Dr.Shah Dengue NS5 and polyIC samples

Author: Amanda Everitt
Began: 8/12/2018
Finished: TBD

[Methods]

Raw data was processed using the Ion Torrent Suite Sever version 5.8.0 (Thermo Fisher). The coverageAnalysis plugin v5.8.0.8 was used to generate coverage and alignment quality information. Reads were aligned to the hg19_AmpliSeq_Transcriptome_v1.1 reference using default settings (tmap mapall -q 50000 -Y -u -o 2 stage1 map4). Gene expression was quantified with using the ampliSeqRNA plugin v5.8.0.3 with default settings. Genes with less than ten reads in more than 75% of the samples were removed. Filtered genes were normalized for sample library size before applying a principal component analysis (PCA) and hierarchical clustering to identify sample outliers. DESeq2 [cite] was used to identify differentially expressed genes (adj pval < 0.05); the script is included in supplementary information. Gene Ontology analysis was preformed using the goseq R-package [cite] using all expressed genes as a background.

Love, M.I., Huber, W., Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 Genome Biology 15(12):550 (2014)
Young, M.D., Wakefield, M.J., Smyth, G.K., Oshlack, A., Gene ontology analysis for RNA-seq: accounting for selection bias, Genome Biology, 11, 2, Feb 2010, R14

[Experimental Design]

- Test 1: GFP+polyIC vs GFP
- Test 2: NS5+polyIC vs NS5
- Test 3: NS5+polyIC vs GFP+polyIC
- Test 4: NS5 vs GFP

[Results at a Glance]

- TBD

Step 1: Set-up

```
knitr::opts_chunk$set(cache=TRUE, autodep=TRUE)
knitr::opts_knit$set(root.dir = "~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/")
#setwd("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/")

New_image_directory = "/Users/AEveritt/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/"
dir.create(New_image_directory, recursive = TRUE, showWarnings = FALSE)
```

Step 2: Load counts

```
suppressPackageStartupMessages(library(data.table))

cts <- read.delim("raw_data/absolute_counts_Chip21.xls", sep = "\t")
metadata <- read.delim("raw_data/Chip21_metadata.csv", sep = ",")
#setdiff(as.vector(metadata21$IonCode), colnames(cts21))
setnames(cts, old = as.vector(metadata$IonCode), new=as.vector(metadata$FullID))
to_remove <- metadata[substr(metadata$FullID, 1, 3) == "ex1", "FullID"] #remove IFNB1 data
metadata <- metadata[!metadata$FullID %in% to_remove,]
cts <- cts[, !colnames(cts) %in% to_remove]
```

Step 3: Remove lowly expressed genes

```
#Filter Data
dont_include <- c("Gene","Target","COSMIC_CGC_FLAG","NCBI_NAME","HGNC_SYMBOL_ACC","MIM_MORBID_DESCRIPTI
dim(cts)
```

```
## [1] 20866    16
```

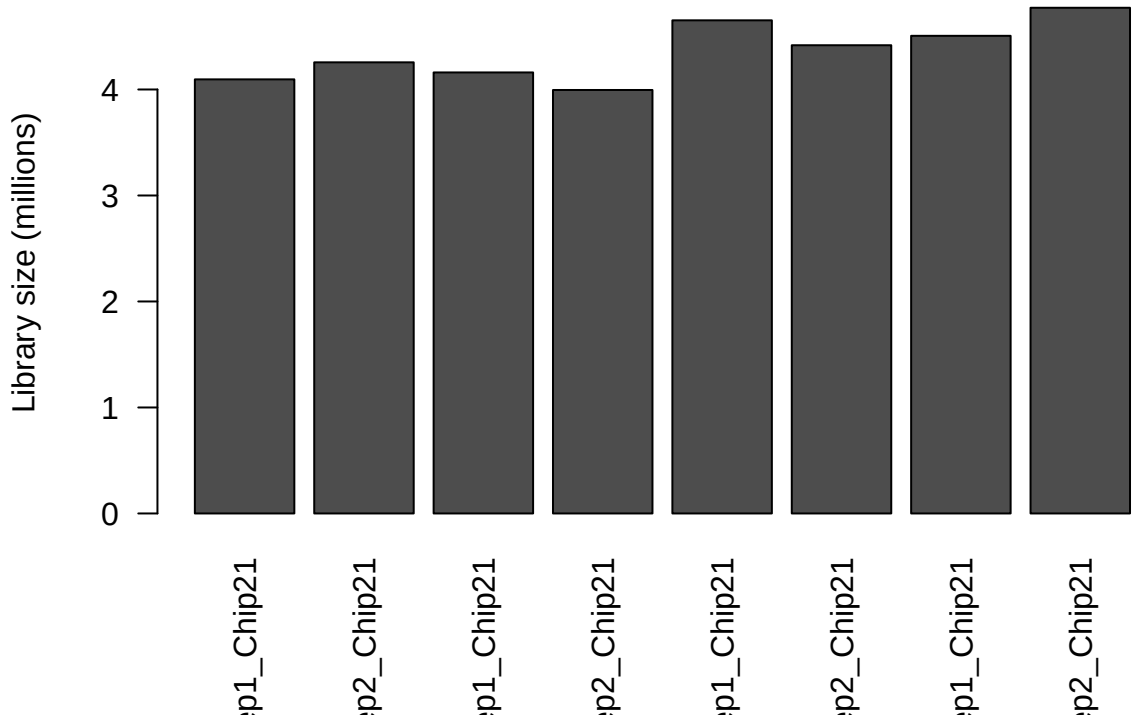
```
cts <- cts[rowSums(cts[, !colnames(cts) %in% dont_include] > 10) >= 2,] #filter counts that have don't
dim(cts)
```

```
## [1] 11364    16
```

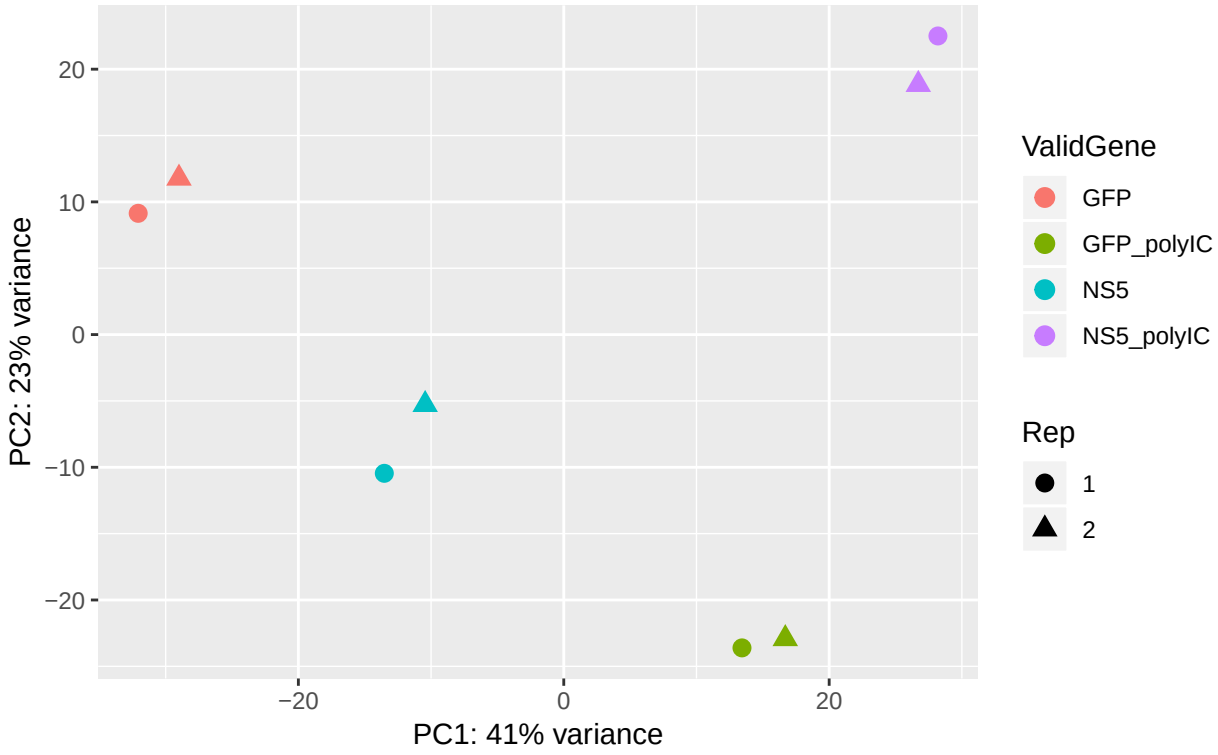
```
#dim(cts[duplicated(cts$Gene), ]) #5 non-unique genes
rownames(cts) = make.names(cts$Gene, unique=TRUE)
rownames(metadata) = make.names(metadata$FullID)

filtered_cts <- cts
cts <- cts[, !colnames(cts) %in% dont_include] #Remove non-numeric columns
```

Step 3: Normalization and outlier removal

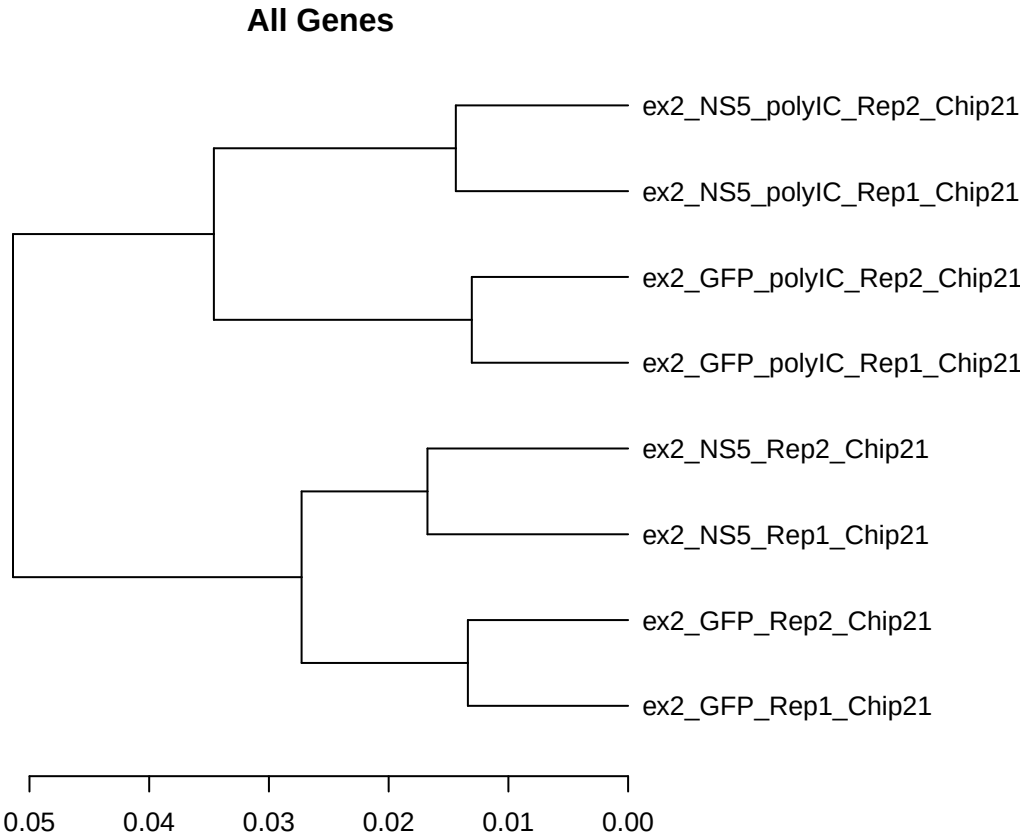
```
library(ggplot2)
library(preprocessCore)

#Distribution of Library Size
cts.libsize <- colSums(cts)*1e-6
barplot(t(as.data.frame(cts.libsize)), ylab="Library size (millions)", las=2)
```



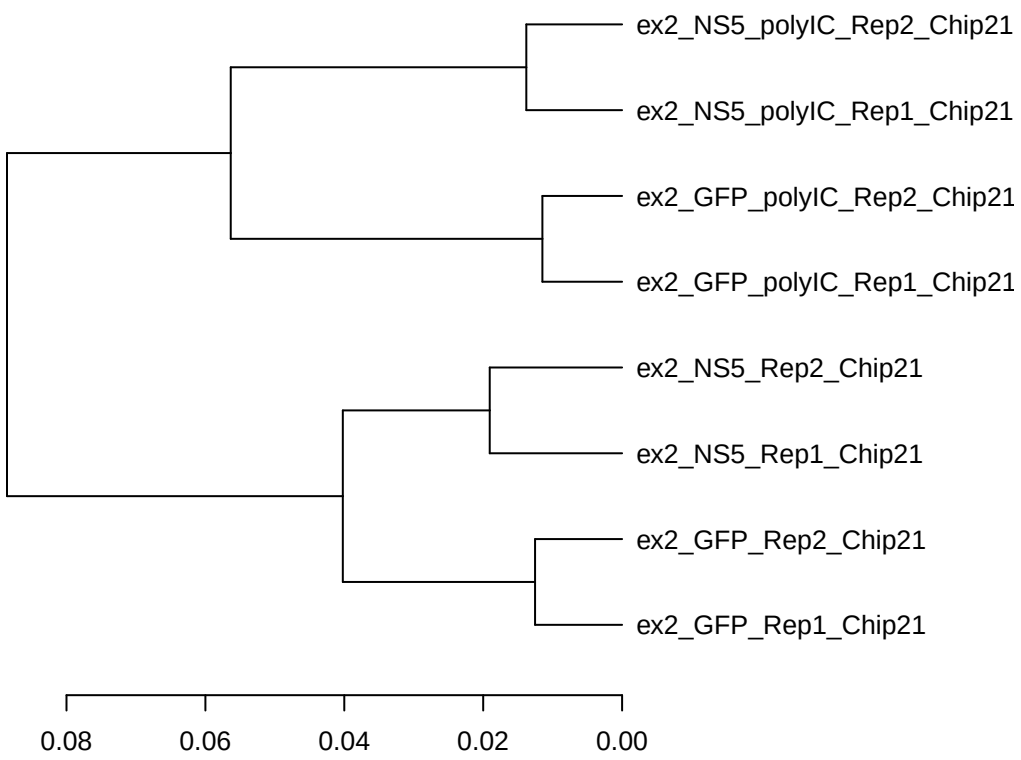

```
#Library Size normalization
cpm <- apply(cts,2, function(x) (x/sum(x))*1000000)
log.cpm <- log2(cpm + 1)

#Quantile Normalization
norm_counts <- normalize.quantiles(as.matrix(log.cpm), copy = TRUE)
colnames(norm_counts) <- colnames(log.cpm)
rownames(norm_counts) <- rownames(log.cpm)

#PCA
par(mfrow=c(1,1))
pca <- prcomp(t(norm_counts))
percentVar <- pca$sdev^2/sum(pca$sdev^2)
d <- data.frame(PC1 = pca$x[, 1], PC2 = pca$x[, 2])
e <- merge(d, metadata, all=TRUE, by="row.names")
e$Rep <- as.factor(e$Rep)
ggplot(data = e, aes_string(x = "PC1", y = "PC2", color = "ValidGene", shape="Rep")) +
        geom_point(size=3) +
        xlab(paste0("PC1: ", round(percentVar[1] * 100), "% variance")) +
        ylab(paste0("PC2: ", round(percentVar[2] * 100), "% variance")) +
        coord_fixed()
```



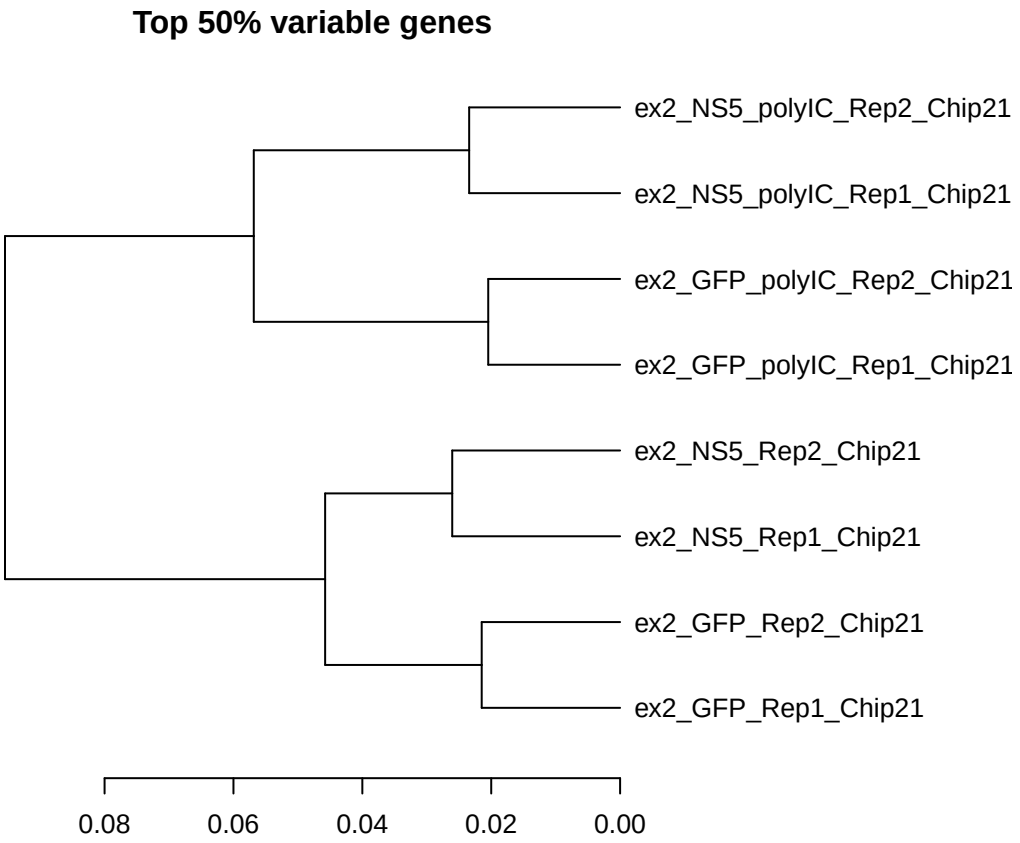

```
library(matrixStats)

#DENDOGRAMS
par(cex=0.8,mar = c(3,1,1,18))
  #-------Base------#
dend <- as.dendrogram(hclust(as.dist(1 - cor(((norm_counts)), use = "pa")),method = "complete"))
plot(dend,horiz=TRUE, main="All Genes")
```




```
#-------Top 50% expressed genes------#
union.top.express <- list()
for (i in colnames(norm_counts)){
  tmp <- as.data.frame(norm_counts[, colnames(norm_counts) == i, drop=F])
  colnames(tmp) <- "value"
  to_use <- rownames(tmp[tmp$value > quantile(tmp$value,prob=50/100), , drop=F])
  union.top.express <- union(union.top.express, to_use)
}
#length(union.top.express)

df <- norm_counts[rownames(norm_counts) %in% as.vector(union.top.express),]
dend <- as.dendrogram(hclust(as.dist(1 - cor((df), use = "pa")),method = "complete"))
plot(dend,horiz=TRUE, main= "Top 50% expressing genes")
```




```
#-------Top 50% variable genes------#
a<- as.data.frame(norm_counts)
a$var = as.vector(rowVars(norm_counts))
top.var <- rownames(a[a$var > quantile(a$var,prob=50/100), , drop=F])
df <- norm_counts[rownames(norm_counts) %in% as.vector(top.var),]
dend <- as.dendrogram(hclust(as.dist(1 - cor((df), use = "pa")),method = "complete"))
plot(dend,horiz=TRUE, main= "Top 50% variable genes")
```



Initial thoughts:

- Clean data, replicates cluster together and away from other treatments (lfcShrink should be fine here)
- No outliers
- No confounding factors to account for in model matrix

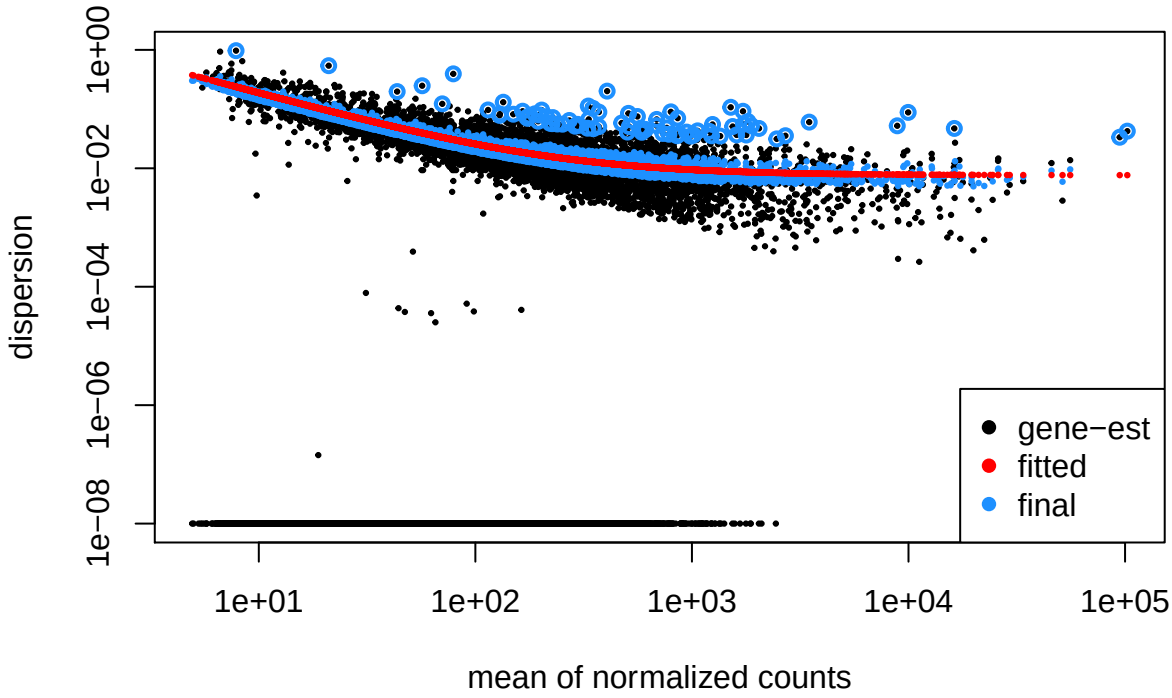
```
#Save Data
write.csv(filtered_cts, file=paste0(New_image_directory, "/count_matrix.csv"))
write.csv(as.data.frame(norm_counts), file=paste0(New_image_directory, "/norm_counts.csv"))
write.csv(metadata, file=paste0(New_image_directory, "/experimental_design.csv"))
```

Step 4: DESeq2

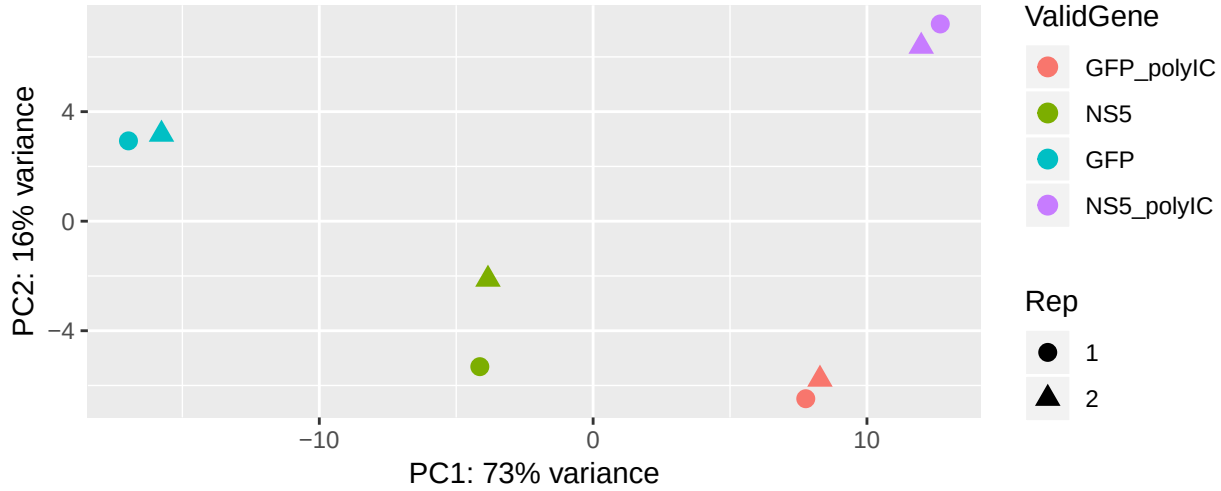
- Test 1: GFP+polyIC vs GFP
- Test 2: NS5+polyIC vs NS5
- Test 3: NS5+polyIC vs GFP+polyIC
- Test 4: NS5 vs GFP

```
suppressPackageStartupMessages(library(DESeq2))
results_list <- list()

#Make model matrix. No confounding factors to control for here
experimental.design <- metadata[order(match(rownames(metadata), colnames(cts))),]

#Test 1
experimental.design$ValidGene <- relevel(experimental.design$ValidGene, ref = "GFP")
dds <- DESeqDataSetFromMatrix(countData = cts, colData = experimental.design, design= ~ValidGene)
dds <- DESeq(dds)
#results_list[["GFP+polyIC_vs_GFP"]] <- results(dds, name="ValidGene_GFP_polyIC_vs_GFP")
#results_list[["NS5_vs_GFP"]] <- results(dds, name="ValidGene_NS5_vs_GFP")
results_list[["GFP+polyIC_vs_GFP"]] <- lfcShrink(dds, coef="ValidGene_GFP_polyIC_vs_GFP", type="apeglm")
results_list[["NS5_vs_GFP"]] <- lfcShrink(dds, coef="ValidGene_NS5_vs_GFP", type="apeglm")

#Test 2
experimental.design$ValidGene <- relevel(experimental.design$ValidGene, ref = "NS5")
dds <- DESeqDataSetFromMatrix(countData = cts, colData = experimental.design, design= ~ValidGene)
dds <- DESeq(dds)
#results_list[["NS5+polyIC_vs_NS5"]] <- results(dds, name="ValidGene_NS5_polyIC_vs_NS5")
results_list[["NS5+polyIC_vs_NS5"]] <- lfcShrink(dds, coef="ValidGene_NS5_polyIC_vs_NS5", type="apeglm")

#Test 3
experimental.design$ValidGene <- relevel(experimental.design$ValidGene, ref = "GFP_polyIC")
dds <- DESeqDataSetFromMatrix(countData = cts, colData = experimental.design, design= ~ValidGene)
dds <- DESeq(dds)
#results_list[["NS5+polyIC_vs_GFP+polyIC"]] <- results(dds, name="ValidGene_NS5_polyIC_vs_GFP_polyIC")
results_list[["NS5+polyIC_vs_GFP+polyIC"]] <- lfcShrink(dds, coef="ValidGene_NS5_polyIC_vs_GFP_polyIC",
```

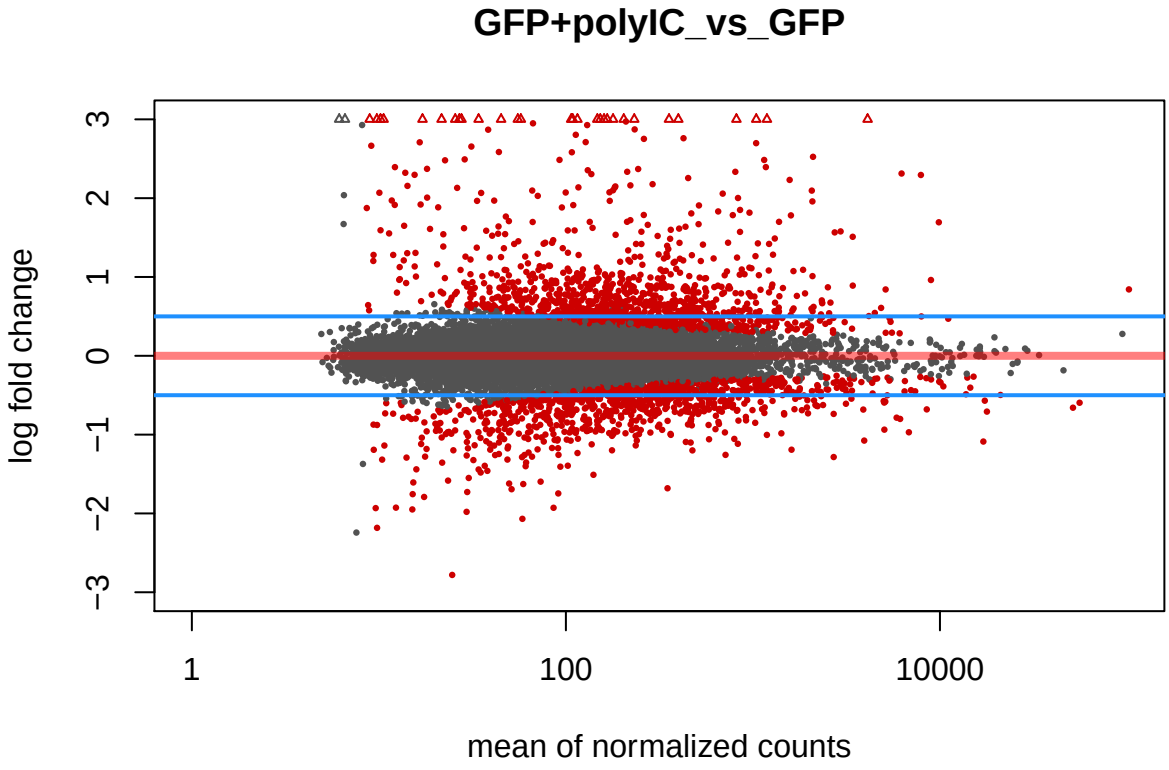
```
##PCA
plotDispEsts(dds)
```

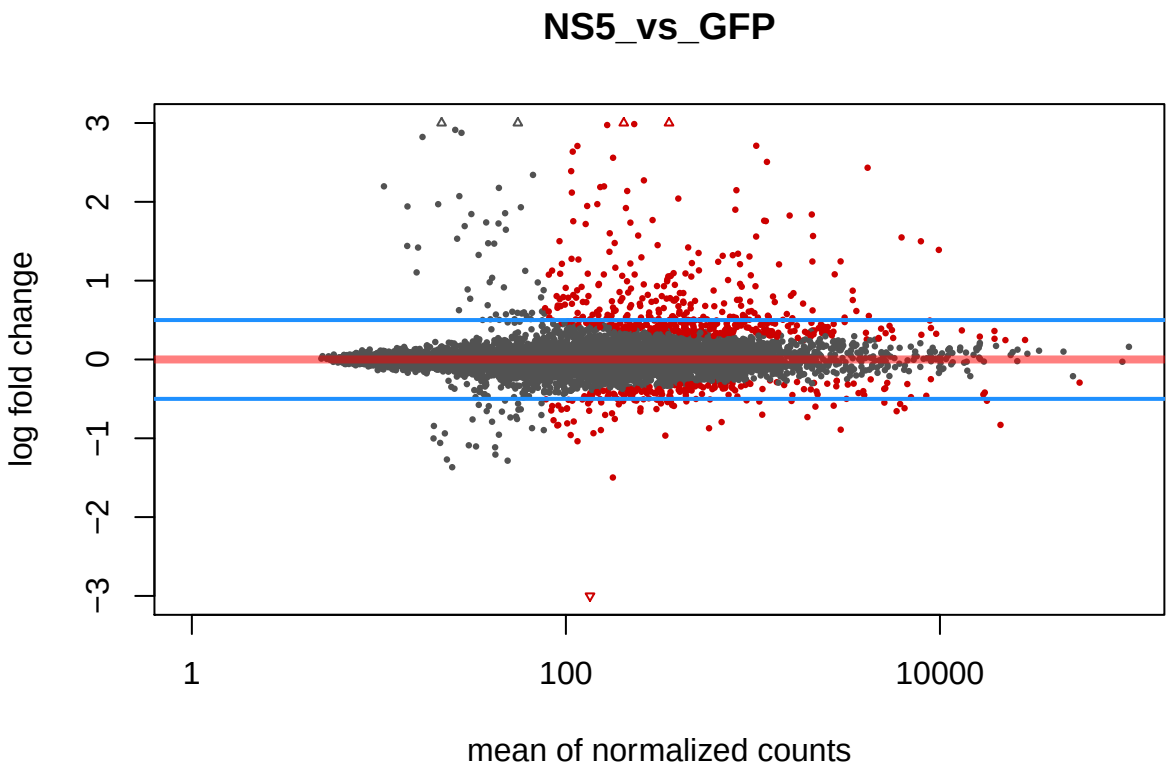


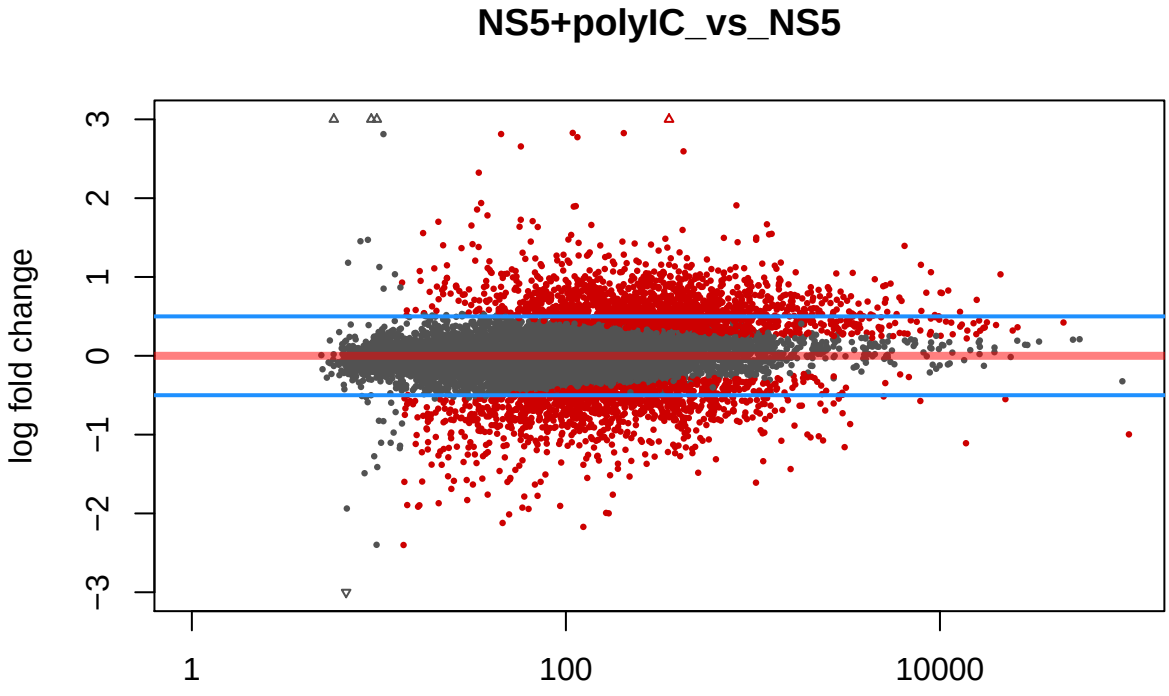

```
vsd <- vst(dds)

pcaData <- plotPCA(vsd, intgroup=c("ValidGene","Rep"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
pcaData$Rep <- as.factor(pcaData$Rep)
ggplot(pcaData, aes(PC1, PC2, color=ValidGene, shape=Rep)) +
    geom_point(size=3) +
    xlab(paste0("PC1: ",percentVar[1],"% variance")) +
    ylab(paste0("PC2: ",percentVar[2],"% variance")) +
    coord_fixed()
```

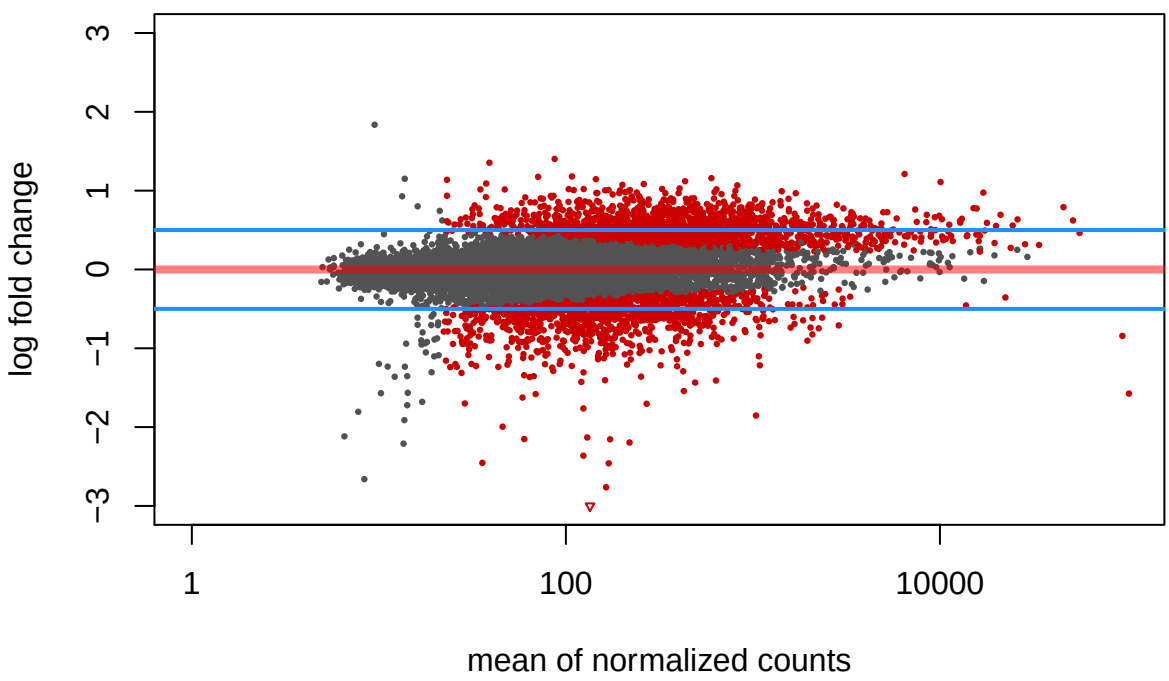



```
#par(mfrow=c(1,2))
drawLines <- function() abline(h=c(-.5,.5),col="dodgerblue",lwd=2)

for (i in names(results_list))
  { DESeq2::plotMA(results_list[[i]], main = i,xlim=c(1,1e5), ylim=c(-3,3)); drawLines() }
```


GFP+polyIC_vs_GFP
log fold change
3
2
1
0
−1
−2
−3
1
100
10000
mean of normalized counts


NS5_vs_GFP
log fold change
3
2
1
0
−1
−2
−3
1
100
10000
mean of normalized counts

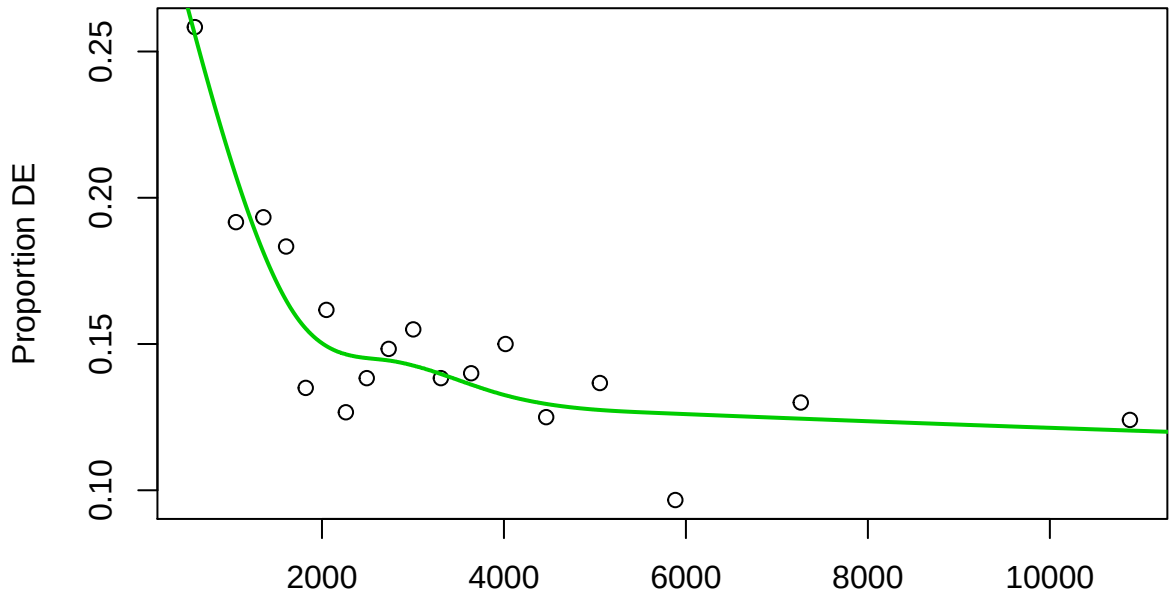

NS5+polyIC_vs_NS5
log fold change
3
2
1
0
−1
−2
−3
1
100
10000
mean of normalized counts


NS5+polyIC_vs_GFP+polyIC
log fold change
3
2
1
0
−1
−2
−3
1
100
10000
mean of normalized counts

Step 5: Output Significant Results

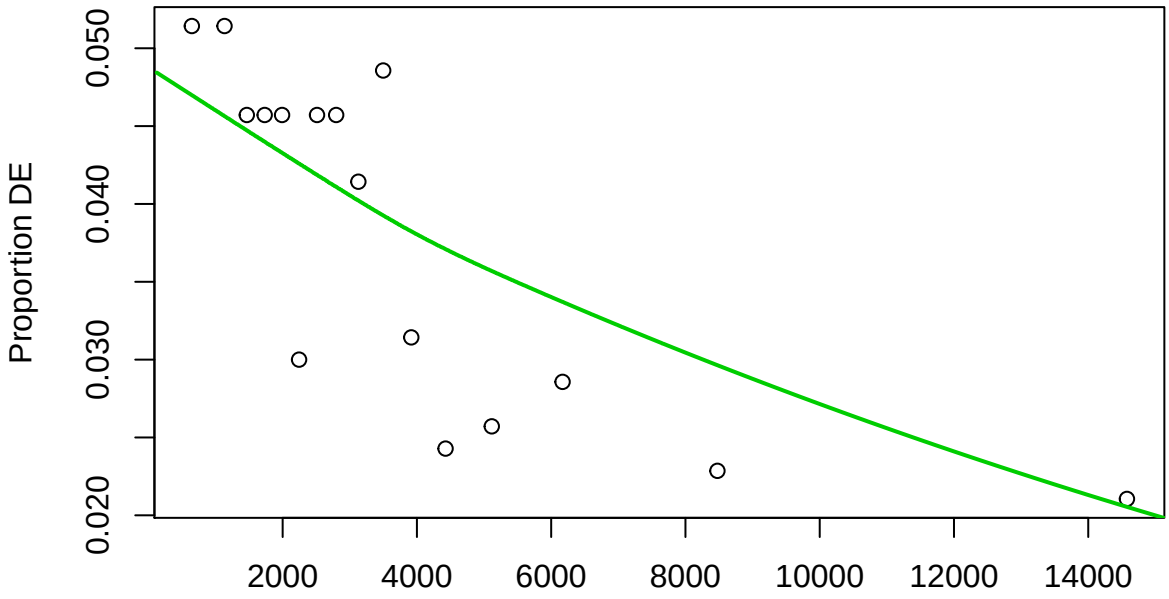
```
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(plyr))
GOoverlap = data.frame(category= character(), file=character(),term=character(),ontology=character())
GENEoverlap = data.frame(Gene=character(), File=character())

for (i in names(results_list)){
  print(i)
  DE_list <- quick_output(results_list[[i]], i, filtered_cts)

  #Make list of genes which overlap
  temp <- data.frame(Gene=rownames(DE_list),
                     File=rep(i, length(rownames(DE_list))),
                     Direction=ifelse(DE_list$log2FoldChange >0 , 'UP', 'DOWN'))
  GENEoverlap <- merge(x=GENEoverlap, y=temp, all.x=TRUE, all.y=TRUE)

  #GO analysis
  returned_object <- run_GOpathway_analysis(
      DEgenes=rownames(DE_list),
      backgroundGenes=rownames(filtered_cts),
      genomeBuild="hg19", geneIdentifier="geneSymbol",cutoff=0.05,
      outputFileFullPath=paste0(New_image_directory, "/GO_", i, ".csv"))
  cat(paste(nrow(returned_object), " :Number of GO terms\n\n"))
  GOoverlap <- rbind(GOoverlap, returned_object[,c("term", "ontology")])
}
```

```
## [1] "GFP+polyIC_vs_GFP"
## GFP+polyIC_vs_GFP
##  11364 total
## 1680 have adj pval < 0.05
## 342 have abs(logFC) > 1
```



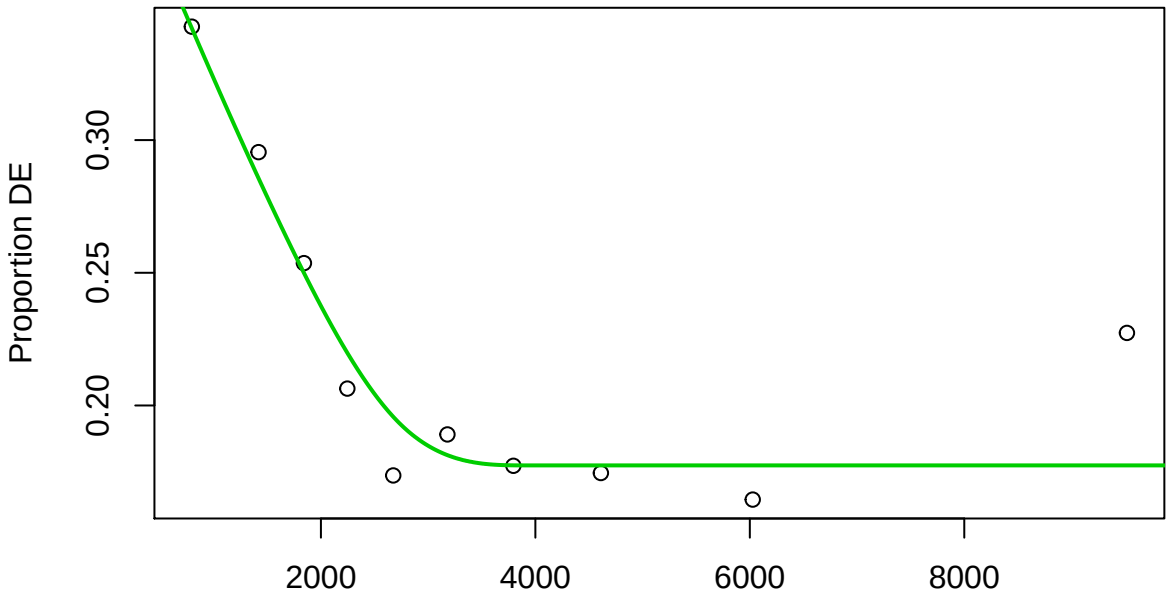
Biased Data in 600 gene bins.

```
## 891  :Number of GO terms
##
## [1] "NS5_vs_GFP"
## NS5_vs_GFP
##  11364 total
## 429 have adj pval < 0.05
## 79 have abs(logFC) > 1
```



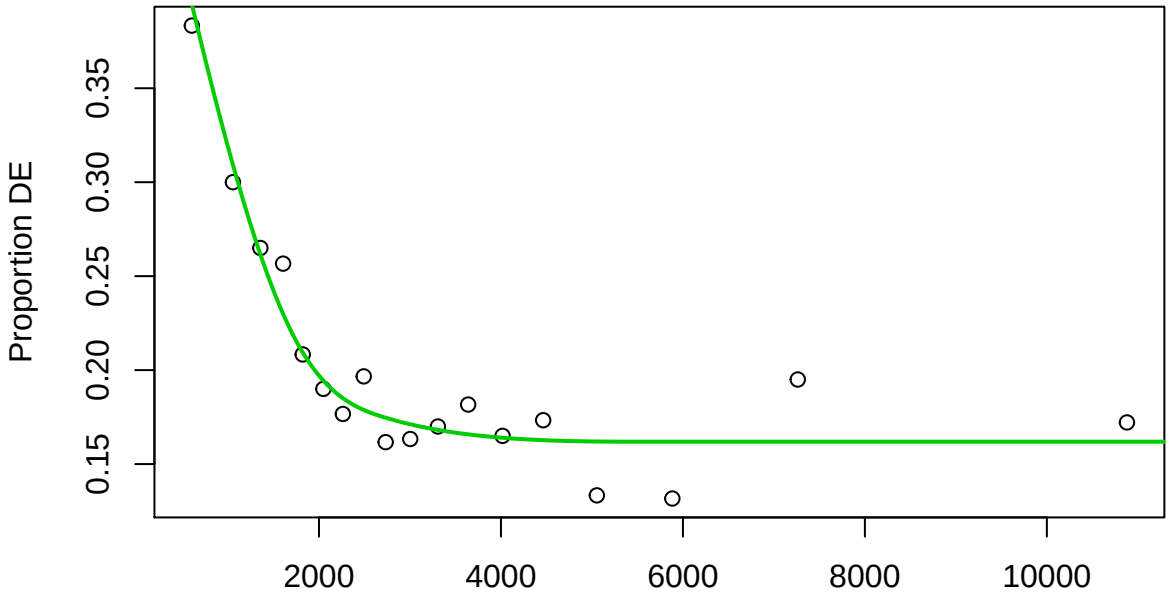
Biased Data in 700 gene bins.

```
## 632  :Number of GO terms
##
## [1] "NS5+polyIC_vs_NS5"
## NS5+polyIC_vs_NS5
##  11364 total
## 2457 have adj pval < 0.05
## 300 have abs(logFC) > 1
```



Biased Data in 1100 gene bins.

```
## 759  :Number of GO terms
##
## [1] "NS5+polyIC_vs_GFP+polyIC"
## NS5+polyIC_vs_GFP+polyIC
##  11364 total
## 2258 have adj pval < 0.05
## 110 have abs(logFC) > 1
```



Biased Data in 600 gene bins.

```
## 760  :Number of GO terms
```

```
GOoutput <- GOoverlap %>% group_by(ontology,term) %>% dplyr::summarise(count=n()) %>% arrange(desc(coun
write.csv(GOoutput, paste0(New_image_directory,"/Overlapping_GOpathways.csv"),row.names = FALSE)

GENEoverlap <- ddply(GENEoverlap, .(Gene,Direction), dplyr::summarize, Count=length(unique(File)), File
write.csv(GENEoverlap, paste0(New_image_directory,"/Overlapping_Gene.csv"),row.names = FALSE)
```

[Summary so far]

- Test 1: GFP+polyIC vs GFP
- 1680 DEX, 342 hcDEX
- 891 GO
- Test 2: NS5+polyIC vs NS5
- 2457 DEX, 300 hcDEX
- 759 GO
- Test 3: NS5+polyIC vs GFP+polyIC
- 2258 DEX, 110 hcDEX
- 760 GO
- Test 4: NS5 vs GFP
- 429 DEX, 79 hcDEX
- 632 GO

```
polyic.gfp <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/DEX_GFP+polyIC_
ns5_vs_gfp <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/DEX_NS5_vs_GFP.
polyic.ns5 <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/DEX_NS5+polyIC_
polyic.ns5_vs_gfp <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/DEX_NS5+

filtered_cts <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/count_matrix.
```

First, I wanted to see how similar the polyIC treatment effect was in GFP and NS5 samples.

```
length(polyic.gfp$X)
```

```
## [1] 1680
```

```
length(polyic.ns5$X)
```

```
## [1] 2457
```

```
length(intersect(polyic.gfp$X, polyic.ns5$X)) #intersect
```

```
## [1] 648
```

```
intersect_up <- intersect(polyic.gfp[polyic.gfp$log2FoldChange > 0,"X"],
          polyic.ns5[polyic.ns5$log2FoldChange > 0,"X"])
intersect_down <- intersect(polyic.gfp[polyic.gfp$log2FoldChange < 0,"X"],
          polyic.ns5[polyic.ns5$log2FoldChange < 0,"X"])

same_direction <- union(intersect_up, intersect_down)
length(same_direction)
```

```
## [1] 490
```

```
#Is this significant
#For hypergeometric
#q= intersect
#m= treatment 1 DE
#n = all exp - treatment 1 DE
#k = treatment 2 DE
phyper(q= length(same_direction),
```

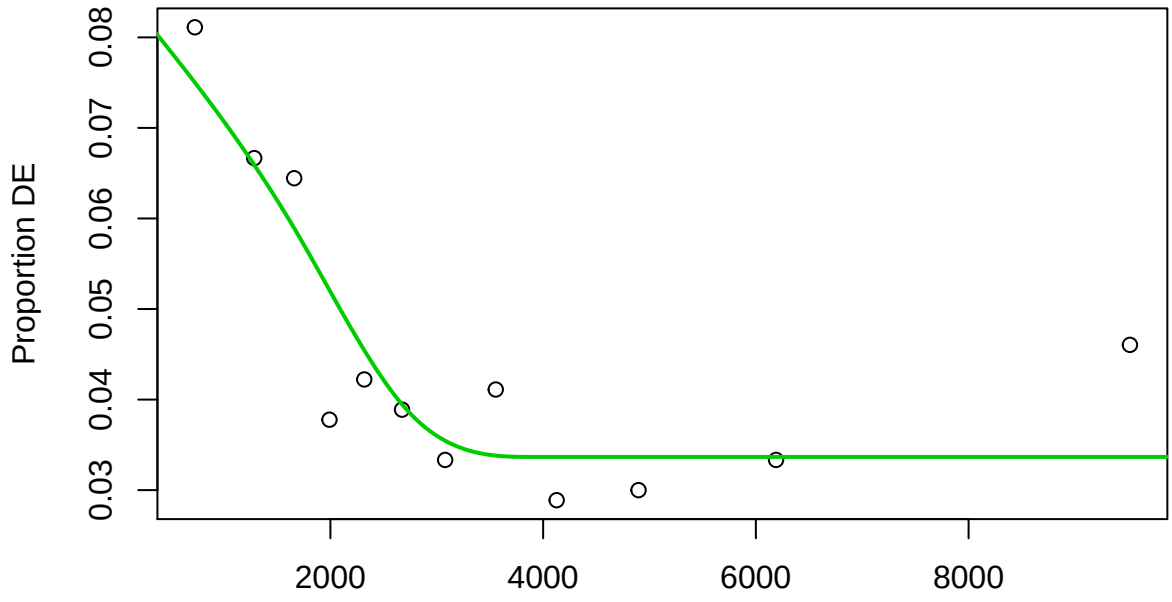
```
       m = length(polyic.gfp$X),
       n = nrow(filtered_cts)-length(polyic.gfp$X),
       k = length(polyic.ns5$X),
       lower.tail=FALSE
)
```

```
## [1] 1.066579e-15
```

Are these 490 enriched for any certain pathways?

```
GO.output <- run_GOpathway_analysis(
      DEgenes=same_direction,
      backgroundGenes=rownames(filtered_cts),
      genomeBuild="hg19", geneIdentifier="geneSymbol",cutoff=0.05,
      outputFileFullPath=paste0(New_image_directory, "/GO_overlap490.csv"))
```

```
## Loading hg19 length data...
## Fetching GO annotations...
## For 1231 genes, we could not find any categories. These genes will be excluded.
## To force their use, please run with use_genes_without_cat=TRUE (see documentation).
## This was the default behavior for version 1.15.1 and earlier.
## Calculating the p-values...
## 'select()' returned 1:1 mapping between keys and columns
```



Biased Data in 900 gene bins.

Okay, now how do these change in the NS5+polyIC vs GFP+polyIC

```
length(polyic.ns5_vs_gfp$X)
```

```
## [1] 2258
```

```
length(intersect(polyic.ns5_vs_gfp$X, same_direction))
```

```
## [1] 143
```

```
length(intersect(intersect_up, polyic.ns5_vs_gfp[polyic.ns5_vs_gfp$log2FoldChange > 0,"X"]))
```

```
## [1] 107
```

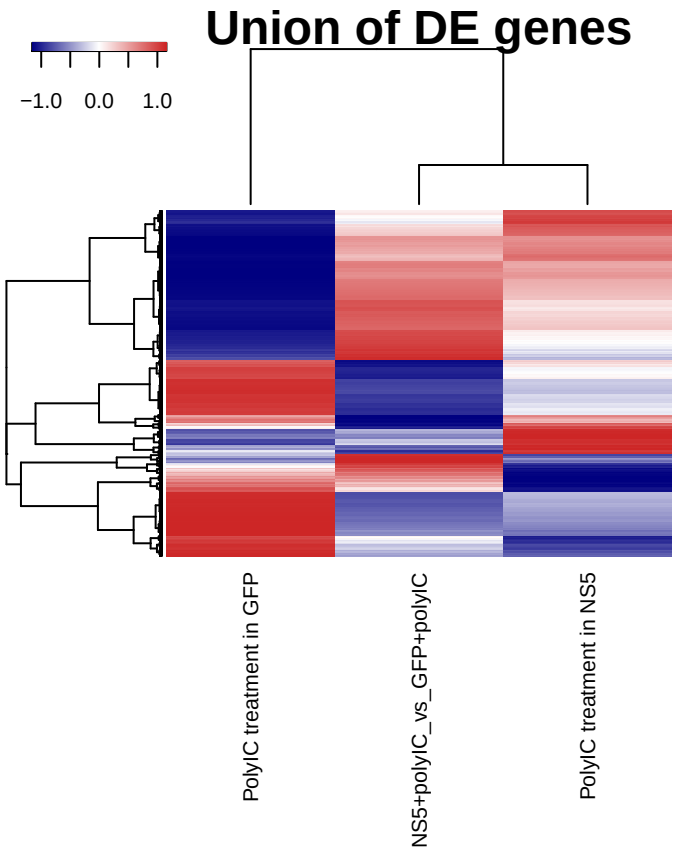
```
length(intersect(intersect_down, polyic.ns5_vs_gfp[polyic.ns5_vs_gfp$log2FoldChange < 0,"X"]))
```

```
## [1] 30
```

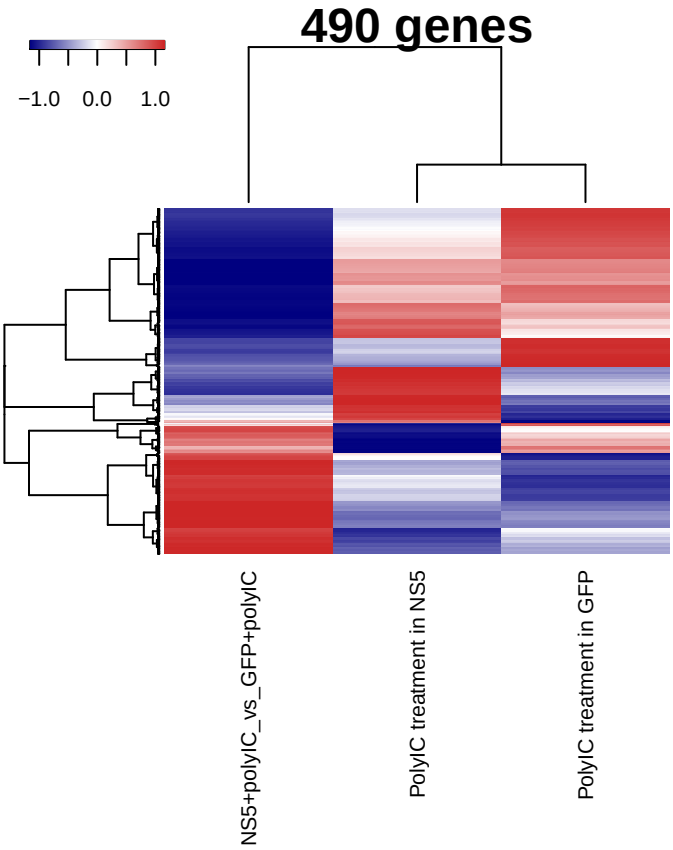
Heatmap to plot 1. Union of all DE genes 2. These 490 genes

```
genes.490 = same_direction
genes.union = union(union(polyic.gfp$X, polyic.ns5$X), polyic.ns5_vs_gfp$X)
```

```
library(heatmap3)
res.polyic.gfp <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/results_GFP
res.polyic.ns5 <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/results_NS5
res.ns5IC.gfpIC <- read.csv("~/projects/AmpliSeq_Analysis/07_dengue_chip20.21/results/polyIC/results_NS


tmp <- merge(x=res.polyic.gfp[, 2,drop=F], y=res.polyic.ns5[,2,drop=F], by="row.names", all=T)
my.log2df <- merge(x=res.ns5IC.gfpIC[, 2,drop=F], y=tmp, by.x="row.names", by.y="Row.names", all=T)
colnames(my.log2df) <- c("GeneID",
                  "NS5+polyIC_vs_GFP+polyIC",
                  "PolyIC treatment in GFP",
                  "PolyIC treatment in NS5")

heatmap3(my.log2df[my.log2df$GeneID %in% genes.union, c(2:4)],
         scale= "row", cexCol = 0.8,margins=c(10,5),labRow = NA,
         main="Union of DE genes")
```




```
heatmap3(my.log2df[my.log2df$GeneID %in% genes.490, c(2:4)],
         scale= "row", cexCol = 0.8,margins=c(10,5),labRow = NA,
         main="490 genes")
```



Gene Ontology Plot by -log10(over represented pval)

- TO ADD

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-apple-darwin17.2.0 (64-bit)
## Running under: macOS High Sierra 10.13.3
##
## Matrix products: default
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions
## LAPACK: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versio
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] heatmap3_1.1.1             bindrcpp_0.2.2
##  [3] org.Hs.eg.db_3.5.0         GO.db_3.5.0
##  [5] AnnotationDbi_1.40.0       goseq_1.30.0
##  [7] geneLenDataBase_1.14.0     BiasedUrn_1.07
##  [9] plyr_1.8.4                 dplyr_0.7.6
## [11] DESeq2_1.18.1              SummarizedExperiment_1.8.1
## [13] DelayedArray_0.4.1         Biobase_2.38.0
## [15] GenomicRanges_1.30.3       GenomeInfoDb_1.14.0
## [17] IRanges_2.12.0             S4Vectors_0.16.0
## [19] BiocGenerics_0.24.0        matrixStats_0.54.0
## [21] preprocessCore_1.40.0      ggplot2_3.0.0
## [23] data.table_1.11.4
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-137            bitops_1.0-6
##  [3] bit64_0.9-7             httr_1.3.1
##  [5] RColorBrewer_1.1-2      progress_1.2.0
##  [7] rprojroot_1.3-2         tools_3.4.3
##  [9] backports_1.1.2         R6_2.2.2
## [11] rpart_4.1-13            mgcv_1.8-23
## [13] Hmisc_4.1-1             DBI_1.0.0
## [15] lazyeval_0.2.1          colorspace_1.3-2
## [17] nnet_7.3-12             withr_2.1.2
## [19] tidyselect_0.2.4        gridExtra_2.3
## [21] prettyunits_1.0.2       RMySQL_0.10.14
## [23] bit_1.1-14              compiler_3.4.3
## [25] htmlTable_1.11.2        rtracklayer_1.38.3
## [27] scales_0.5.0            checkmate_1.8.5
## [29] genefilter_1.60.0       stringr_1.3.0
## [31] digest_0.6.15           Rsamtools_1.30.0
## [33] foreign_0.8-69          rmarkdown_1.9
## [35] XVector_0.18.0          base64enc_0.1-3
## [37] pkgconfig_2.0.1         htmltools_0.3.6
## [39] htmlwidgets_1.2         rlang_0.2.1
## [41] rstudioapi_0.7          RSQLite_2.1.0
## [43] bindr_0.1.1             BiocParallel_1.12.0
```

```
## [45] acepack_1.4.1           RCurl_1.95-4.11
## [47] magrittr_1.5            GenomeInfoDbData_1.0.0
## [49] Formula_1.2-3           Matrix_1.2-14
## [51] Rcpp_0.12.18            munsell_0.5.0
## [53] stringi_1.2.4           yaml_2.1.19
## [55] zlibbioc_1.24.0         grid_3.4.3
## [57] blob_1.1.1              crayon_1.3.4
## [59] lattice_0.20-35         Biostrings_2.46.0
## [61] splines_3.4.3           GenomicFeatures_1.30.3
## [63] annotate_1.56.2         hms_0.4.2
## [65] locfit_1.5-9.1          knitr_1.20
## [67] pillar_1.2.1            fastcluster_1.1.25
## [69] codetools_0.2-15        geneplotter_1.56.0
## [71] biomaRt_2.34.2          XML_3.98-1.12
## [73] glue_1.3.0              evaluate_0.10.1
## [75] latticeExtra_0.6-28     gtable_0.2.0
## [77] purrr_0.2.5             assertthat_0.2.0
## [79] xtable_1.8-2            survival_2.42-3
## [81] tibble_1.4.2            GenomicAlignments_1.14.2
## [83] memoise_1.1.0           cluster_2.0.7-1
```