Exploring NSD3 Driven Gene Expression Profiles in Lung Squamous Cell Carcinoma

David Dilworth and Dalia Barsyte-Lovejoy

2018-07-17

Objective

NSD3 is frequently amplified as part of the 8p11-12 focal amplification in several types of cancer. Here we set out to explore the gene expression profiles of NSD3 amplified versus non-amplified lung squamous cancer samples from the TCGA. However, because this event takes place in the context of a focal amplification we will also compare samples with high versus low NSD3 expression. The goal of this experiment is to identify putative oncogenic signalling pathways NSD3 may regulate, as well as inform on the biological role of this protein in the context of gene expression.

Experimental Methods

This experiment takes advantage of several phenomenal R packages, which we use for interfacing with data repositories, as well as analyzing and presenting data. The following is the full code used in the analysis. The results shown here are in whole or part based upon data generated by the TCGA Research Network: HTTP://cancergenome.nih.gov/." [1]

Load Required Packages into R

```
library(SummarizedExperiment)  ## [2]
library(biomaRt)  ## [3]
library(TCGAbiolinks)  ## [4]
library(FirebrowseR)  ## [5]
library(tidyverse)  ## [6]
library(clusterProfiler)  ## [7]
library(limma)  ## [8]
library(Glimma)  ## [9]
library(magrittr)  ## [10]
library(edgeR)  ## [11]
```

Download & save copy number data for NSD3 (WHSC1L1)

```
# This code will use FirebrowseR to download and save NSD3
# copy number data for TCGA-LUSC. If the data has already
# been downloaded then it will be loaded into R.

if (!file.exists(file = "NSD3.copynumber.rds")) {

    cn <- Analyses.CopyNumber.Genes.All(format = "csv", gene = "WHSC1L1",
        cohort = "LUSC", page_size = 1000)
    saveRDS(cn, file = "NSD3.copynumber.rds")

} else {

    cn <- readRDS(file = "NSD3.copynumber.rds")

}
```

Annotation of Amplified & Non-Amplified TCGA-LUSC Samples based on GISTIC Copy Number Score

```
# Here we are assigning the amplification status of NSD3 for
# each patient sample.

amp <- cn %>% filter(all_copy_number >= 2) %>% dplyr::select(tcga_participant_barcode) %>%
    .[, 1]

non.amp <- cn %>% filter(all_copy_number < 1 & all_copy_number >
    -1) %>% dplyr::select(tcga_participant_barcode) %>% .[, 1]
```

Download & save TCGA-LUSC data

```
# Download and Save Transcriptome Data using TCGAbiolinks. If
# the data has previously been downloaded it will be loaded
# directly into R.

if (!file.exists(file = "lusc_HTSeqCounts.rda")) {
    query.exp <- GDCquery(project = "TCGA-LUSC", data.category = "Transcriptome Profiling",
        data.type = "Gene Expression Quantification", workflow.type = "HTSeq - Counts",
        sample.type = "Primary solid Tumor")

    GDCdownload(query.exp)

    lusc.counts <- GDCprepare(query = query.exp, save = TRUE,
        save.filename = "lusc_HTSeqCounts.rda")

} else {

    load(file = "lusc_HTSeqCounts.rda")
    lusc.counts <- data
}
```

Perform differential expression analysis using Limma

```
# Annotate NSD3 Amplified Samples with the Gene Expression
# dataset.

colData(lusc.counts)$Status <- ifelse(grepl(paste(amp, collapse = "|"),
    colData(lusc.counts)$patient), "Amplified", "Non-Amplified")
```

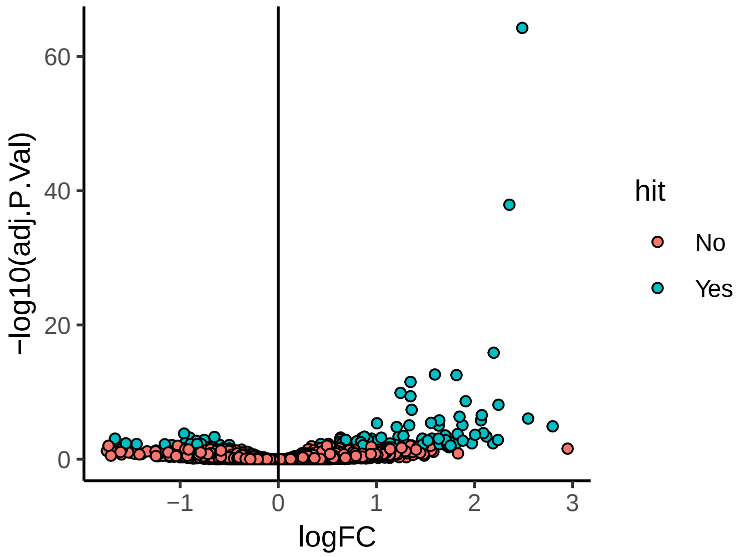
```r
counts.dge <- DGEList(assay(lusc.counts), group = colData(lusc.counts)$Status)

# Set the experimental design for testing of differenetial
# expression

design <- cbind(Amplified = (grepl(paste(amp, collapse = "|"),
    colData(lusc.counts)$patient)) * 1, Non.Amplified = (grepl(paste(non.amp,
    collapse = "|"), colData(lusc.counts)$patient)) * 1)

cont.matrix <- makeContrasts(AmplifiedvsNonAmplified = "Amplified-Non.Amplified",
    levels = design)


# Run Differential Expression Analysis Using Limmma

keep <- filterByExpr(counts.dge, design)
counts.dge <- counts.dge[keep, , keep.lib.sizes = FALSE]
counts.dge <- calcNormFactors(counts.dge)

voom.out <- voom(counts.dge, design)
vfit <- lmFit(voom.out, design)
vfit <- contrasts.fit(vfit, contrasts = cont.matrix)
efit <- eBayes(vfit)

results <- decideTests(efit, p.value = 0.01)

summary(decideTests(efit))
```

```
##        AmplifiedvsNonAmplified
## Down                       105
## NotSig                   27662
## Up                         196
```

```r
# Use biomaRt to annotate genes. This will save the
# annotation data the first time it is downloaded.

if (!file.exists(file = "annotation.amp.rds")) {
    mart <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")

    ensembl_ids <- unique(rownames(efit$p.value))

    anno <- getBM(mart = mart, attributes = c("ensembl_gene_id",
        "hgnc_symbol", "entrezgene", "description", "chromosome_name",
        "band"), filters = "ensembl_gene_id", values = ensembl_ids,
        uniqueRows = FALSE)

    anno <- anno[!duplicated(anno$ensembl_gene_id), ]

    saveRDS(anno, file = "annotation.amp.rds")

} else {

    anno <- readRDS(file = "annotation.amp.rds")
}


# Get results in table format with annotations. The results
# are saved as a csv file.

annotated.results <- as_tibble(topTable(efit, number = Inf, sort = "none"),
    rownames = "ensembl_gene_id") %>% inner_join(anno, by = "ensembl_gene_id") %>%
    mutate(hit = ifelse(adj.P.Val < 0.01, "Yes", "No"))

write.csv(annotated.results, file = "DE.NSD3_amplified.csv")
```

Plot results of differential expression analysis

```
# Volcano Plot of Differentially Expressed Genes

ggplot(annotated.results, aes(logFC, -log10(adj.P.Val))) + geom_point(pch = 21,
    aes(fill = hit)) + theme_classic() + geom_vline(xintercept = 0)
```



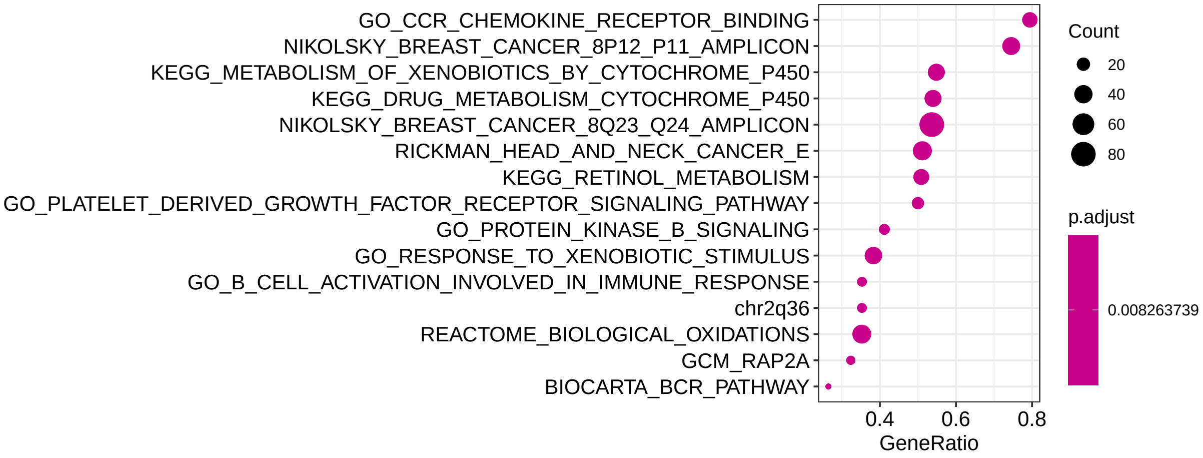
Perform Gene Set Enrichment Analysis

To determine what types of genes may be differentially expressed we use clusterProfiler to perform gene set enrichment analysis. Gene sets were downloaded from the Molecular Signatures Database (http://software.broadinstitute.org/gsea/msigdb/index.jsp).

```
# Read in molecular signatures from all collections

all <- read.gmt("msigdb.v6.2.entrez.gmt")

# Prepare a ranked gene list for clusterProfiler GSEA
# analysis. The data has been ranked by Limma's t-statistic.

genelist <- annotated.results %>% dplyr::select(entrezgene, t) %>%
    arrange(desc(t)) %>% as.data.frame()

geneList <- genelist[, 2]

names(geneList) <- as.character(genelist[, 1])

geneList <- sort(geneList, decreasing = TRUE)

# Run GSEA from clusterProfiler package

gsea.amp <- GSEA(geneList, TERM2GENE = all, seed = 2)
```

Plot 15 Gene Set Terms Associcated

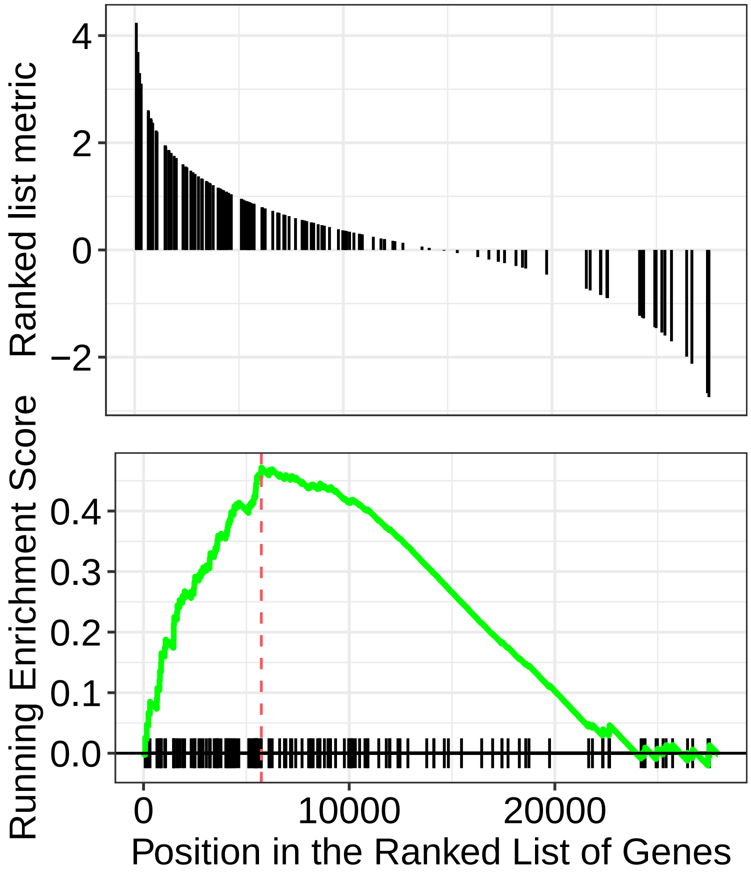
```
# Plot using clusterProfiler's dotplot function

dotplot(gsea.amp, showCategory = 15)
```



Here we can see an enrichment for genes found within the chromosome 8 amplification.

Plot GSEA for Chr8

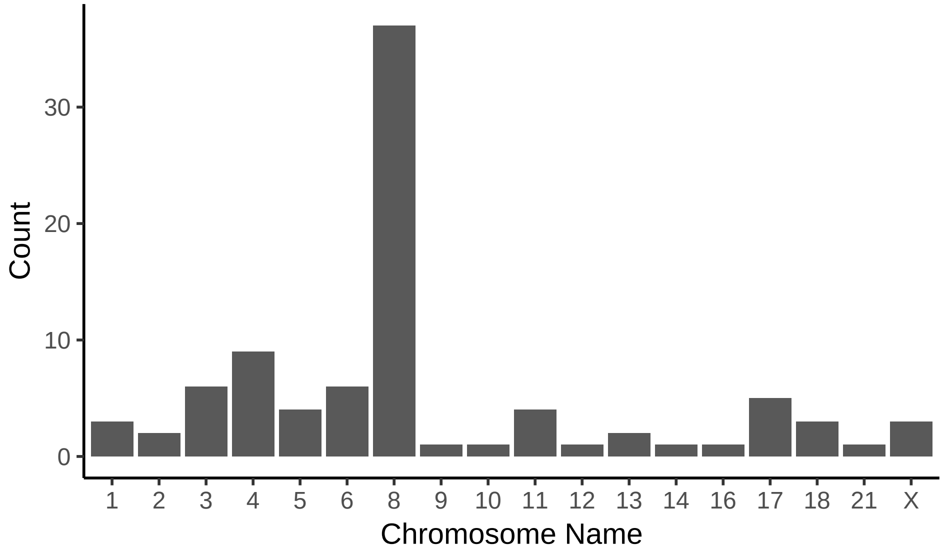
```
gseaplot(gsea.amp, "chr8q24")
```



Barplot of upregulated DE genes shows an enrichment in genes located on chromosome 8

```
# This shows that most of the upregulated genes are within
# the focal amplification.


de.pval <- as_tibble(efit$p.value, rownames = "ensembl_gene_id") %>%
    inner_join(anno, by = "ensembl_gene_id") %>% rename_at(2,
    ~"p.value")

genes.up <- results[, 1] %>% as.data.frame() %>% rownames_to_column() %>%
    magrittr::set_colnames(c("Gene", "DE")) %>% filter(DE ==
    1) %>% .[, 1]

plot.chr.up <- de.pval %>% filter(ensembl_gene_id %in% genes.up) %>%
    group_by(chromosome_name) %>% summarise(count = n()) %>%
    mutate(chromosome_name = factor(chromosome_name, levels = c(as.character(seq(1,
        22, 1)), "X"))) %>% ggplot(aes(chromosome_name, count)) +
    geom_bar(stat = "identity") + ylab("Count") + xlab("Chromosome Name") +
    theme_classic()
```

```
plot.chr.up
```



Observations.

As previously stated, NSD3 amplification in cancer most commonly occurs as the result of a large focal amplification of the 8p11-12 locus. Here we can see that when you perform differential gene expression analysis on NSD3 amplified samples you get an enrichment for genes located within the amplicon. There are a number of genes encoded in this region, including chromatin modifiers and assumed oncogenes. Therefore, it is difficult to untangle NSD3-dependent changes in gene expression versus those that may be driven by other gene products originating from this region. Therefore, to look at differences in gene expression profiles based solely on stratifying samples by NSD3 expression, we can first filter out all samples that have the 8p11-12 amplification and then compare gene expression between the 1st and 5th quintiles of NSD3 expression within TCGA-LUSC.

Differential Gene Expression Analysis of Sample with High vs Low NSD3 Expression

```
# First we filter out any samples that we flagged as NSD3
# amplified in the previous analysis.

NSD3.exp <- Samples.mRNASeq(format = "csv", gene = "WHSC1L1",
    cohort = "LUSC", page_size = 600)

NSD3.exp.nonamp <- NSD3.exp %>% filter(sample_type == "TP") %>%
    filter(tcga_participant_barcode %in% non.amp)

# Next, samples are assigned into a quintile based on NSD3
# expression levels. The 1st quintile will represent 20% of
# samples with the lowest NSD3 expression, while the 5th
# presents the top 20% based on NSD3 expression

NSD3.exp.quantile <- within(NSD3.exp.nonamp, quartile <- cut(expression_log2,
    quantile(expression_log2, probs = 0:5/5), include.lowest = TRUE,
    labels = FALSE))

NSD3.low <- NSD3.exp.quantile %>% filter(quartile == 1) %>% .[,
    1]

NSD3.high <- NSD3.exp.quantile %>% filter(quartile == 5) %>%
    .[, 1]

lusc.filter <- lusc.counts[, lusc.counts$patient %in% NSD3.low |
    lusc.counts$patient %in% NSD3.high]

colData(lusc.filter)$quantile <- ifelse(grepl(paste(NSD3.low,
    collapse = "|"), colData(lusc.filter)$patient), "Low", "High")

counts.dge <- DGEList(assay(lusc.filter), group = colData(lusc.filter)$quantile)



# Set the experimental design for differential gene
# expression testing with Limma

design <- cbind(high = (grepl(paste(NSD3.high, collapse = "|"),
    colData(lusc.filter)$patient)) * 1, low = (grepl(paste(NSD3.low,
    collapse = "|"), colData(lusc.filter)$patient)) * 1)


cont.matrix <- makeContrasts(highvslow = "high-low", levels = design)


# Run Limma DE analysis

keep <- filterByExpr(counts.dge, design)
counts.dge <- counts.dge[keep, , keep.lib.sizes = FALSE]
counts.dge <- calcNormFactors(counts.dge)

voom.out <- voom(counts.dge, design)
vfit <- lmFit(voom.out, design)
vfit <- contrasts.fit(vfit, contrasts = cont.matrix)
efit <- eBayes(vfit)
tfit <- treat(vfit, lfc = log2(1.2))


results <- decideTests(tfit, p.value = 0.01)

summary(decideTests(tfit, p.value = 0.01))
```

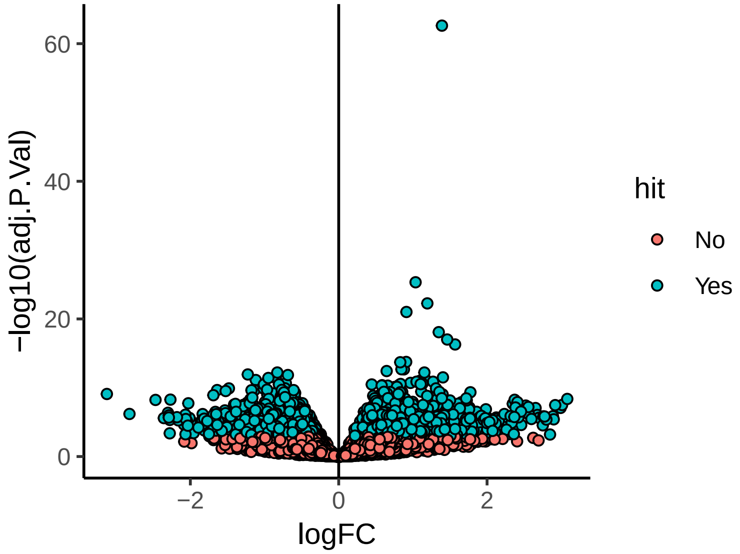
```
##        highvslow
## Down         424
## NotSig     20540
## Up           573
```

```
# Use biomaRt to annotate the genes with several identifiers
# and save the annotation for future loading.
```

```r
if (!file.exists(file = "annotation.exp.rds")) {
    mart <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")

    ensembl_ids <- unique(rownames(efit$p.value))

    anno <- getBM(mart = mart, attributes = c("ensembl_gene_id",
        "hgnc_symbol", "entrezgene", "description", "chromosome_name",
        "band"), filters = "ensembl_gene_id", values = ensembl_ids,
        uniqueRows = FALSE)

    anno <- anno[!duplicated(anno$ensembl_gene_id), ]

    saveRDS(anno, file = "annotation.exp.rds")

} else {

    anno <- readRDS(file = "annotation.exp.rds")

}


# Get results in table format with annotations


annotated.results <- as_tibble(topTable(efit, number = Inf, sort = "none"),
    rownames = "ensembl_gene_id") %>% inner_join(anno, by = "ensembl_gene_id") %>%
    mutate(hit = ifelse(adj.P.Val < 0.001, "Yes", "No"))

write.csv(annotated.results, file = "DE.NSD3_HighLow.csv")
```

PLot Differential Expression Results as a Volcano Plot

```r
# Volcano Plot of Differentially Expressed Genes

ggplot(annotated.results, aes(logFC, -log10(adj.P.Val))) + geom_point(pch = 21,
    aes(fill = hit)) + theme_classic() + geom_vline(xintercept = 0)
```



Note: We observe significantly more DE genes than in the previous analysis.

Use Glimma package to generate an interactive html plot and table of DE genes.

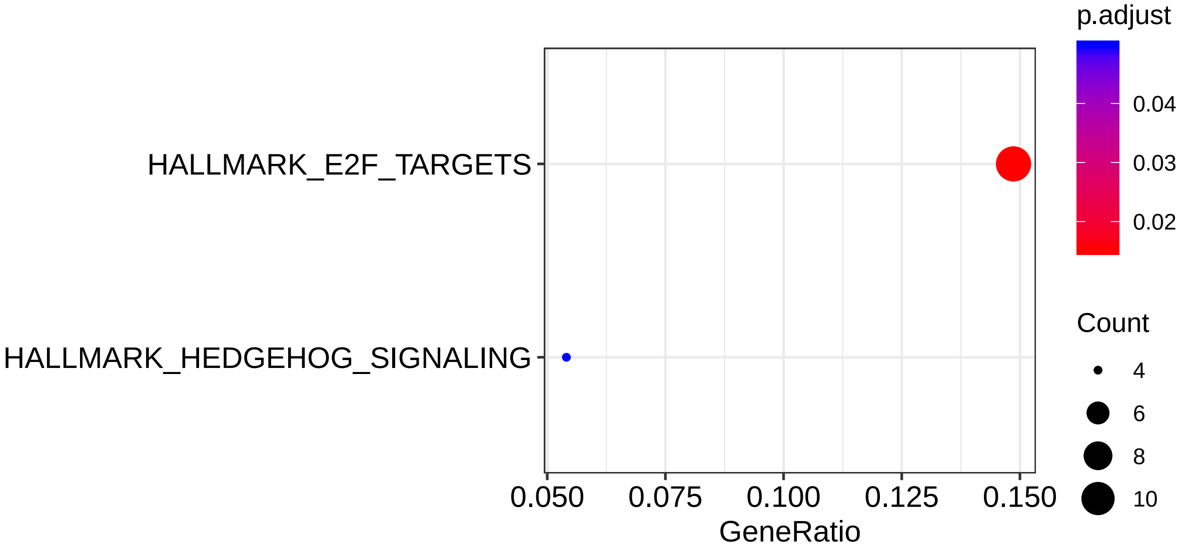
```
# Run glMDPlot function from Glimma package. Plot has been
# posted to opennotebook.org

glMDPlot(tfit, anno = anno, side.main = "hgnc_symbol", counts = voom.out,
    groups = counts.dge$samples$group, launch = TRUE, status = results[,
        1], main = "TCGA-LUSC Gene Expression - NSD3 Low Vs. High")
```

Test Enrichment of Hallmark Gene Set for Up and Down Regulated Genes

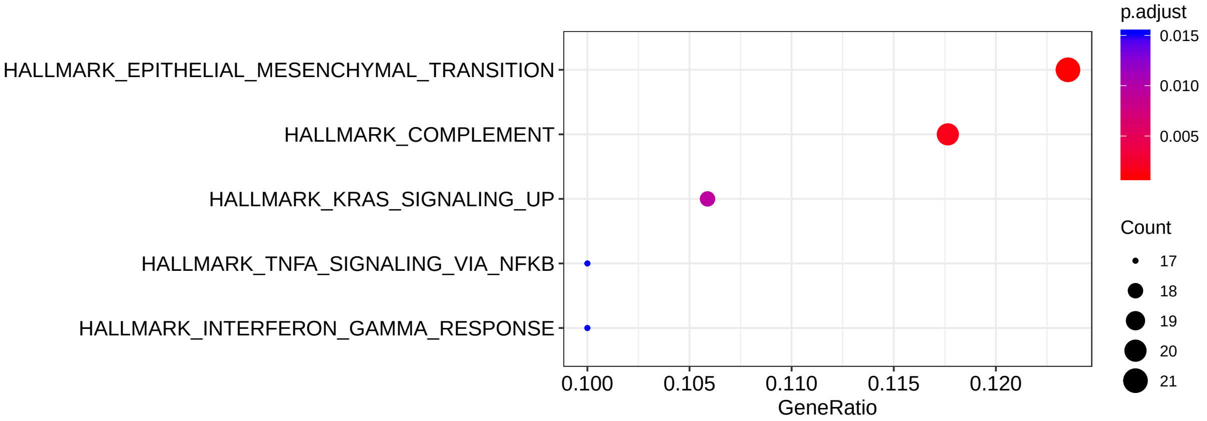
```
# Load hallmark gene set - downloaded from
# http://software.broadinstitute.org/gsea/msigdb/index.jsp.

H <- read.gmt("h.all.v6.2.entrez.gmt")

# Get entrez ids for up and down regulated genes

mart <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")

genes.up <- results[, 1] %>% as.data.frame() %>% rownames_to_column() %>%
    magrittr::set_colnames(c("Gene", "DE")) %>% filter(DE ==
    1) %>% .[, 1]

entrez.up <- getBM(mart = mart, attributes = "entrezgene", filters = "ensembl_gene_id",
    values = genes.up, uniqueRows = TRUE)

genes.down <- results[, 1] %>% as.data.frame() %>% rownames_to_column() %>%
    magrittr::set_colnames(c("Gene", "DE")) %>% filter(DE ==
    -1) %>% .[, 1]
```

```
entrez.down <- getBM(mart = mart, attributes = "entrezgene",
    filters = "ensembl_gene_id", values = genes.down, uniqueRows = TRUE)

# Run enrichment analysis using clusterProfilers enricher
# function

H.enrich.up <- enricher(entrez.up[, 1], TERM2GENE = H)

H.enrich.down <- enricher(entrez.down[, 1], TERM2GENE = H)
```

```
dotplot(H.enrich.up)
```



```
dotplot(H.enrich.down)
```

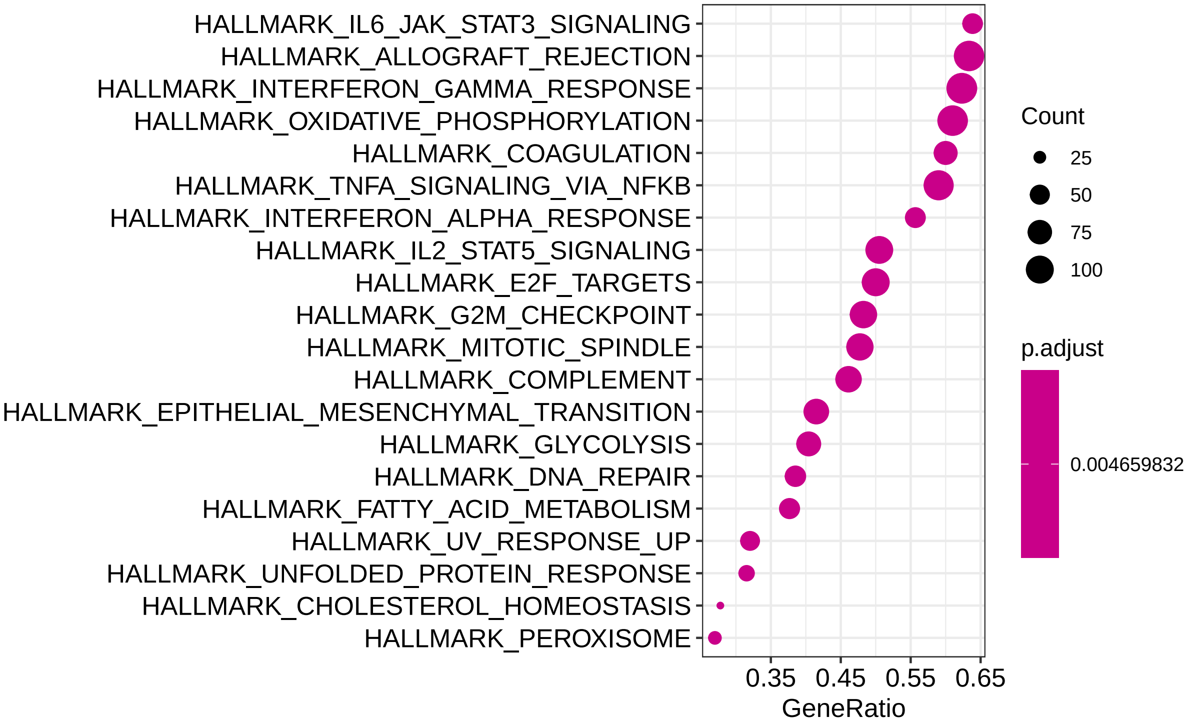


Perform Gene Set Enrichment Analysis on Ranked List of DE Genes

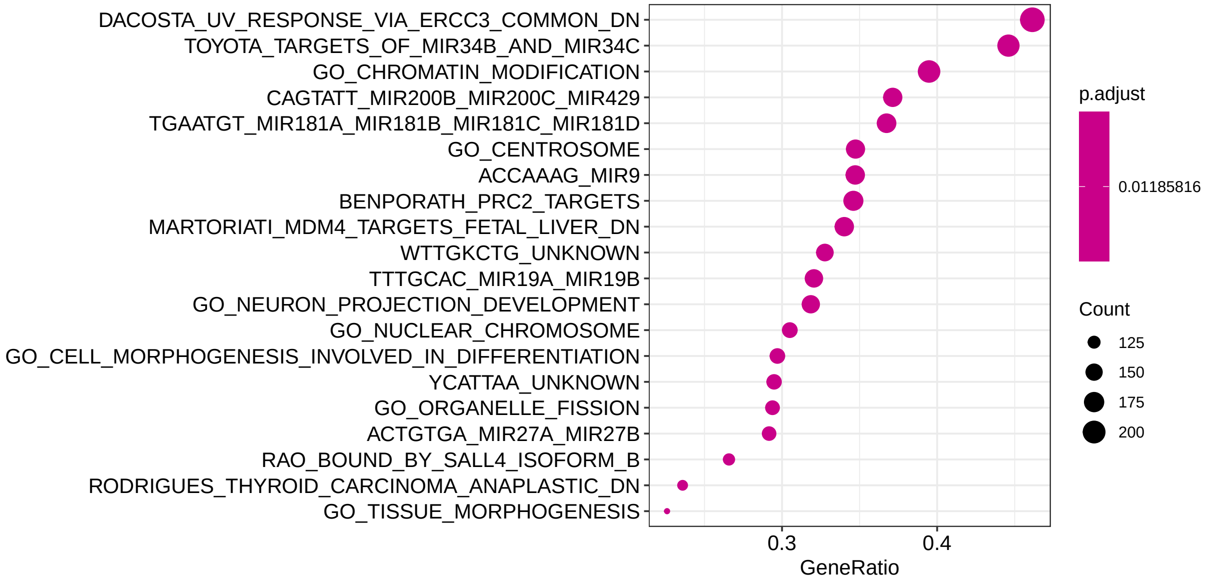
Here we will use the Hallmark and complete Gene Sets for the Analysis

```
# Create a ranked genelist for GSEA analysis

genelist <- annotated.results %>% # mutate(rank = ifelse(logFC > 0, 1/P.Value, -1/P.Value)) %>%
dplyr::select(entrezgene, t) %>% arrange(desc(t)) %>% as.data.frame()

geneList <- genelist[, 2]

names(geneList) <- as.character(genelist[, 1])

geneList <- sort(geneList, decreasing = TRUE)

GSEA.exp.H <- GSEA(geneList, TERM2GENE = H)

GSEA.exp.A <- GSEA(geneList, TERM2GENE = all)
```

```
dotplot(GSEA.exp.H, showCategory = 20)
```

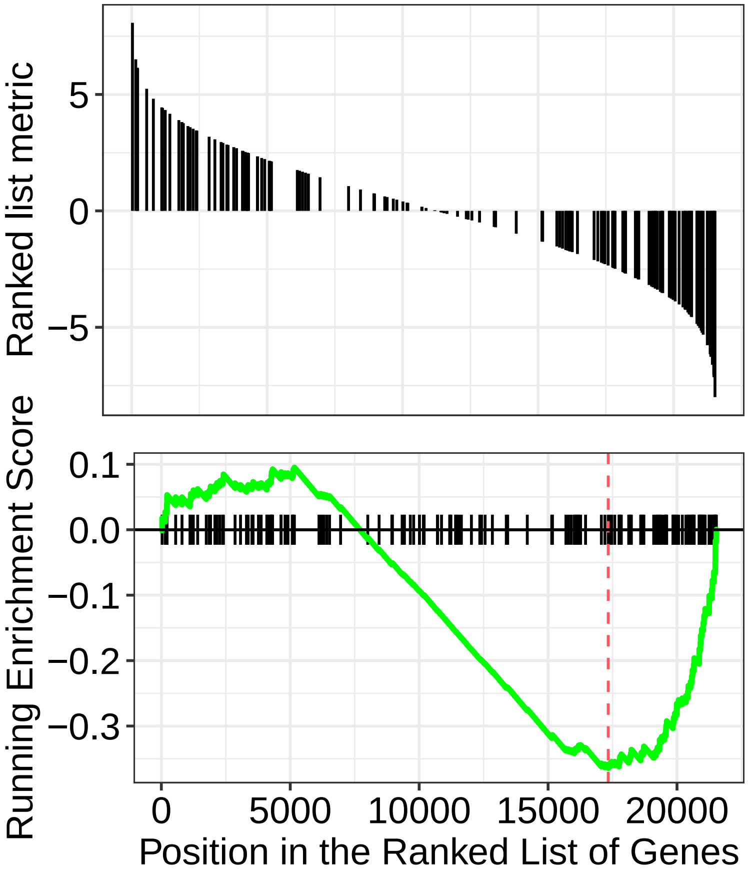


```
dotplot(GSEA.exp.A, showCategory = 20)
```



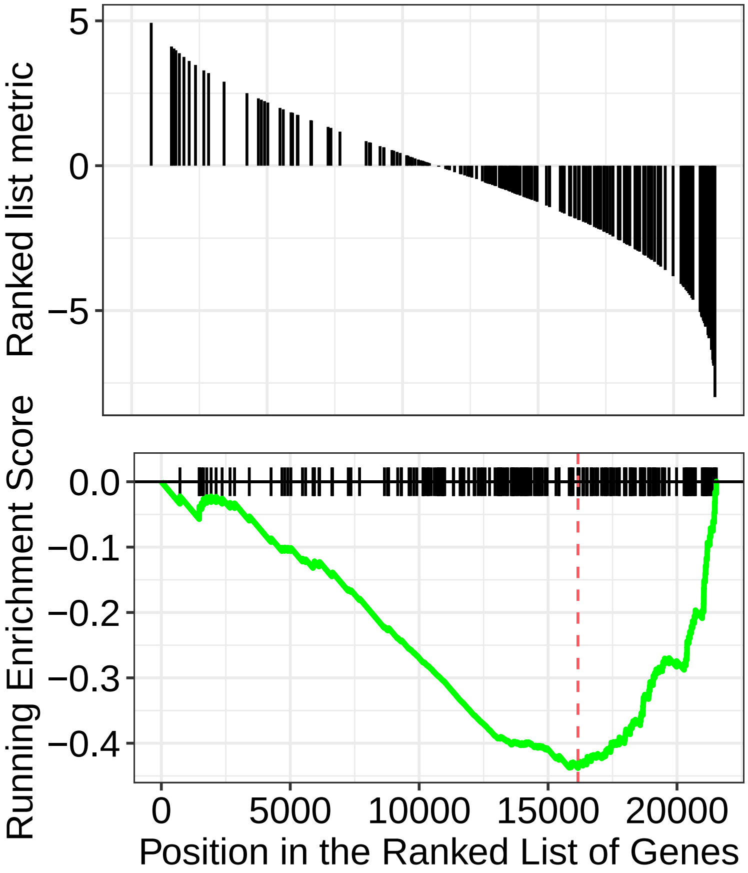
GSEA Plot for Hallmark DNA Repair

```
gseaplot(GSEA.exp.H, geneSetID = "HALLMARK_DNA_REPAIR")
```



GSEA Plot for EMT

```
gseaplot(GSEA.exp.H, "HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION")
```



Observations.

When we analyze differential gene expression in samples with high vs low NSD3 expression, we identified significantly more DE genes than the analysis of amplified samples. In the upregulated gene set we see a subtle enrichment in hedgehog signalling genes and targets of E2F transcription factor. Surprisingly, there was an enrichment in the downregulated genes in EMT, this includes a number of proteases. At this point is difficult to interpret the significance of these transcriptional changes and how they may relate to what I've seen in my experiments. Further research and experimentation is required and this data set may be offer further insights in the context of future results.

References

1. Weinstein JN, Collisson EA, Mills GB, et al. The Cancer Genome Atlas Pan-Cancer Analysis Project. Nature genetics. 2013;45(10):1113-1120. doi:10.1038/ng.2764.

2. Martin Morgan, Valerie Obenchain, Jim Hester and Hervé Pagès (2017). SummarizedExperiment: SummarizedExperiment container. R package version 1.8.1.

3. BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. Steffen Durinck, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma and Wolfgang Huber, Bioinformatics 21, 3439-3440 (2005).

4. Antonio Colaprico, Tiago Chedraoui Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais Sabedot, Tathiane Malta, Stefano M. Pagnotta, Isabella Castiglioni,Michele Ceccarelli, Gianluca Bontempi Houtan Noushmehr. TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data Nucleic Acids Research (05 May 2016) 44 (8): e71. (doi:10.1093/nar/gkv1507)

5. Mario Deng (2016). FirebrowseR: An 'API' Client for Broads 'Firehose' Pipeline. R package version 1.1.35.

6. Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

7. Guangchuang Yu, Li-Gen Wang, Yanyan Han and Qing-Yu He. clusterProfiler: an R package for comparing biological themes among gene clusters. OMICS: A Journal of Integrative Biology 2012, 16(5):284-287

8. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research 43(7), e47

9. Shian Su, Charity W. Law, Casey Ah-Cann, Marie-Liesse Asselin-Labat, Marnie E. Blewitt, Matthew E. Ritchie; Glimma: interactive graphics for gene expression analysis. Bioinformatics 2017 btx094. doi: 10.1093/bioinformatics/btx094

10. Stefan Milton Bache and Hadley Wickham (2014). magrittr: A Forward-Pipe Operator for R. R package version 1.5. https://CRAN.R-project.org/package=magrittr

11. Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139-140

ExpID-026