPYSL2B* was the most common high risk genotype compared to 11 DNR for low risk [5].

To verify their hypothesis, *pham et al* created two HEK cell lines one containing 11 DNRs the other containing 13 DNRs To identify what proteins interacted with the modified and and unmodified isoforms a protein microarray analysis was carried out.(4000 proteins) the results indicated that the unmodified 11DNR samples bound to mTOR effector proteins and HuD exclusively [6].

Western blot analysis showed that the 13 DNR cells produced siginificantly less CRMP2B protein than the 11 DNR genotype when compared to CRMP2A (P6.5x10-6 & P=0.0077). *Pham et al* then looked at the effect of Rapamycin on protein levels. Translation of CRMP2B was found to be reduced for both genotypes however reduction was again most significant in the 13 DNR cells. Pham et al interprets these results as such, 13 DNR are weak mTOR responders and therefore are more sensitive to its inhibition than the wildtype 11 DNR cells. A criticism is of this is paper is the time spent discussing the RNA-seq results compared to the time spent comparing it to work previously done. To be critical of this paper means to be critical of its predecessors.

The aim of this analysis is to reproduce the results obtained by *Pham et al* using Salmon a tool which quantifies RNA transcripts in a fraction of the time compared to conventional methods such as the tuxedo suite pipeline [7]. The first step Salmon indexes a reference transcriptome fasta file which allows for quasi mapping of fastq files the alignment model uses a varying first order markv model over a set of CIGAR symbols and nucleotides. The second stage is an online estimation of model parameters and initial expression levels using bayesian inference. The final step or off-line stage in the analysis is either a standard EM algorithm or a variational Bayesian EM algorithm. The real advantage of Salmon is the time it takes from reads to differentially expressed gene set. This analysis took less than 5hr to complete due to the fact that a bootstraping parameter set to 100 was included. Without out this step however the full analysis is less than 1 hr. To find differentially expressed genes DESeq2 was used on the salmon TPM files. Pathway analysis was carried out using goseq on the significant differentially expressed genes.

# *Code for replication*

## Aspera

Aspera code to retrieve the sra files from the SRA database

```
#!/bin/sh
#$ -N DB_fq_dump
#$ -q all.q
#$ -cwd
#$ -S /bin/bash
#$ -v PATH
#$ -v LD_LIBRARY_PATH
#$ -v PYTHONPATH
for i in 16 18 20 22 24 26 28 30 31 33 35 37; do
/home/nextgen2015/.aspera/connect/bin/ascp -i /home/nextgen2015/.aspera/connect/etc/\
asperaweb_id_dsa.openssh -k1 -Tr -l200m anonftp@ftp-trace.ncbi.nlm.nih.gov:/sra/sra-\
instant/reads/ByStudy/sra/SRP/SRP076/SRP076104/SRR26240${i}/SRR26240${i}.sra ./
done
```

---

Fastq dump will extract the fastq files from sra format. Fastqc showed that all files contained high quality trimmed sequences. No further trimming ws required.

```
#!/bin/sh
#$ -N DB_sra_extract
#$ -q all.q
#$ -cwd
#$ -S /bin/bash
#$ -v PATH
#$ -v LD_LIBRARY_PATH
#$ -v PYTHONPATH

for s in *.sra; do fastq-dump -I --split-files $s; done
```

---

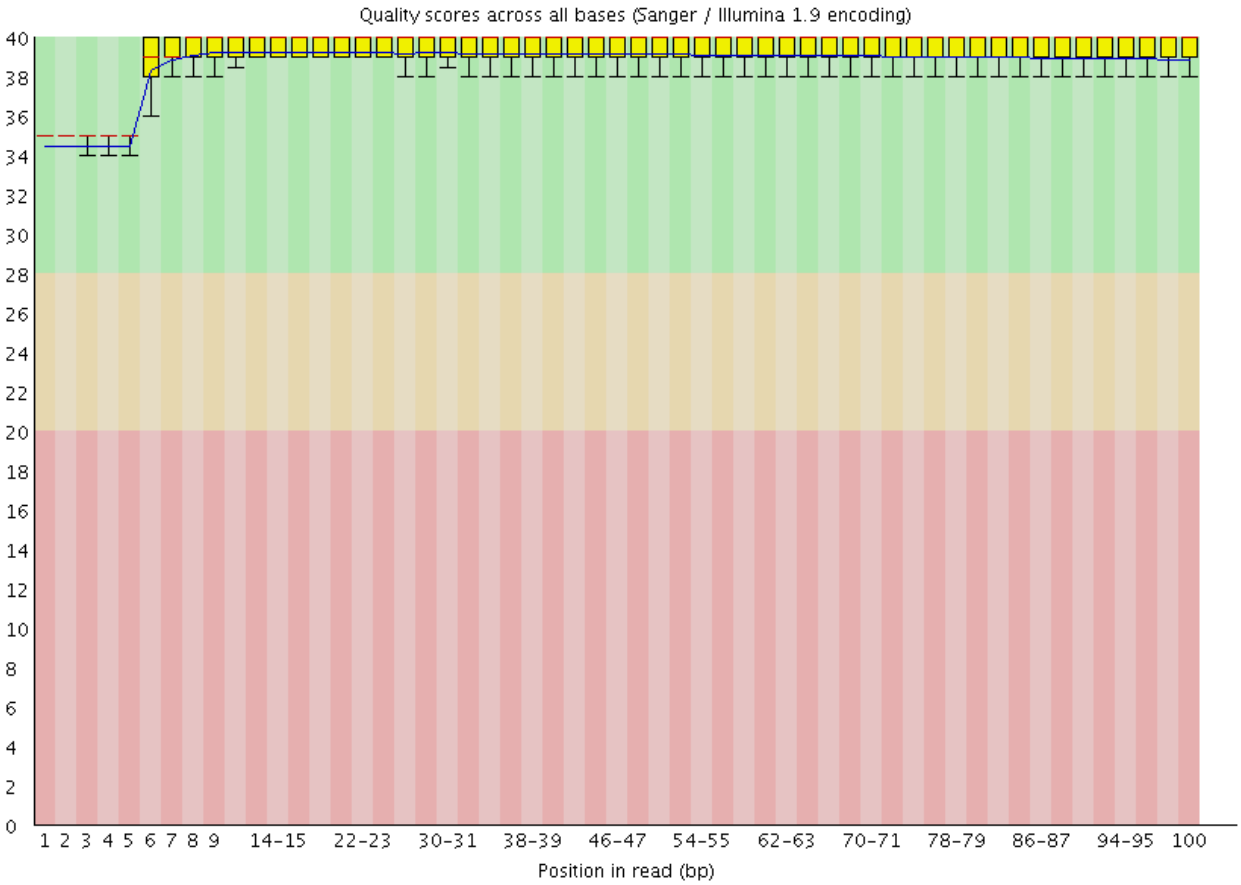
rename files as sample names

```
#!/bin/sh
#$ -N DB_rename_sleep
#$ -q all.q
#$ -cwd
#$ -S /bin/bash
#$ -v PATH
#$ -v LD_LIBRARY_PATH
#$ -v PYTHONPATH

# made in text editor not ideal and definitely a better more \
# general way
mv SRR3624016_1.fastq mod_0_r1.fq
mv SRR3624018_1.fastq mod_1_r1.fq
mv SRR3624020_1.fastq mod_2_r1.fq
mv SRR3624022_1.fastq mod_3_r1.fq
mv SRR3624024_1.fastq unmod_0_r1.fq
```

```
mv SRR3624026_1.fastq unmod_1_r1.fq
mv SRR3624028_1.fastq unmod_2_r1.fq
mv SRR3624030_1.fastq unmod_3_r1.fq
mv SRR3624031_1.fastq unmod_4_r1.fq
mv SRR3624033_1.fastq unmod_5_r1.fq
mv SRR3624035_1.fastq unmod_6_r1.fq
mv SRR3624037_1.fastq unmod_7_r1.fq
mv SRR3624016_2.fastq mod_0_r2.fq
mv SRR3624018_2.fastq mod_1_r2.fq
mv SRR3624020_2.fastq mod_2_r2.fq
mv SRR3624022_2.fastq mod_3_r2.fq
mv SRR3624024_2.fastq unmod_0_r2.fq
mv SRR3624026_2.fastq unmod_1_r2.fq
mv SRR3624028_2.fastq unmod_2_r2.fq
mv SRR3624030_2.fastq unmod_3_r2.fq
mv SRR3624031_2.fastq unmod_4_r2.fq
mv SRR3624033_2.fastq unmod_5_r2.fq
mv SRR3624035_2.fastq unmod_6_r2.fq
mv SRR3624037_2.fastq unmod_7_r2.fq
```

**Fastqc**

All fastq files where checked for sequencing quality using fastqc every fastq file contained high quality reads. Below is an example of the results:




## Salmon

We will need the transcript fasta file these are taken from ensembl ftp server. Next we will begin the quasi mapping stage of the salmon analysis by indexing our transcript fasta file before quasi mapping our fastq files.

```
#!/bin/sh
#  job name after -N
#$ -N DB_salmon_run
# The job should be placed into the queue 'all.q'
#$ -q all.q
# Running in the current directory
#$ -cwd
#$ -m bea
#$ -M d.bennett1@nuigalway.ie
# Export some necessary environment variables
#$ -S /bin/bash
#$ -v PATH
#$ -v LD_LIBRARY_PATH
#$ -v PYTHONPATH
# Download non-coding and coding fasta files
wget ftp://ftp.ensembl.org/pub/release-75/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh37.75\
.cdna.all.fa.gz
wget ftp://ftp.ensembl.org/pub/release-75/fasta/homo_sapiens/ncrna/Homo_sapiens.GRCh37.75\
.ncrna.fa.gz
```

```
## merge together

gunzip -c Homo_sapiens.GRCh37.75.cdna.all.fa.gz Homo_sapiens.GRCh37.75.ncrna.fa.gz > \
Homo_sapiens.GRCh37.75.cdna.ncrna.fa

# Create index file
salmon index -t Homo_sapiens.GRCh37.75.cdna.ncrna.fa -i Homo_sapiens.GRCh37.75_quasi_index


# salmon quantification step ##stupid folder output
for i in /data4/Declan/data_analysis/SRA/*r1.fq;
do
    base=$(echo "$i" | rev | cut -c 6- | rev) # awkward way to remove the r1.fq so we can use\
    #loop for the paired end data
    salmon quant --numBootstraps 100 -p 12 -i Homo_sapiens.GRCh37.75_quasi_index -l IU -1 \
    ${base}r1.fq -2 ${base}r2.fq -o salmon/${base}quant
done
```

## DESeq2

We will need some libraries for DESeq2 to work.

```
library(tximport)
library(DESeq2)
library(tximportData)
library(ggplot2)
library(gplots)
library(pheatmap)
library(RColorBrewer)
library(vsn)
library(EnsDb.Hsapiens.v75)
library(org.Hs.eg.db)
library(grid)
library(Biobase)
library(goseq)
```

Salmon works at the transcipt level so we will need to map the ensembl transcript ids to their gene id counterparts. We also need a pointer file for each of our samples. samples.txt contains a folder and file name path

```
edb <- EnsDb.Hsapiens.v75  # database

# make transcripts to gene file
Tx.ensembl <- transcripts(edb, columns = c("tx_id", "gene_id", "gene_name"),
    return.type = "DataFrame")
nrow(Tx.ensembl)
```

```
## [1] 215647
```

```
tx2gene <- Tx.ensembl[, c(1, 2)]
head(tx2gene)
```

```
## DataFrame with 6 rows and 2 columns
##             tx_id         gene_id
```

```
##       <character>     <character>
## 1 ENST00000000233 ENSG00000004059
## 2 ENST00000000412 ENSG00000003056
## 3 ENST00000000442 ENSG00000173153
## 4 ENST00000001008 ENSG00000004478
## 5 ENST00000001146 ENSG00000003137
## 6 ENST00000002125 ENSG00000003509
```

```
# set directory
dir = "/home/user09/"

# load in pointer file
samples = read.table(file.path(dir, "samples.txt"))
# add conditions
samples$condition = factor(c(rep("modified", 4), rep("unmodified", 8)))
# dress up samples.txt
rownames(samples) = samples$V1
colnames(samples) = c("Name", "condition")
files = file.path(dir, "salmon", samples$Name, "quant.sf")
names(files) = samples$Name
# load in quant files not sure which is best
txi = tximport(files, type = "salmon", tx2gene = tx2gene)

# for the bootstrapped values takes too long txi.inf.rep <- tximport(files,
# type = 'salmon', txOut = TRUE)

# DESeq2 analysis set up
DESeq <- data.frame(condition = factor(c(rep("_modified", 4), rep("_unmodified",
    8))))
rownames(DESeq) <- colnames(txi$counts)
# table should be good to go
dds <- DESeqDataSetFromTximport(txi, DESeq, ~condition)

dds <- DESeq(dds)
res <- results(dds)
resultsNames(dds)
```

```
## [1] "Intercept"           "condition_modified"   "condition_unmodified"
```

```
# take the significant results
res05 <- results(dds, alpha = 0.05)
summary(res05)
```

```
##
## out of 37625 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)     : 1675, 4.5%
## LFC < 0 (down)   : 1296, 3.4%
## outliers [1]     : 59, 0.16%
## low counts [2]   : 19656, 52%
## (mean count < 5)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
sum(res05$padj < 0.05, na.rm = TRUE)
```

```
## [1] 2971
```

```
resOrdered <- res[order(res$padj), ]  # order results total
res <- subset(resOrdered, padj < 0.1)  # take just results under 0.1

# check the column names. important for next line
columns(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"
##  [5] "ENSEMBLTRANS" "ENTREZID"     "ENZYME"       "EVIDENCE"
##  [9] "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"
## [13] "IPI"          "MAP"          "OMIM"         "ONTOLOGY"
## [17] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [25] "UNIGENE"      "UNIPROT"
```

```
# Add some extrainformation important for goseq
res$symbol <- mapIds(org.Hs.eg.db, keys = row.names(res), column = "SYMBOL",
    keytype = "ENSEMBL", multiVals = "first")
res$entrez <- mapIds(org.Hs.eg.db, keys = row.names(res), column = "ENTREZID",
    keytype = "ENSEMBL", multiVals = "first")
res$name <- mapIds(org.Hs.eg.db, keys = row.names(res), column = "GENENAME",
    keytype = "ENSEMBL", multiVals = "first")

# significant
head(res)
```

```
## log2 fold change (MAP): condition _unmodified vs _modified
## Wald test p-value: condition _unmodified vs _modified
## DataFrame with 6 rows and 9 columns
##                   baseMean log2FoldChange      lfcSE      stat
##                  <numeric>      <numeric>  <numeric> <numeric>
## ENSG00000155090  679.23445      1.6364273 0.15568280 10.511291
## ENSG00000050555  357.91965      1.4621801 0.14286162 10.234940
## ENSG00000130741 6035.32192      0.4545139 0.05053621  8.993828
## ENSG00000271880   89.03494      2.5135449 0.28275934  8.889343
## ENSG00000070669 4525.31699      1.8675259 0.21598425  8.646584
## ENSG00000100867  142.08271      2.0035744 0.23827193  8.408772
##                       pvalue         padj      symbol      entrez
##                    <numeric>    <numeric> <character> <character>
## ENSG00000155090 7.663679e-26 1.373638e-21       KLF10        7071
## ENSG00000050555 1.382809e-24 1.239273e-20       LAMC3       10319
## ENSG00000130741 2.387657e-19 1.426546e-15      EIF2S3        1968
## ENSG00000271880 6.147110e-19 2.754520e-15          NA          NA
## ENSG00000070669 5.306408e-18 1.902241e-14        ASNS         440
## ENSG00000100867 4.143258e-17 1.237729e-13       DHRS2       10202
##                                                                 name
##                                                          <character>
## ENSG00000155090                                 Kruppel like factor 10
## ENSG00000050555                                laminin subunit gamma 3
## ENSG00000130741 eukaryotic translation initiation factor 2 subunit gamma
## ENSG00000271880                                                   NA
## ENSG00000070669           asparagine synthetase (glutamine-hydrolyzing)
```

```
## ENSG00000100867                          dehydrogenase/reductase 2
```

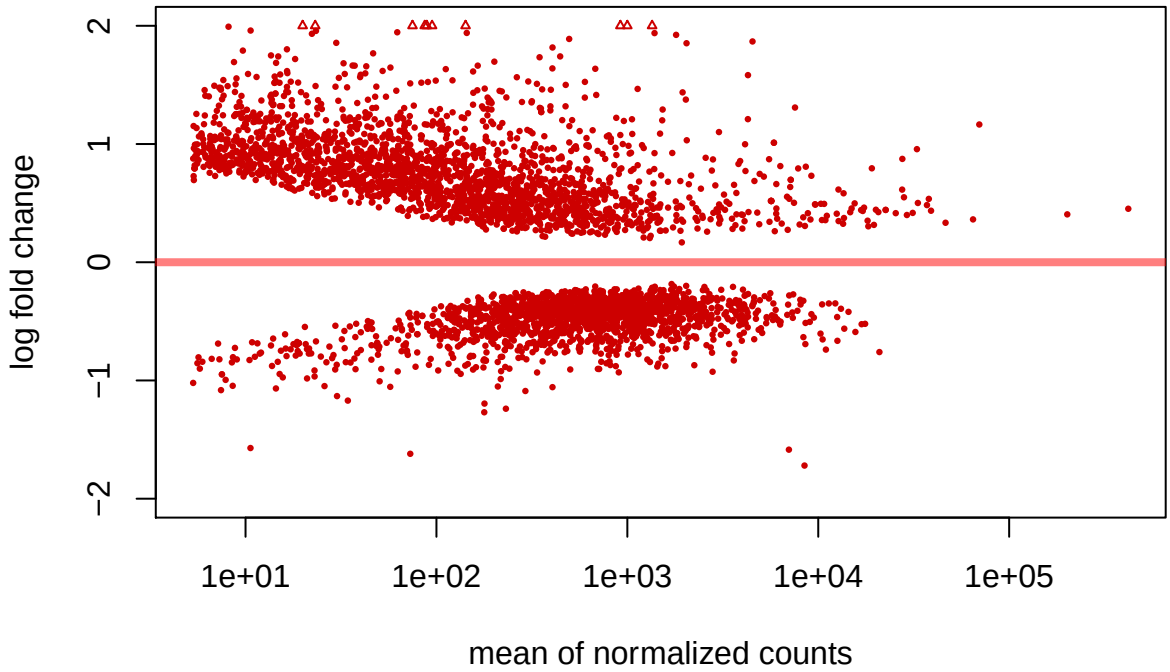
```
# csv of sig results
write.csv(as.data.frame(res05), file = "condition_treated_results.csv")
```

We can see from the summary of significant results giving percentage of genes up or down regulated dependent on the significance threshold. Next up is to plot our results so we can gain a better handle on whats happening in the data. The first plot is a MAplot which shows the log of the fold change on the y axis and the mean of the normalised counts on the x axis. We see genes with the greatest differece highlight in red. where as the nonsignificant points have been removed. The next plot is the normalized counts per group. we see that ech condition cluster together albeit with some variance intrasample. This plot is repeated using ggplot2.

The first heat map doesnt show stark contrasts between conditions but seeing as there should only be two nucleotides difference between each condition any difference is positive result. Interestingly in the original analysis we see unmod 5 and 6 cluster early and seperately to the rest of the samples and here we see clear differences in these samples also. For the heatmap with dendrogram we see again when clustered using the r-log transformation we see the modified samples are grouped together but not totally seperate from the unmodified groups.
PCA analysis shows clear segregation between conditions and some clustering of samples. A potential reason for this could be homogeneity between some the samples depending on how the samples were passaged and kept as well as efficiency of the CRISPR/Cas9 cell line manipulation. Cooks distance is a measurements of the effect of deleting an outlier from the data. The box plot of the cooks distance is consistent for all samples.

## Analysis plots

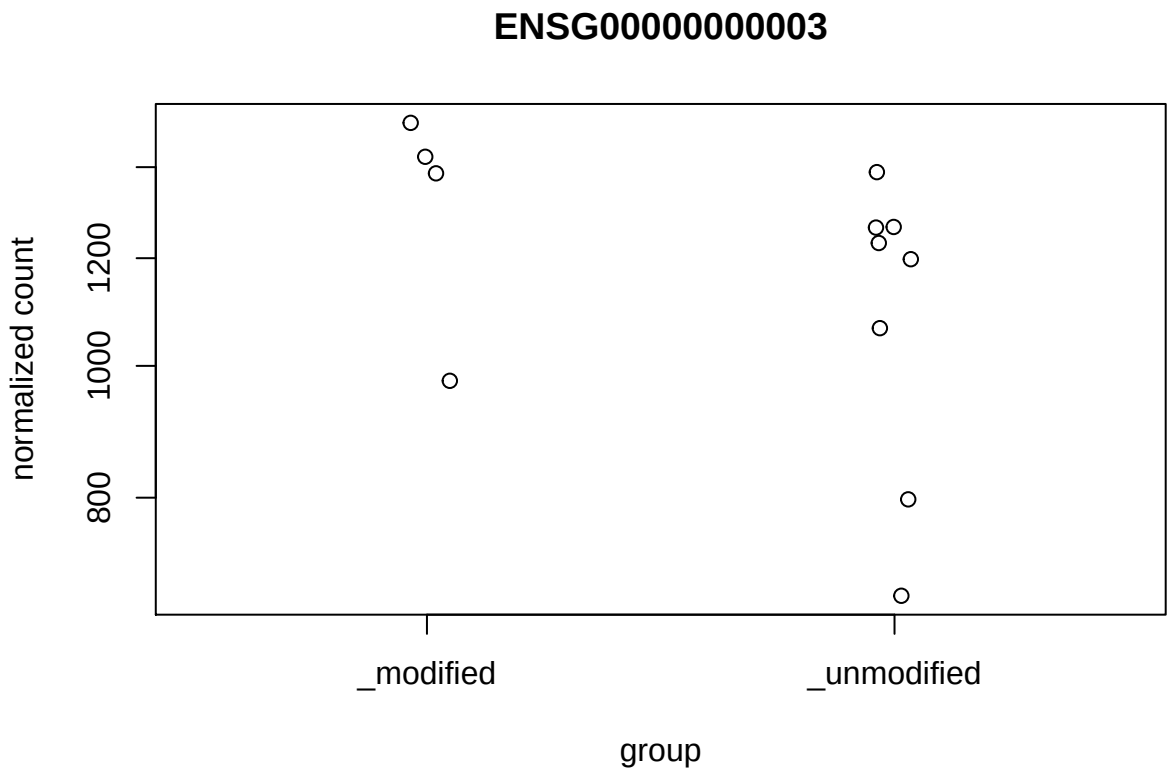
```
# plot results
plotMA(res, ylim=c(-2,2))
idx <- identify(res$baseMean, res$log2FoldChange)
```



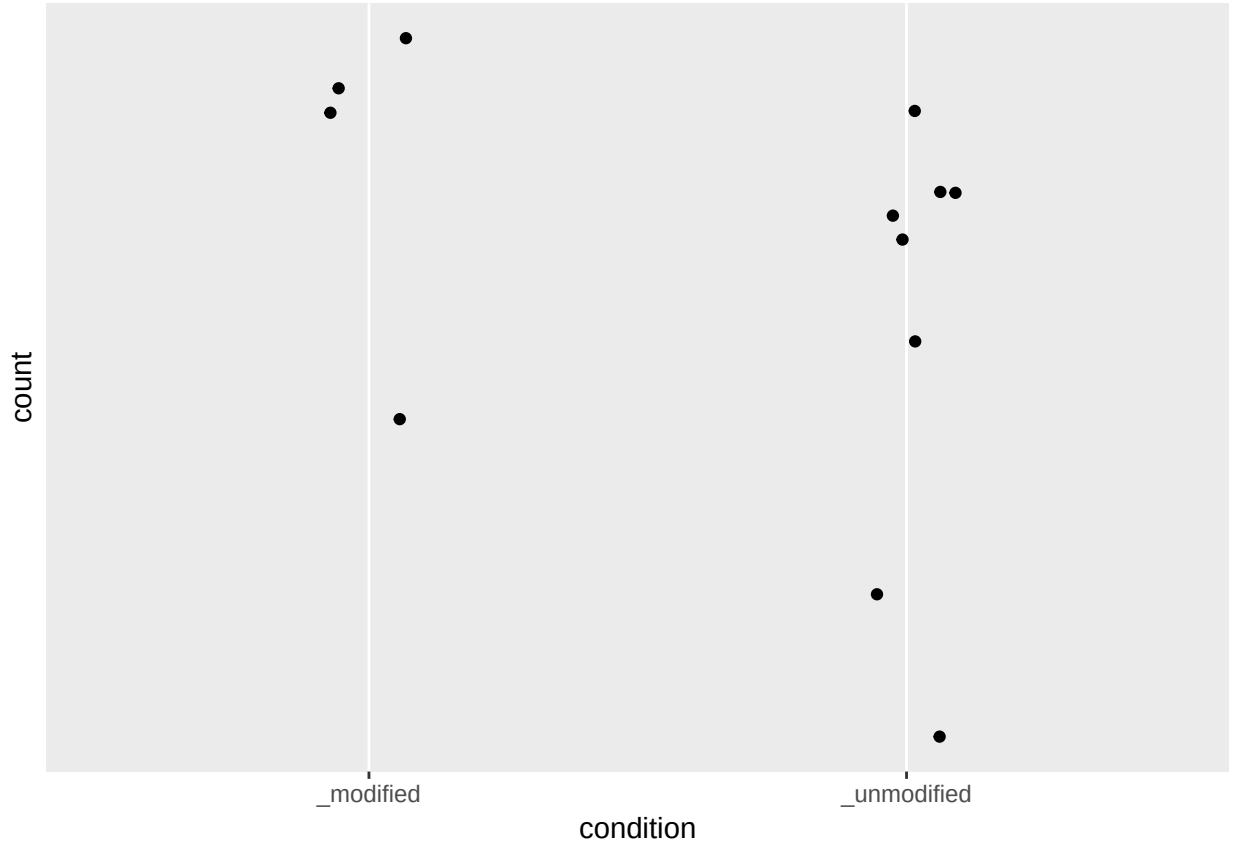

```
rownames(res)[idx]
```

```
## character(0)
```

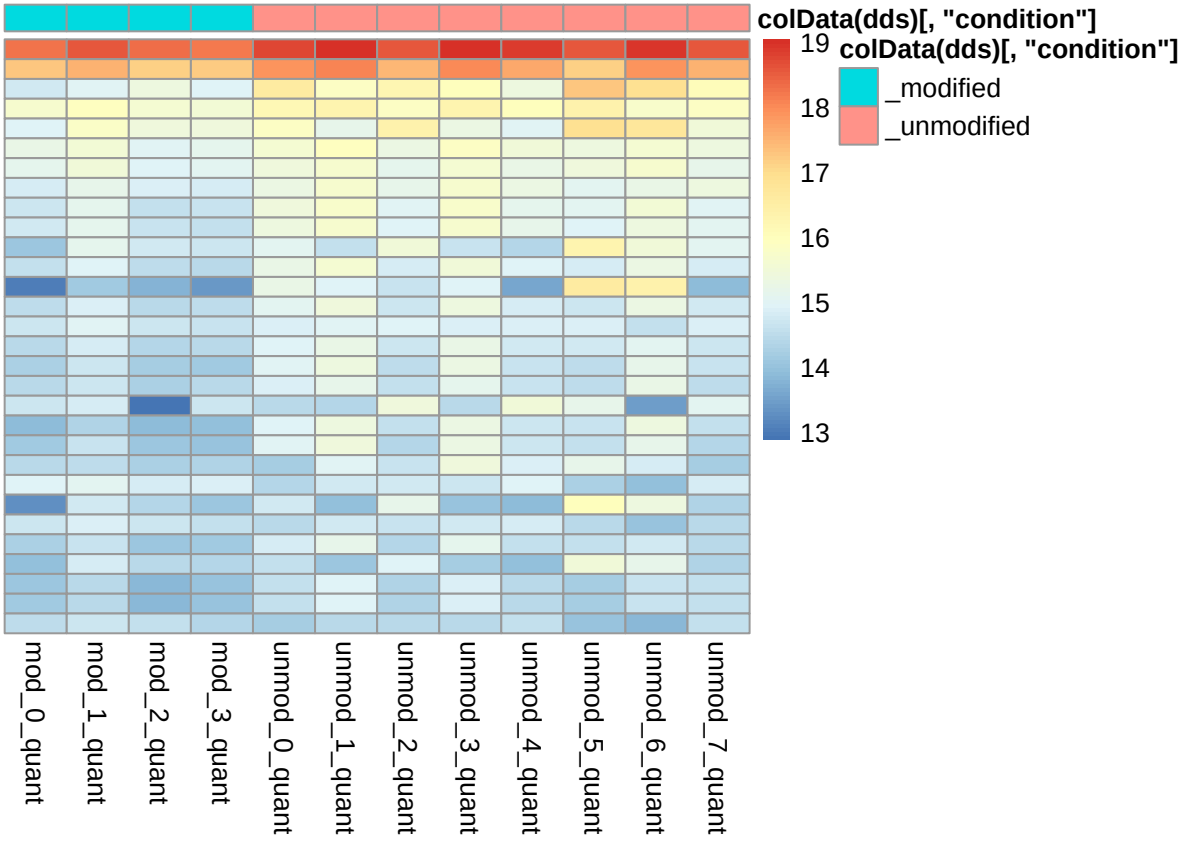
```
plotCounts(dds, gene=which.min(res$padj), intgroup="condition")
```



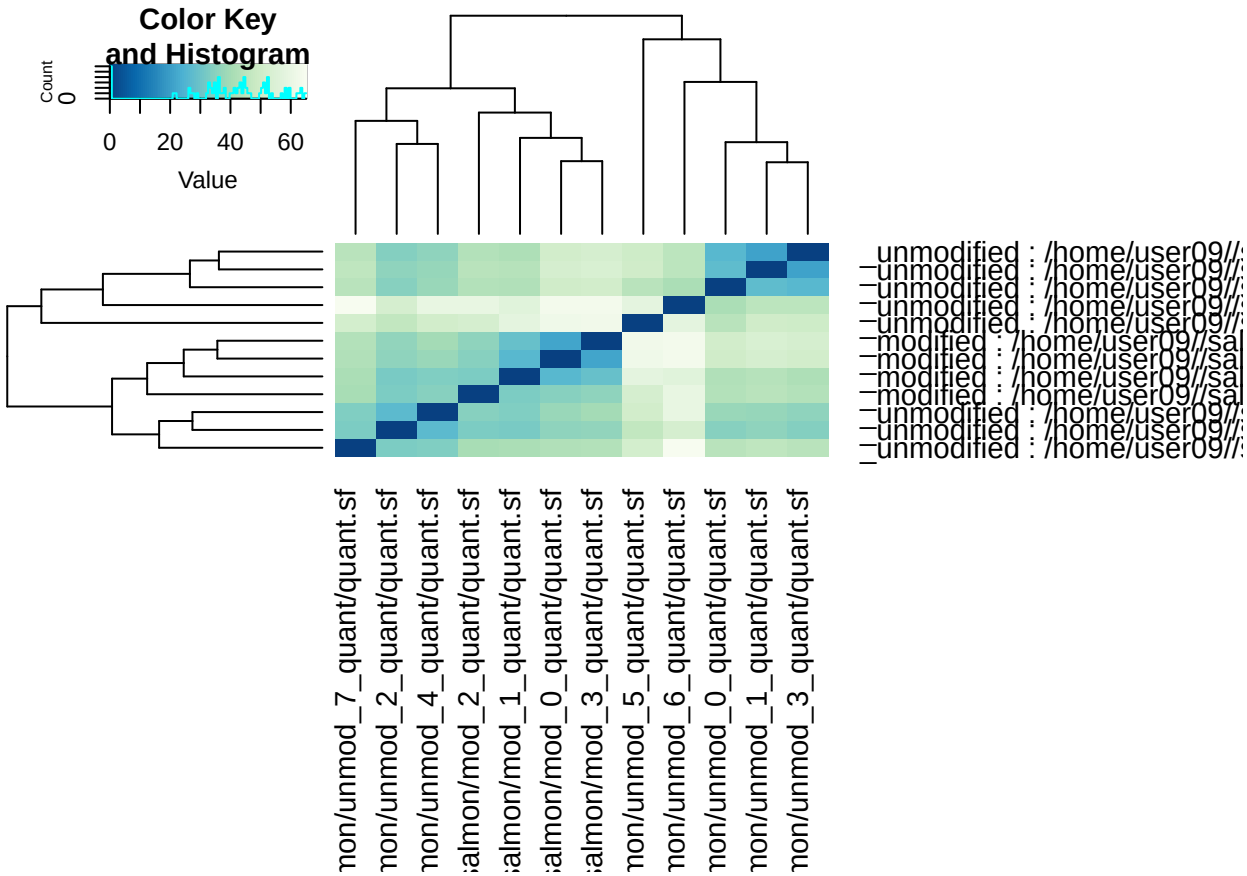

```
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="condition", returnData=TRUE)

ggplot(d, aes(x=condition, y=count, label="condition")) +
  geom_point(position=position_jitter(w=0.1,h=0)) +
  scale_y_log10(breaks=c(25,100,400))
```



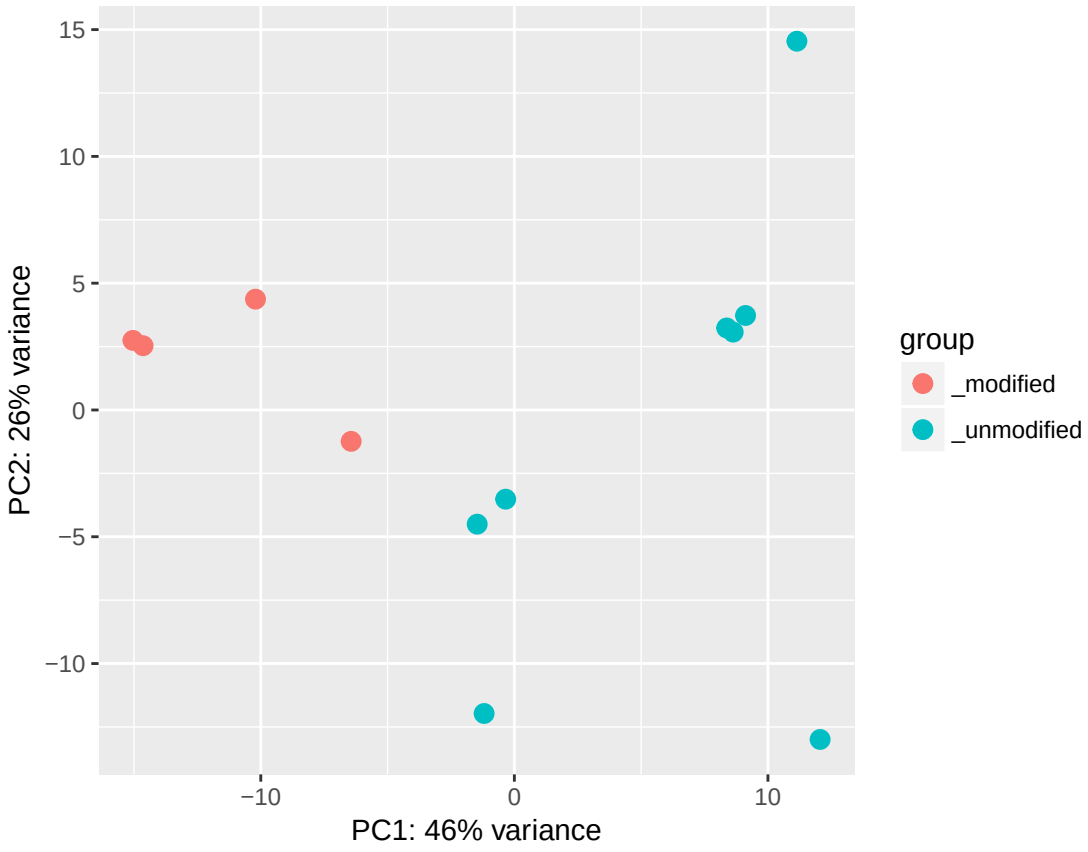

```
# nice heatmap, pheatmap
hmcol <- colorRampPalette(brewer.pal(9,'GnBu'))(100)
select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE)[1:30]
ntd <- normTransform(dds)
df <- as.data.frame(colData(dds)[,"condition"])
rownames(df) <- colnames(assay(ntd)[select,])
pheatmap(assay(ntd)[select,], cluster_rows=FALSE, show_rownames=FALSE,cluster_cols=FALSE,annotation_col=
```



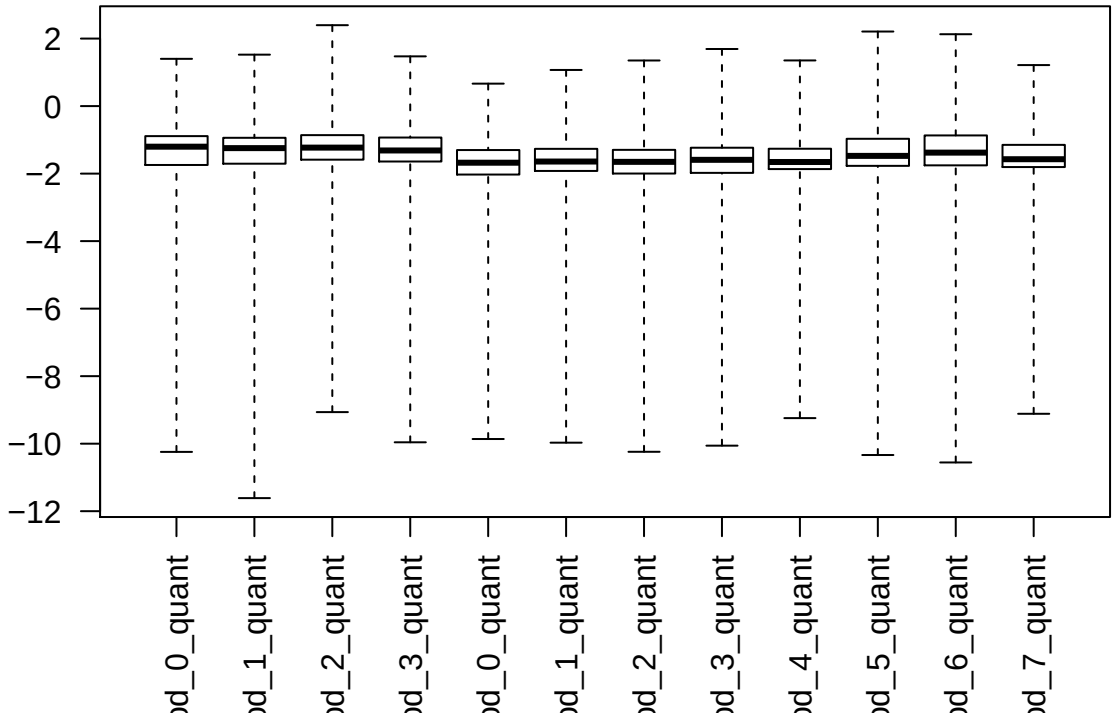

```
# transformed values
rld <- rlog(dds, blind=FALSE)
vsd <- varianceStabilizingTransformation(dds, blind=FALSE)
vsd.fast <- vst(dds, blind=FALSE)

# heatmap cluster by distances
# rld is the regularised log (rlog) transformed data looks like a distancce matrix
distsRL <- dist(t(assay(rld)))
mat<- as.matrix(distsRL)
rownames(mat) <- colnames(mat) <- with(colData(dds),paste(condition,files , sep=' : '))

hc <- hclust(distsRL)
heatmap.2(mat, Rowv=as.dendrogram(hc),symm=TRUE, trace='none',col = rev(hmcol), margin=c(13, 13))
```




```
# plot PCA
plotPCA(rld, intgroup=c("condition"))
```




```
# plot cooks dist
boxplot(log10(assays(dds)[["cooks"]]), range=0, las=2)
```




The curved plot is a plot of the probability weighting function. This plots the porportion of genes differentially

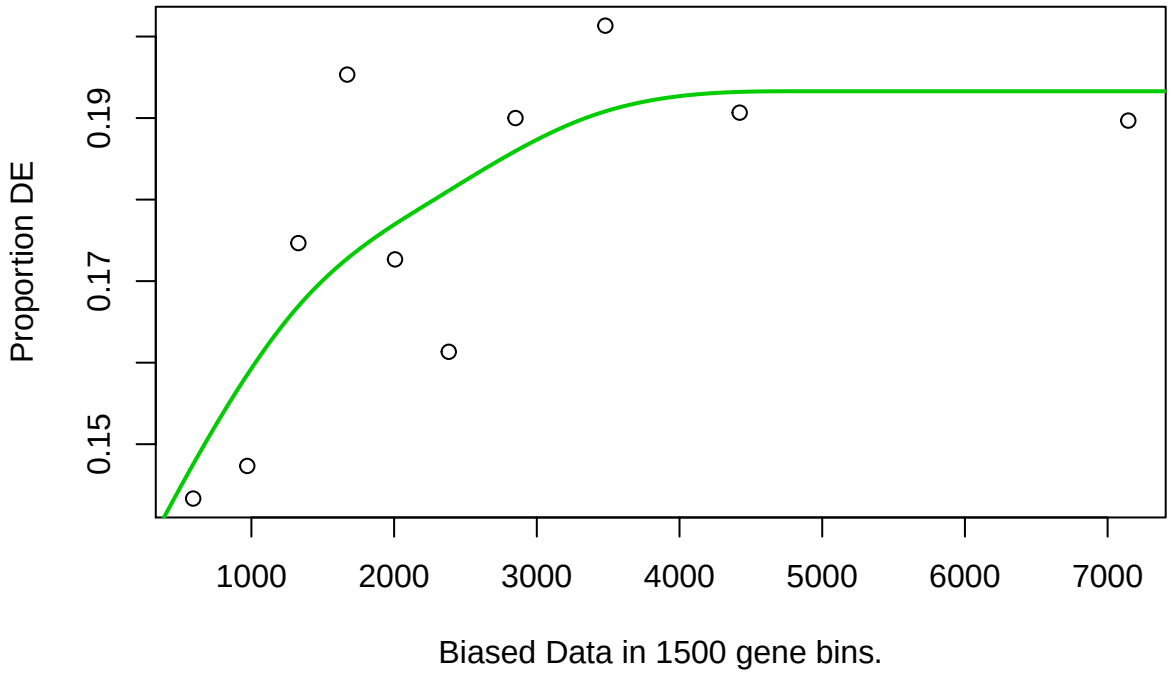
expressed (encoded 0 for non-differentially expressed and 1 for differentially expressed) against some bias data, in this case the bias data is the gene length. The pwf allows us to correct for gene length bias in our analysis. The wallenius approximation returns a list of ontologies that are associated with our significant gene set.

```
genes <- as.integer(res05$padj < 0.05)  # sig genes after adjustment
not_na <- !is.na(genes)  # not not
names(genes) = rownames(dds)
genes <- genes[not_na]  #is not not
head(genes)
```

```
## ENSG00000000003 ENSG00000000419 ENSG00000000457 ENSG00000000460
##               0               0               0               1
## ENSG00000001036 ENSG00000001084
##               0               0
```

```
pwf <- nullp(genes, "hg19", "ensGene")  # probability weight function
```

```
## Warning in pcls(G): initial point very close to some inequality constraints
```




```
GO.wall <- goseq(pwf, "hg19", "ensGene", use_genes_without_cat = F)  # wallenius approximation \
# over and under GO categories
head(GO.wall, 20)
```

```
##         category over_represented_pvalue under_represented_pvalue
## 11115 GO:0044822           6.145317e-25                         1
## 1473  GO:0003723           1.026705e-21                         1
## 2501  GO:0005730           3.433227e-13                         1
## 2898  GO:0006396           6.856713e-12                         1
## 8922  GO:0034660           8.454703e-11                         1
## 2881  GO:0006364           2.845835e-10                         1
## 8849  GO:0034470           2.940762e-10                         1
## 5373  GO:0016072           3.623287e-10                         1
## 9977  GO:0042254           8.154149e-10                         1
```

```
## 7685  GO:0031981           1.062016e-09                         1
## 7683  GO:0031974           9.960921e-09                         1
## 10498 GO:0043233           9.960921e-09                         1
## 14595 GO:0070013           9.960921e-09                         1
## 7079  GO:0030684           1.117761e-08                         1
## 6659  GO:0022613           1.435709e-08                         1
## 10988 GO:0044428           3.091182e-08                         1
## 1452  GO:0003676           4.364994e-08                         1
## 19008 GO:1990904           1.722045e-07                         1
## 16535 GO:0097190           1.739296e-07                         1
## 7010  GO:0030529           2.436379e-07                         1
##       numDEInCat numInCat                                    term ontology
## 11115        350     1182                     poly(A) RNA binding       MF
## 1473         418     1538                             RNA binding       MF
## 2501         221      786                               nucleolus       CC
## 2898         238      897                          RNA processing       BP
## 8922         158      559                 ncRNA metabolic process       BP
## 2881          87      266                         rRNA processing       BP
## 8849         121      406                        ncRNA processing       BP
## 5373          88      271                  rRNA metabolic process       BP
## 9977         101      328                     ribosome biogenesis       BP
## 7685         719     3323                           nuclear lumen       CC
## 7683         842     4021                membrane-enclosed lumen       CC
## 10498        842     4021                        organelle lumen       CC
## 14595        842     4021           intracellular organelle lumen       CC
## 7079          35       75                             preribosome       CC
## 6659         130      471    ribonucleoprotein complex biogenesis       BP
## 10988        772     3667                             nuclear part       CC
## 1452         733     3460                    nucleic acid binding       MF
## 19008        184      744               ribonucleoprotein complex       CC
## 16535        131      482             apoptotic signaling pathway       BP
## 7010         183      743 intracellular ribonucleoprotein complex       CC
```

```
length(GO.wall[, 1])
```

```
## [1] 19681
```

```
enriched.GO <- GO.wall$category[p.adjust(GO.wall$over_represented_pvalue, method = "BH") <
    0.05]  # the enriched Go categories
head(enriched.GO)
```

```
## [1] "GO:0044822" "GO:0003723" "GO:0005730" "GO:0006396" "GO:0034660"
## [6] "GO:0006364"
```

```
library(GO.db)  # Enriched information for top 5 ontologies
for (go in enriched.GO[1:5]) {
    print(GOTERM[[go]])
    cat("--------------------------------------------------------\n")
}
```

```
## GOID: GO:0044822
## Term: poly(A) RNA binding
## Ontology: MF
## Definition: Interacting non-covalently with a poly(A) RNA, a RNA
##     molecule which has a tail of adenine bases.
## Synonym: poly(A)-RNA binding
```

```
## Synonym: poly-A RNA binding
## ----------------------------------------------------
## GOID: GO:0003723
## Term: RNA binding
## Ontology: MF
## Definition: Interacting selectively and non-covalently with an RNA
##     molecule or a portion thereof.
## ----------------------------------------------------
## GOID: GO:0005730
## Term: nucleolus
## Ontology: CC
## Definition: A small, dense body one or more of which are present
##     in the nucleus of eukaryotic cells. It is rich in RNA and
##     protein, is not bounded by a limiting membrane, and is not
##     seen during mitosis. Its prime function is the transcription
##     of the nucleolar DNA into 45S ribosomal-precursor RNA, the
##     processing of this RNA into 5.8S, 18S, and 28S components of
##     ribosomal RNA, and the association of these components with 5S
##     RNA and proteins synthesized outside the nucleolus. This
##     association results in the formation of ribonucleoprotein
##     precursors; these pass into the cytoplasm and mature into the
##     40S and 60S subunits of the ribosome.
## ----------------------------------------------------
## GOID: GO:0006396
## Term: RNA processing
## Ontology: BP
## Definition: Any process involved in the conversion of one or more
##     primary RNA transcripts into one or more mature RNA molecules.
## Synonym: GO:0006394
## Secondary: GO:0006394
## ----------------------------------------------------
## GOID: GO:0034660
## Term: ncRNA metabolic process
## Ontology: BP
## Definition: The chemical reactions and pathways involving
##     non-coding RNA transcripts (ncRNAs).
## Synonym: ncRNA metabolism
## ----------------------------------------------------
```

# Discussion

We see from the results of the differential expression analysis that of the total 37,625 non zero genes, 1675 (4.5%) genes are significantly up regulated while 1296 (3.4%) genes are significantly down regulated. The box plot of cooks distance would suggest not to remove any samples where the original analysis removed two samples from analysis However looking at the heatmap and dendrogram we can see that the removed samples (unmod 5 & 6) do show altered expression levels different to that of the unmodified and modified samples. PCA analysis showed that the first two principal components contained 72% of the variance between samples and that each treatment grouped seperately. Interestingly we see RNA binding as the top two results one associated with poly-A binding the other a more general RNA binding. Looking through the top ten differentially down regulated genes we see the appearence immune genes such as interleukin 9 receptor *IL9R* and *PAG1* for the over-all significant genes (up and down regulated) some are linked to neurodevelopment but none have been strongly linked to schizophrenia. The published paper states that immune processes are

significantly effected in the pathway analysis however immune system processes do not appear in the top 50 effeceted ontologies. Looking through the list of differentially expressed genes we do see that only ~20 genes are down regulated (FC < -1 & padj <0.05). We do see RNA process and ribosome processes are effected. mTOR activation has been previously shown to activate ribosomal RNA transcription. Loss of mTOR has also been shown to decrease phosphorylation of *FOXO1* and *FOXO3a* activating stress response proteins [8]. These result would seem to suggest that the mTOR signalling pathway is affected by 13 DNR in the 5' region of the *DPYSL* gene.

# Bibliography


Martins-de-Souza, Daniel, et al. "The protein interactome of collapsin response mediator protein-2 (CRMP2/DPYSL2) reveals novel partner proteinsin brain tissue." PROTEOMICS-Clinical Applications 9.9-10 (2015): 817-831.

Pulver, Ann E., et al. "Schizophrenia: a genome scan targets chromosomes 3p and 8p as potential sites of susceptibility genes." American journal of medical genetics 60.3 (1995): 252-260.

Holmans, Peter A., et al. "Genomewide linkage scan of schizophrenia in a large multicentre pedigree sample using single nucleotide polymorphisms." Molecular psychiatry 14.8 (2009): 786-795.

Fallin, M. Daniele et al. "Bipolar I Disorder and Schizophrenia: A 440- single-Nucleotide Polymorphism Screen of 64 Candidate Genes among Ashkenazi Jewish Case-Parent Trios." American Journal of Human Genetics 77.6 (2005): 918-936. Print.

Liu, Yaping, et al. "Functional variants in DPYSL2 sequence increase risk of schizophrenia and suggest a link to mTOR signaling." G3: Genes| Genomes| Genetics 5.1 (2015): 61-72.

Pham, X., et al. "The DPYSL2 gene connects mTOR and schizophrenia." Translational Psychiatry 6.11 (2016): e933.

Patro, Rob, Geet Duggal, and Carl Kingsford. "Salmon: accurate, versatile and ultrafast quantification from RNA-seq data using lightweight-alignment." bioRxiv (2015): 021592.

Laplante, Mathieu, and David M. Sabinini. "mTOR signaling at a glance." Journal of cell science 122.20 (2009): 3589-3594.