Preliminary Analysis

```
library(tidyverse)
library(edgeR)
library(limma)
library(ggplot2)
library(dplyr)
library(gplots)
#
# For PDF export:
# install.packages("devtools")
library(devtools)
# install_version("rmarkdown",version=1.8)
```

Content

---

1 Host expression

Loading data

Read in the host count matrix and convert to log2cpm

```
host_counts <- read_tsv("data/MATRIX_Counts-FINAL-Host.txt")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   EntrezGeneID = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
host_samples <- read_tsv("data/SampleInfo_Host-STAR.txt")
```

```
## Parsed with column specification:
## cols(
##   FileName = col_character(),
##   SampleName = col_character(),
##   CellType = col_character(),
##   Strain = col_character()
```

```
## )
```

```
host_annotations <- read_tsv("data/Aten_diamond_annot_20190117_1327.txt")
```

```
## Parsed with column specification:
## cols(
##   EntrezGeneID = col_character(),
##   Annotation = col_character()
## )
```

```
host_exon_annnotations <- read_tsv("data/Host-Gene-Annotation.txt")
```

```
## Parsed with column specification:
## cols(
##   EntrezGeneID_0 = col_character()
## )
```

```
# Exons
host_DGE <- DGEList(host_counts[, -1], genes = host_counts[,
    1] %>% bind_cols(host_exon_annnotations),
    samples = host_samples)

filt <- filterByExpr(host_DGE, design = model.matrix(~Strain,
    data = host_DGE$samples), min.count = 40)

host_filtered1 <- host_DGE[filt, , keep.lib.sizes = F]

host_filtered2 <- calcNormFactors(host_filtered1)

# run limma voom
host_v <- voom(host_filtered2, design = model.matrix(~Strain,
    data = host_filtered2$samples), plot = F)
```

Collate to gene level

Might be better for some uses to look at whole gene level expression initally before drilling down to exon level data.

```
gene_host_counts <- host_counts %>% separate(EntrezGeneID,
    c("gene", "exon"), sep = "\\.exon") %>%
    group_by(gene) %>% summarise_if(is.numeric,
    sum)

# check that host_annotations and genes
# IDs match.. needs to be all TRUE events
table((host_annotations[, 1] == (gene_host_counts[,
    1])))
```

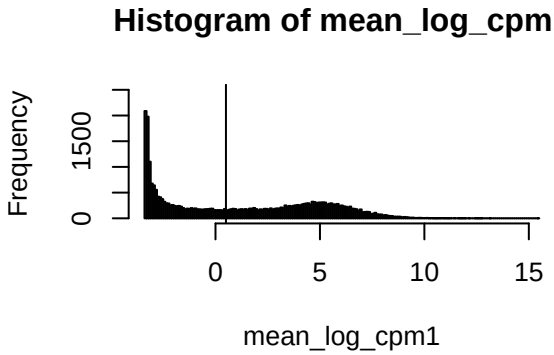
```
##
##  TRUE
## 30327
```

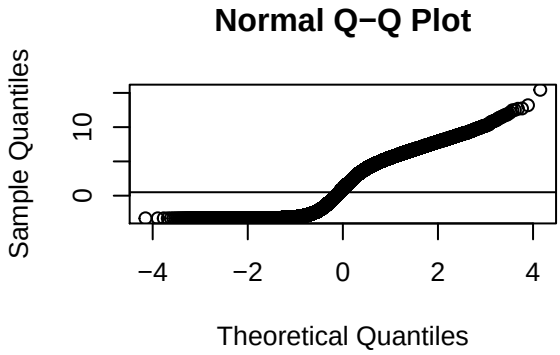
```
# then bind host_annotations to gene list
gene_host_DGE <- DGEList(gene_host_counts[,
    -1], genes = gene_host_counts[, 1] %>%
    bind_cols(host_annotations[, -1]), samples = host_samples)
```

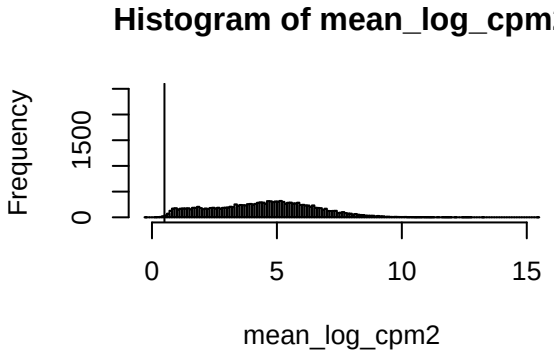
```r
# check that annotation table and
# gene_host_DGE match.. needs to be all
# TRUE events. Two columns to match
table((gene_host_DGE$genes == (host_annotations)))
```

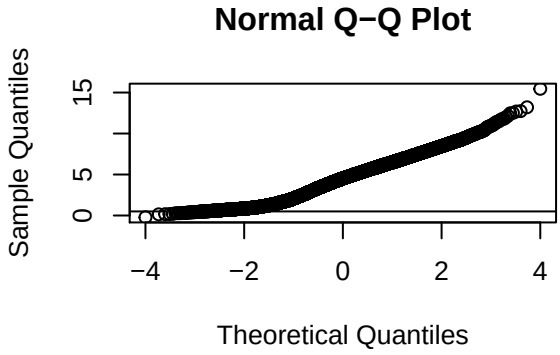
```
##
##  TRUE
## 59449
```

```r
gene_host_filt <- filterByExpr(gene_host_DGE,
    design = model.matrix(~Strain, data = gene_host_DGE$samples),
    min.count = 40)

gene_host_filtered1 <- gene_host_DGE[gene_host_filt,
    , keep.lib.sizes = F]

gene_host_filtered2 <- calcNormFactors(gene_host_filtered1)

# Check filter cut off before and after
par(mfrow = c(2, 2))
mean_log_cpm1 <- aveLogCPM(gene_host_DGE$counts)
filter_threshold <- 0.5
hist(mean_log_cpm1, breaks = 200, ylim = c(0,
    2500))
abline(v = filter_threshold)
qqnorm(mean_log_cpm1)
abline(h = filter_threshold)
#
mean_log_cpm2 <- aveLogCPM(gene_host_filtered1$counts)
filter_threshold <- 0.5
hist(mean_log_cpm2, breaks = 200, ylim = c(0,
    2500))
abline(v = filter_threshold)
qqnorm(mean_log_cpm2)
abline(h = filter_threshold)
```





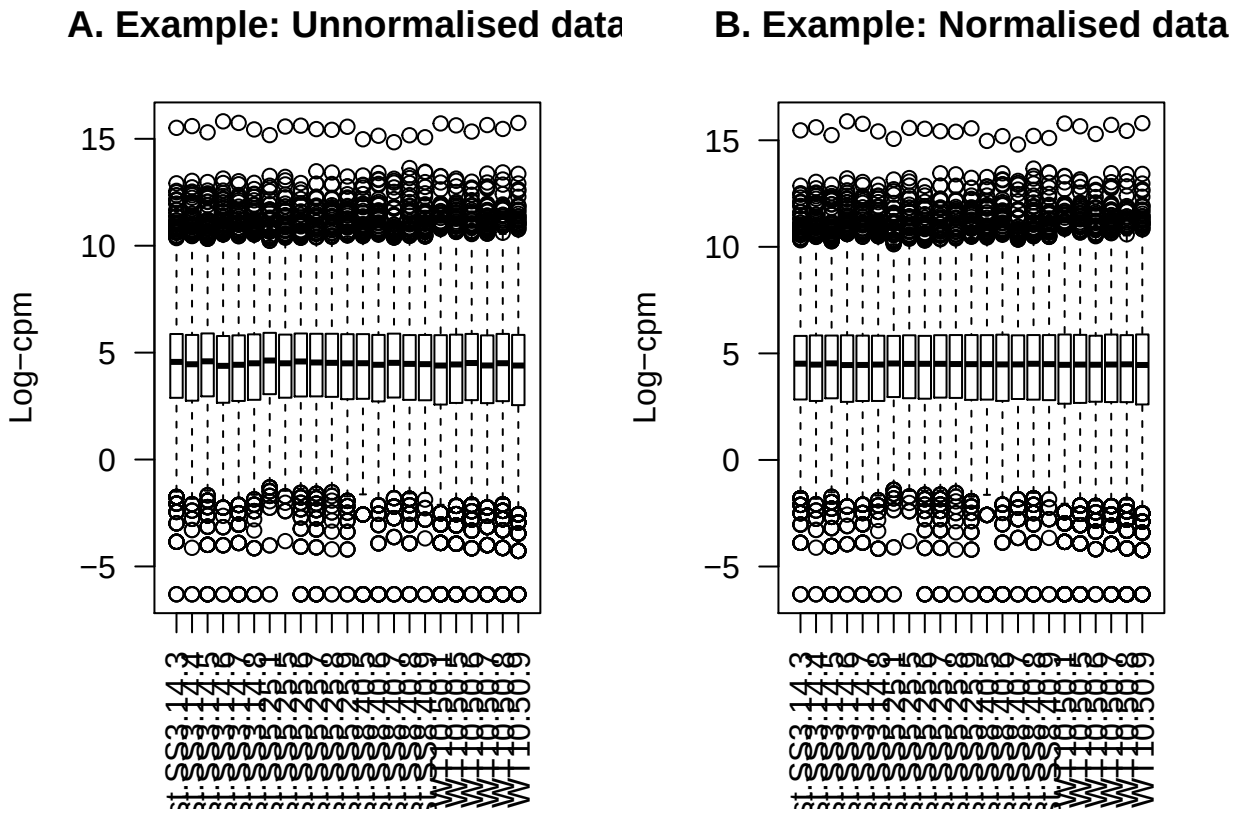





```
# Check normalised and unnormalised data
par(mfrow = c(1, 2))
host_lcpm1 <- cpm(gene_host_filtered1, log = TRUE)
boxplot(host_lcpm1, las = 2, main = "")
title(main = "A. Example: Unnormalised data",
    ylab = "Log-cpm")
#
gene_host_filtered2$samples$norm.factors
```

```
##  [1] 1.0350153 0.9890877 1.0436984 0.9516820 0.9815912 1.0175096 1.0752830
##  [8] 0.9958163 1.0530484 1.0194929 1.0152048 1.0020292 1.0060431 0.9650482
## [15] 1.0276482 0.9733857 0.9765775 0.9539049 0.9800737 1.0322795 0.9471989
## [22] 1.0126325 0.9590868
```

```
host_lcpm2 <- cpm(gene_host_filtered2, log = TRUE)
boxplot(host_lcpm2, las = 2, main = "")
title(main = "B. Example: Normalised data",
    ylab = "Log-cpm")
```




```
dev.off()
```

```
## null device
##           1
```

```
# check that data is still all matching
# up
table(host_samples$SampleName == colnames(gene_host_filtered2))
```

```
##
## TRUE
##   23
```

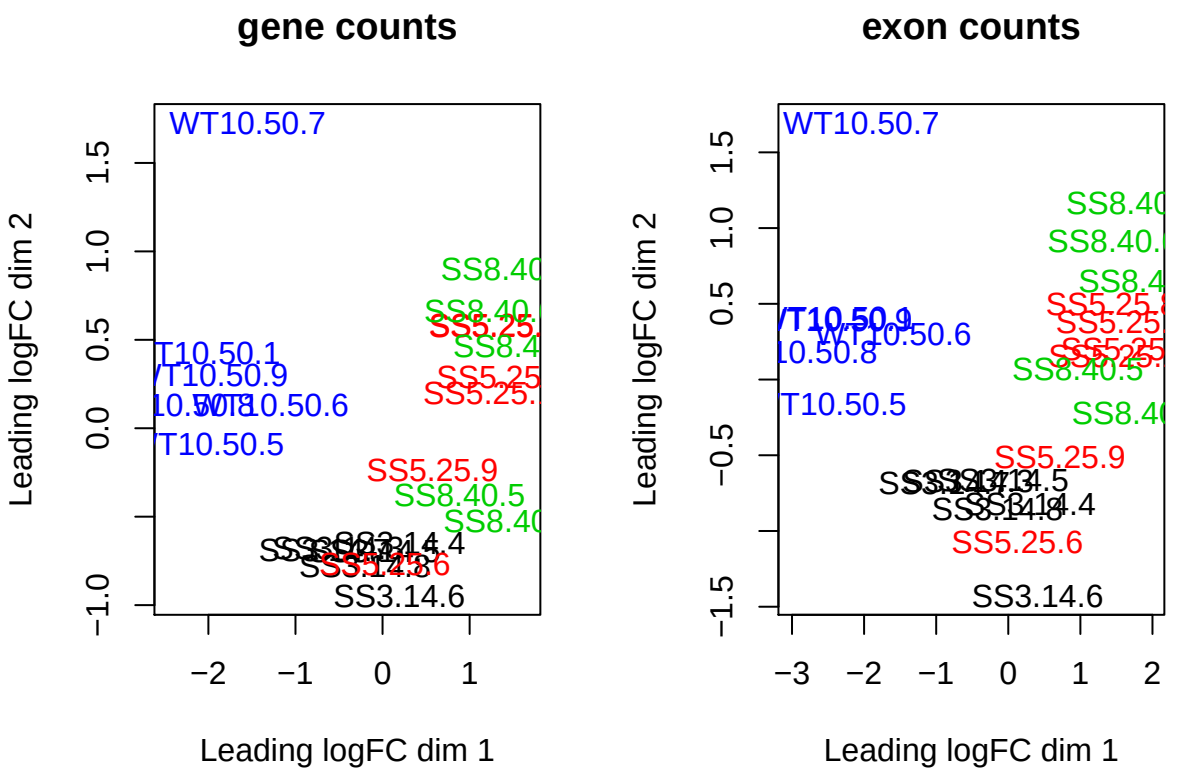
```
gene_host_v <- voom(gene_host_filtered2,
    design = model.matrix(~Strain, data = gene_host_filtered2$samples),
    plot = T)
```

Fig. 1 Limma voom model.

MDS to check overall expression differences

```
# par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))
par(mfrow = c(1, 2))
plotMDS(gene_host_v, main = "gene counts",
    labels = host_v$targets$FileName %>%
        str_remove("Host\\."), col = factor(host_v$targets$FileName %>%
```

```
        str_replace("Host\\.(.*?)\\..*",
            "\\1")) %>% as.integer())
#
plotMDS(host_v, main = "exon counts", labels = host_v$targets$FileName %>%
    str_remove("Host\\."), col = factor(host_v$targets$FileName %>%
    str_replace("Host\\.(.*?)\\..*", "\\1")) %>%
    as.integer())
```




```
dev.off()
```

```
## null device
##           1
```

Fig. 2 MDS plot. Left side - estimate according to whole gene counts. Right side - estimate according to exon expression.

Fit expression model and check stats

```
gene_host_fit <- lmFit(gene_host_v, design = model.matrix(~Strain,
    data = gene_host_filtered2$samples)) %>%
    eBayes()

# Summary
summary(decideTests(gene_host_fit))
```

```
##        (Intercept) StrainStrain3 StrainStrain5 StrainStrain8
## Down           125          2511          2401          2769
```

```
## NotSig         665         10996         11043         10239
## Up           14993          2276          2339          2775
```

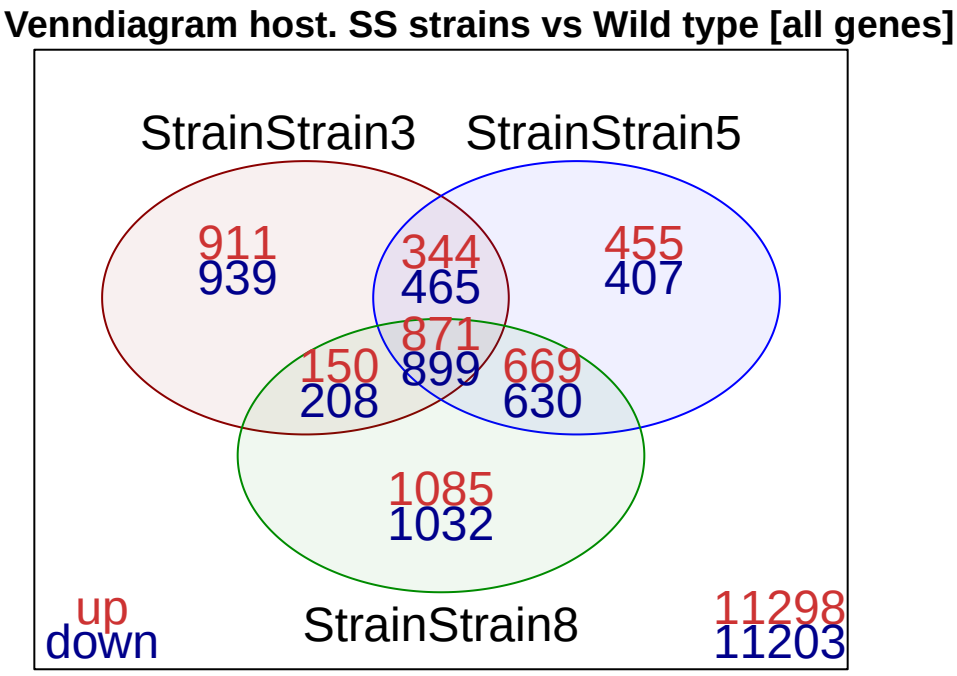
```
# view fit stats
topTable(gene_host_fit)
```

```
## Removing intercept from test coefficients
```

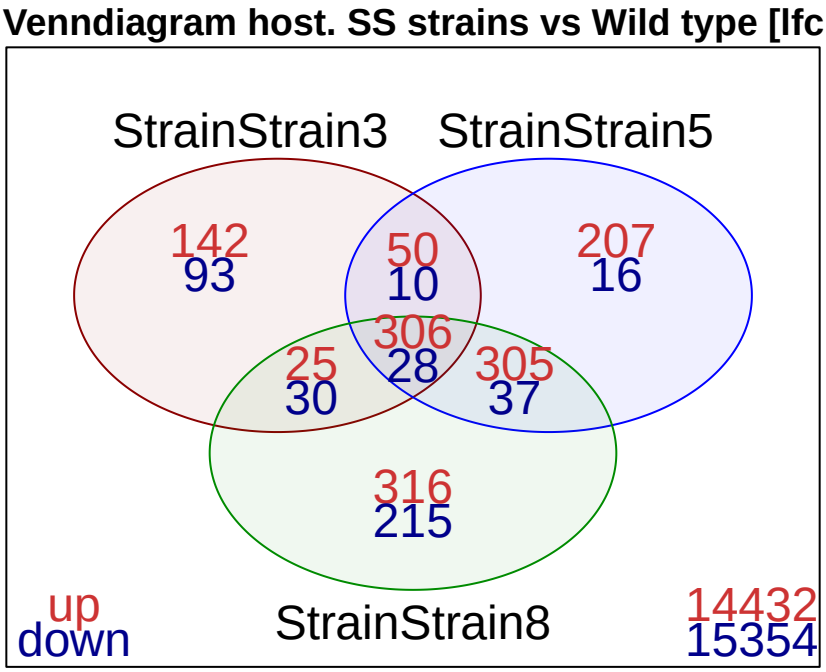
```
##                       gene
## 809   aten_0.1.m1.10825.m1
## 7167  aten_0.1.m1.17839.m1
## 24922   aten_0.1.m1.404.m1
## 14174 aten_0.1.m1.25815.m1
## 29156    aten_0.1.m1.86.m1
## 21481 aten_0.1.m1.34557.m1
## 6700  aten_0.1.m1.17333.m1
## 30169  aten_0.1.m1.9818.m1
## 2243  aten_0.1.m1.12397.m1
## 2971  aten_0.1.m1.13197.m1
##                                                Annotation StrainStrain3
## 809                    aarF domain-containing kinase 4-like      1.852557
## 7167                    integral membrane GPR155 isoform X1      2.477389
## 24922                class E basic helix-loop-helix 40-like     -1.273090
## 14174                  bifunctional TH2 mitochondrial-like      1.969079
## 29156                  L-rhamnose-binding lectin CSL2-like      1.801456
## 21481      PREDICTED: uncharacterized protein LOC107340645      2.037587
## 6700       PREDICTED: uncharacterized protein LOC107342533      2.775486
## 30169 uncharacterized aarF domain-containing kinase 1-like      1.762680
## 2243                        PRELI domain containing 3B-like      1.050141
## 2971  cyclic AMP-dependent transcription factor ATF-4-like     -1.222651
##       StrainStrain5 StrainStrain8  AveExpr        F      P.Value
## 809       1.1793309    0.32023369 5.171959 202.7191 2.321922e-16
## 7167      1.6736772    0.99212579 6.107115 201.2935 2.505276e-16
## 24922    -1.1256505   -0.79516442 8.557621 178.6786 9.013914e-16
## 14174     1.2862645    0.39598116 4.382975 171.2078 1.424097e-15
## 29156     2.1623499    3.28394680 6.694654 161.6697 2.627474e-15
## 21481     1.6076211    0.54894711 8.860846 160.2270 2.891153e-15
## 6700      1.6567288    0.91621341 2.963169 149.1089 6.217868e-15
## 30169     1.1805122    0.50667283 4.768507 144.6574 8.579385e-15
## 2243      0.6732786    0.03656377 6.406543 133.2702 2.044413e-14
## 2971     -0.9122953   -1.36795835 9.193336 130.1382 2.628073e-14
##          adj.P.Val
## 809   1.977039e-12
## 7167  1.977039e-12
## 24922 4.742220e-12
## 14174 5.619129e-12
## 29156 7.605179e-12
## 21481 7.605179e-12
## 6700  1.401952e-11
## 30169 1.692605e-11
## 2243  3.585219e-11
## 2971  3.873946e-11
```

```
# call genes as DE for different
# coefficients Limiting to a 2X change or
# greater (1 log2 fold change)
```

```
gene_host_coef_sig <- decideTests(gene_host_fit,
    lfc = 1) %>% as.data.frame() %>% bind_cols(gene_host_fit$genes)

# recreate venn diagram from prior work
# par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))

decideTests(gene_host_fit)[, -1] %>% vennDiagram(include = c("up",
    "down"), counts.col = c("brown3", "darkblue"),
    circle.col = c("darkred", "blue", "green4"))
title("Venndiagram host. SS strains vs Wild type [all genes]",
    line = 1)
```




```
decideTests(gene_host_fit, lfc = 1)[, -1] %>%
    vennDiagram(include = c("up", "down"),
        counts.col = c("brown3", "darkblue"),
        circle.col = c("darkred", "blue",
            "green4"))
title("Venndiagram host. SS strains vs Wild type [lfc > 1]",
    line = 1)
```




```
# dev.off()
```

Fig. 3 Venndiagram showing genes that are significantly different expressed between the wild type and respective selected strains. For example: With lfc > 1 Strain 5 has 207 gene significantly upregulated and 16 genes downregulated compared to the wild type.

Exploratory analysis 1

PLOTTING NEEDS MORE WORK

Filter genes from Venndiagram. Which are the genes that all selected strains have significantly expressed compared to the wild type?

DATA EXPORT

Code silenced and hidden. See DGE table: host_DGE_expression_FULLlist.xlsx.

Exploratory analysis 2

PLOTTING NEEDS MORE WORK

Filter genes according to annotation. Which are heat stress related genes that are differentially expressed between the wild type and selected strains?

```
# filter based on gene host_annotations
head(topTable(gene_host_fit, number = Inf,
    p.value = 0.05) %>% filter(Annotation %>%
    str_detect("[H|h]eat|[G|g]lutathione|[C|c]haperone|[D|d]ismutase")))
```

```
## Removing intercept from test coefficients
```

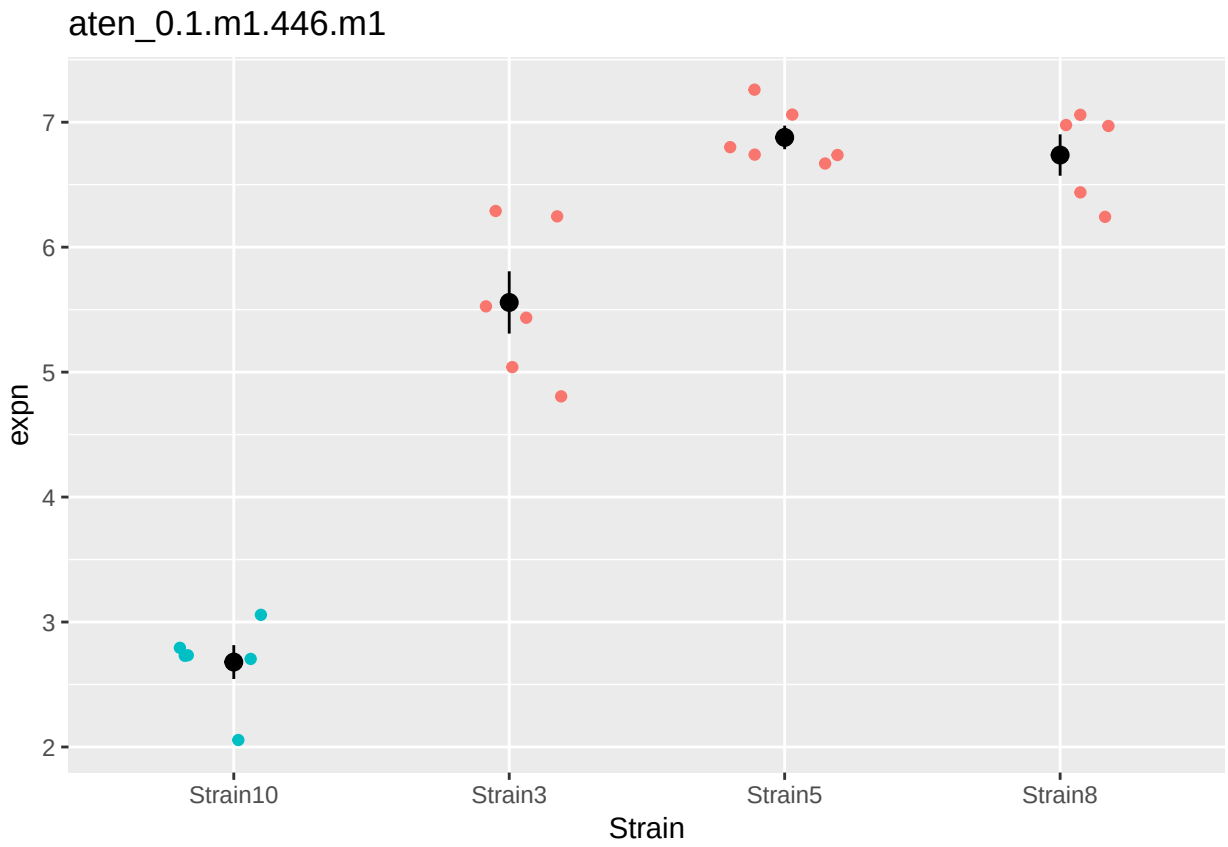
```
##                   gene                                    Annotation
## 1   aten_0.1.m1.446.m1                                  heat shock 70
## 2  aten_0.1.m1.4452.m1                                small heat shock
## 3 aten_0.1.m1.12315.m1 microsomal glutathione S-transferase 1-like
## 4  aten_0.1.m1.3047.m1      glutathione S-transferase omega-1-like
## 5 aten_0.1.m1.27797.m1                       97 kDa heat shock -like
## 6 aten_0.1.m1.24908.m1                  Glutathione S-transferase 1
##   StrainStrain3 StrainStrain5 StrainStrain8  AveExpr        F      P.Value
## 1    2.90606567     4.2181271     4.1020821 5.407729 69.58501 1.688761e-11
## 2    0.10567057     1.5212756     1.9736976 3.998256 30.98113 3.823637e-08
## 3    0.01857967     0.2579576     0.6969552 5.504046 29.01991 6.828279e-08
## 4    0.12940315     0.8073737     0.8794700 5.755704 25.98795 1.783309e-07
## 5   -0.18186461    -0.1413062     0.1900266 7.252046 24.13672 3.345909e-07
## 6   -0.03255637     0.5967238     1.2688668 5.773081 22.46486 6.098464e-07
##      adj.P.Val
## 1 4.595468e-09
## 2 1.275866e-06
## 3 1.914223e-06
## 4 3.903741e-06
## 5 5.987357e-06
## 6 9.510699e-06
```

Plotting, by gene and by exon.

ISSUE MARKED IN GITHUB - Gene expression does not seem to match with exon expression. Exon plots dont seem to make sense??

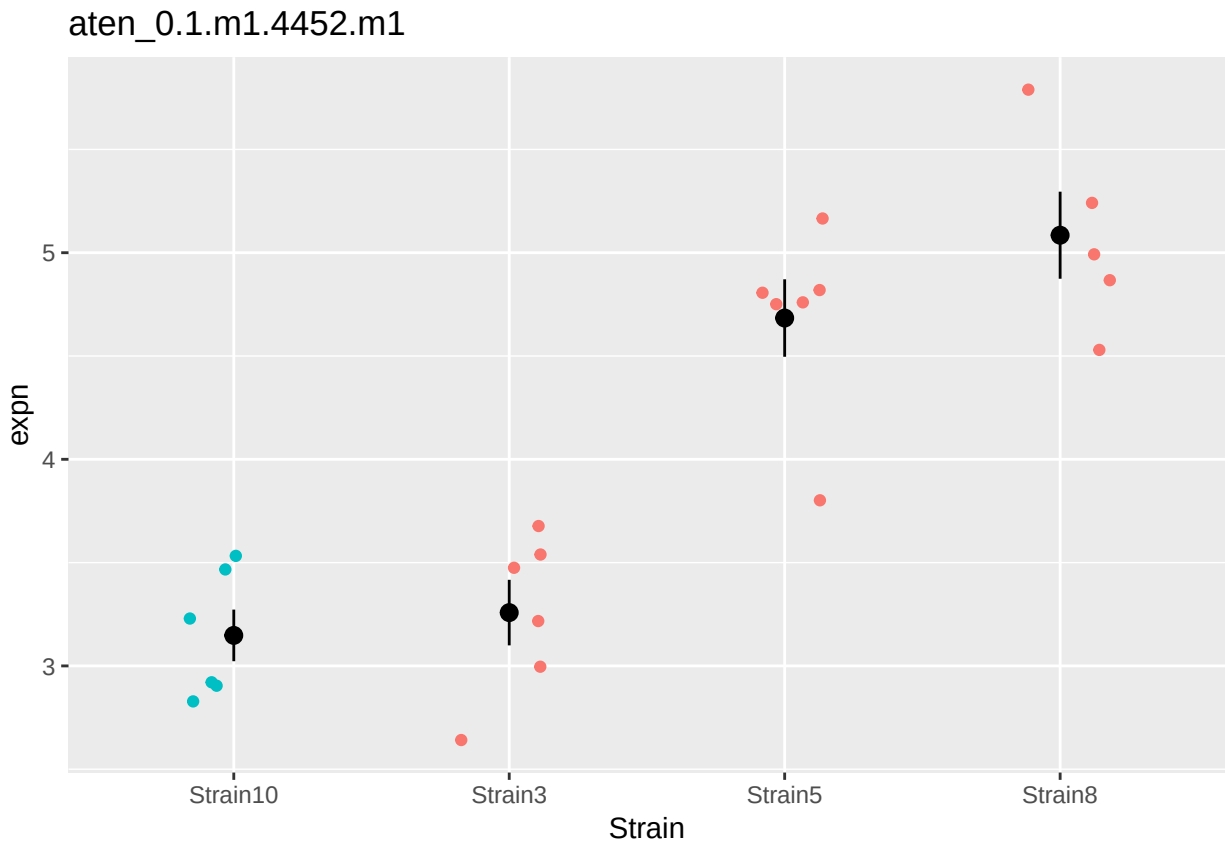
```
## Plot by gene, according to annotation
plot_host_gene <- function(gene) {
    # expn <-
    # gene_host_v$E[gene_host_v$genes$Annotation
    # == gene, ]
    expn <- gene_host_v$E[gene_host_v$genes$gene ==
        gene, ]
    gene_host_v$targets %>% mutate(expn = expn) %>%
        ggplot(aes(x = Strain, y = expn,
            colour = CellType)) + geom_point(position = position_jitter(width = 0.2)) +
        stat_summary(colour = "black") +
        ggtitle(gene) + theme(legend.position = "none")
}

# plot_host_gene('heat shock 70')
# plot_host_gene('small heat shock')
# plot_host_gene('Glutathione
# S-transferase 1')
# plot_host_gene('molecular chaperone')

# gene of choice par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))
plot_host_gene("aten_0.1.m1.446.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```



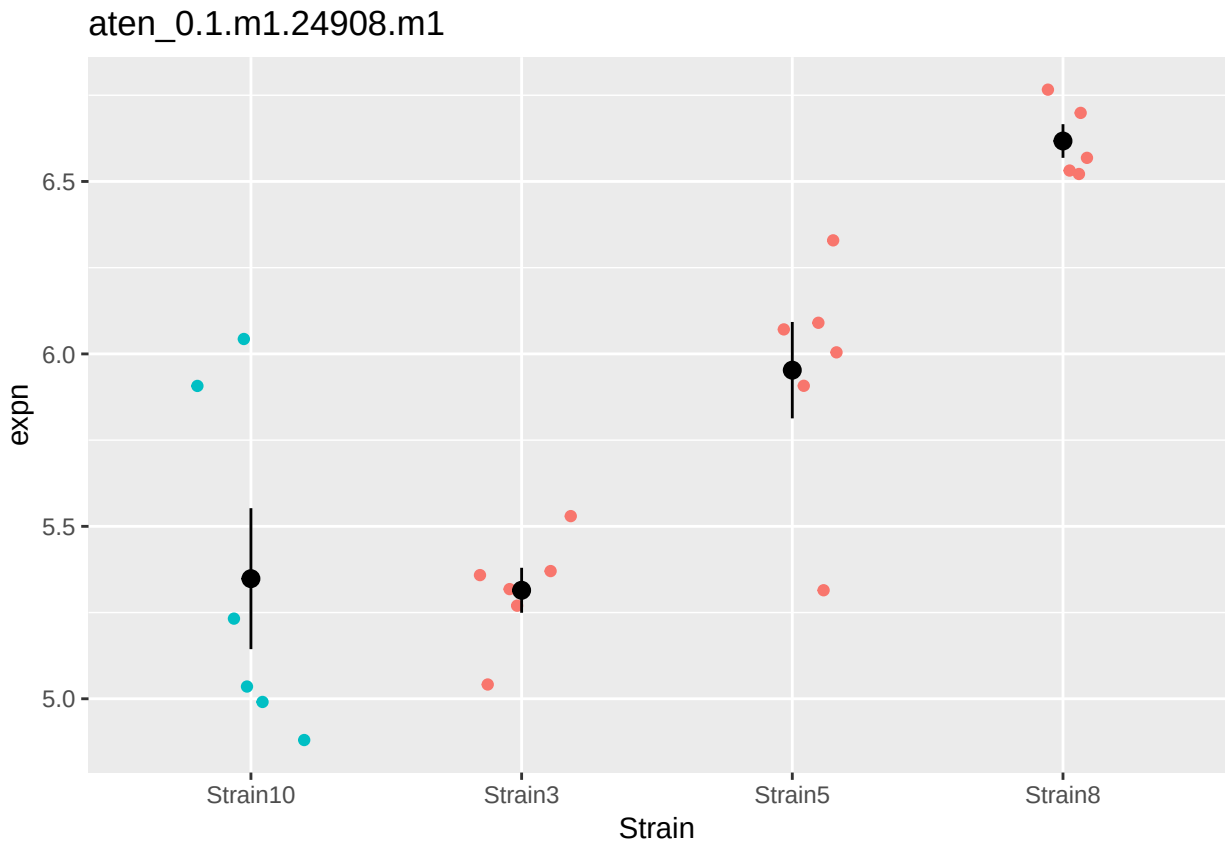

```
plot_host_gene("aten_0.1.m1.4452.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```



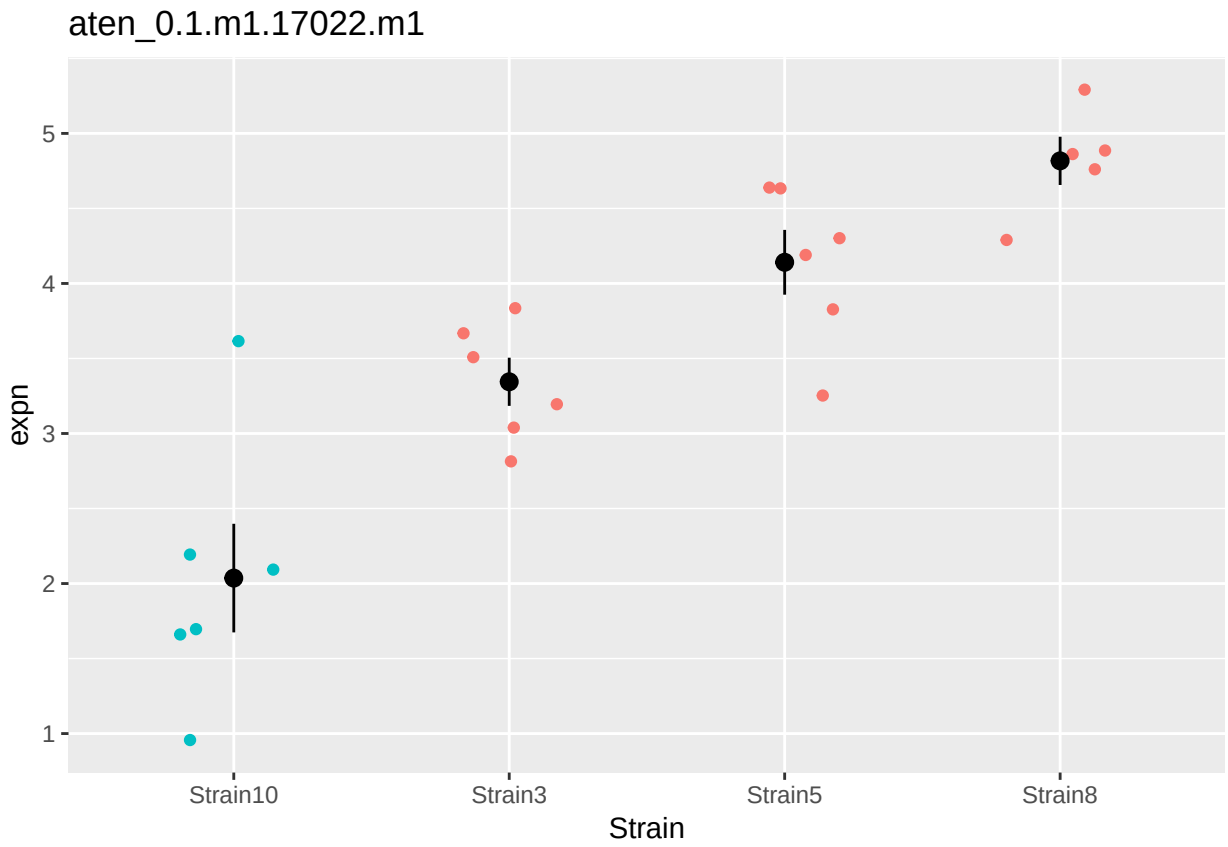

```
plot_host_gene("aten_0.1.m1.24908.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```




```
plot_host_gene("aten_0.1.m1.17022.m1")
```

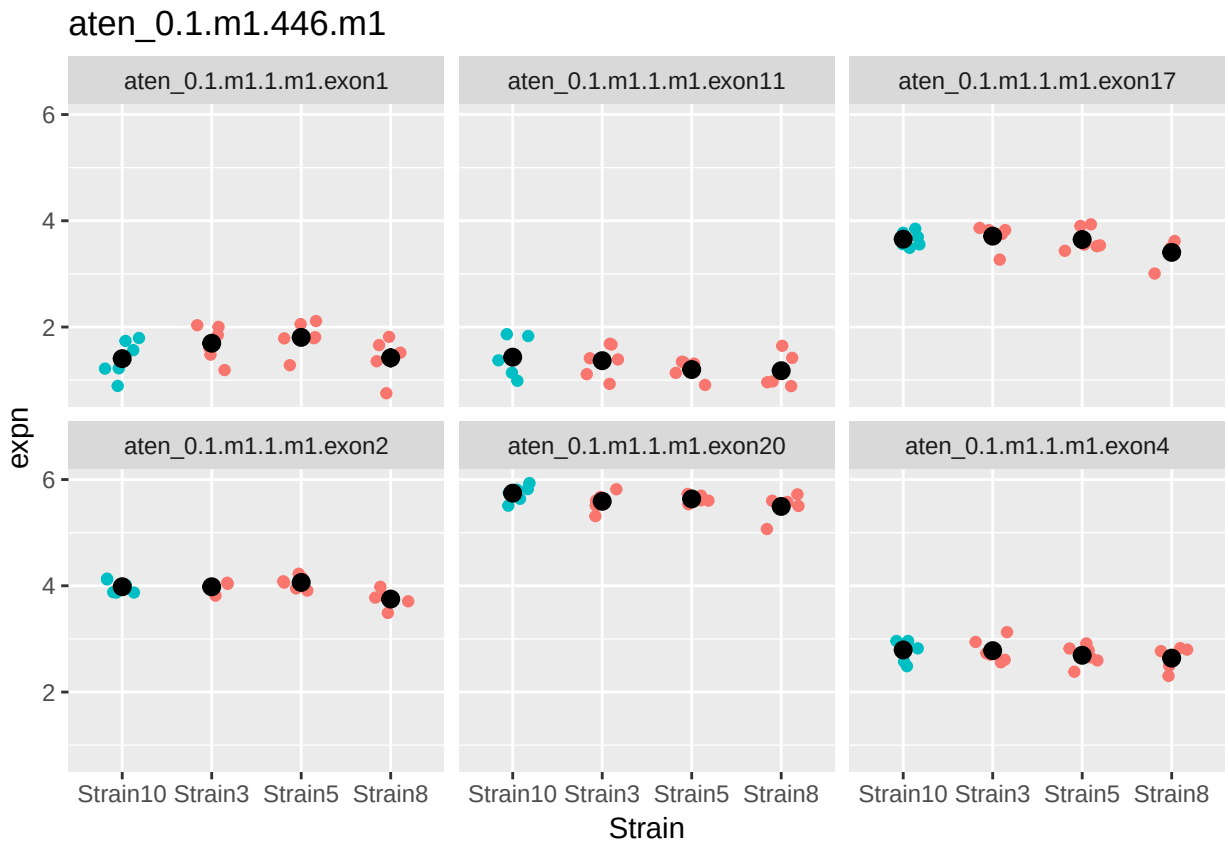
```
## No summary function supplied, defaulting to `mean_se()
```




```
# dev.off()
# plot_host_gene('aten_0.1.m1.10016.m1')


## Plot by exon
plot_host_gene_by_exon <- function(gene) {
    host_v$E[str_detect(host_v$genes$EntrezGeneID,
        "aten_0.1.m1.1.m1."), ] %>% as.data.frame() %>%
        mutate(exon = host_v$genes$EntrezGeneID[str_detect(host_v$genes$EntrezGeneID,
            "aten_0.1.m1.1.m1.")]) %>% gather(sample,
        expn, -exon) %>% left_join(host_v$targets %>%
        as.data.frame() %>% rownames_to_column("sample")) %>%
        ggplot(aes(x = Strain, y = expn,
            colour = CellType)) + geom_point(position = position_jitter(width = 0.2)) +
        stat_summary(colour = "black") +
        ggtitle(gene) + facet_wrap(~exon) +
        theme(legend.position = "none")
}
# exons of gene of choice

# par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))
plot_host_gene_by_exon("aten_0.1.m1.446.m1")
```

```
## Joining, by = "sample"
```

```
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
## No summary function supplied, defaulting to `mean_se()
```




```
# plot_host_gene_by_exon('aten_0.1.m1.4452.m1')
# plot_host_gene_by_exon('aten_0.1.m1.24908.m1')
# plot_host_gene_by_exon('aten_0.1.m1.17022.m1')

# dev.off()
```

---

2 Symbiont expression

Loading data

Read in the symbiont count matrix and convert to log2cpm

```
symbiont_counts <- read_tsv("data/MATRIX_Counts-FINAL-Symbiont.txt")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   EntrezGeneID = col_character()
```

```
## )
```

```
## See spec(...) for full column specifications.
```

```
symbiont_samples <- read_tsv("data/SampleInfo_Symbiont-STAR.txt")
```

```
## Parsed with column specification:
## cols(
##   FileName = col_character(),
##   SampleName = col_character(),
##   CellType = col_character(),
##   Strain = col_character()
## )
```

```
symbiont_annotations <- read_tsv("data/Symbiont_diamond_annot_20190117_1327.txt")
```

```
## Parsed with column specification:
## cols(
##   EntrezGeneID = col_character(),
##   Annotation = col_character(),
##   GOterms = col_character()
## )
```

```
# symbiont_exon_annnotations <-
# read_tsv('data/Symbiont-Gene-Annotation.txt')

# Exons
symbiont_DGE <- DGEList(symbiont_counts[,
    -1], genes = symbiont_counts[, 1], samples = symbiont_samples)

filt <- filterByExpr(symbiont_DGE, design = model.matrix(~Strain,
    data = symbiont_DGE$samples), min.count = 10)

symbiont_filtered <- symbiont_DGE[filt, ,
    keep.lib.sizes = F]

symbiont_filtered <- calcNormFactors(symbiont_filtered)

symbiont_v <- voom(symbiont_filtered, design = model.matrix(~Strain,
    data = symbiont_filtered$samples), plot = F)
```

Collate to gene level

Might be better for some uses to look at whole gene level expression initally before drilling down to exon level data.

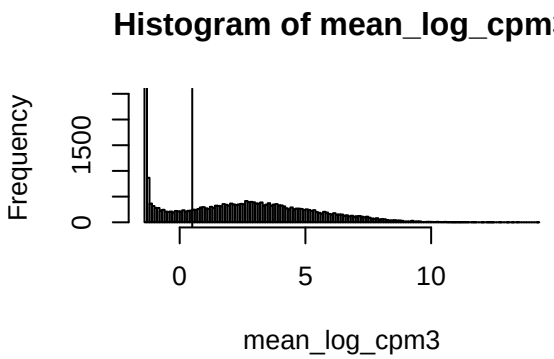
```
gene_symbiont_counts <- symbiont_counts %>%
    separate(EntrezGeneID, c("gene", "exon"),
        sep = "\\.exon") %>% group_by(gene) %>%
    summarise_if(is.numeric, sum)

# check that symbiont_annotations and
# genes IDs match.. needs to be all TRUE
# events
table((symbiont_annotations[, 1] == (gene_symbiont_counts[,
    1])))
```

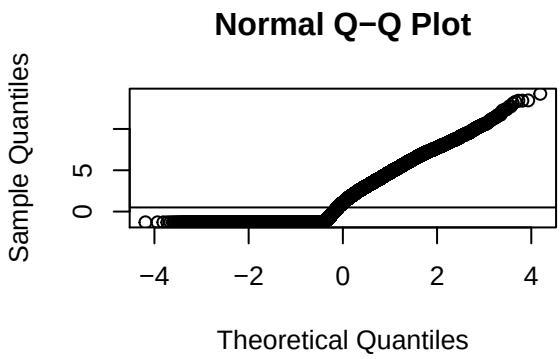
```
##
##  TRUE
## 35913
```

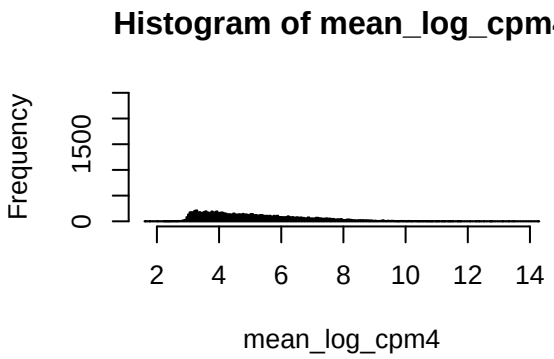
```
# then bind symbiont_annotations to gene
# list
gene_symbiont_DGE <- DGEList(gene_symbiont_counts[,
    -1], genes = gene_symbiont_counts[, 1] %>%
    bind_cols(symbiont_annotations[, -1]),
    samples = symbiont_samples)

# check that annotation table and
# gene_symbiont_DGE match.. needs to be
# all TRUE events, two columns to match
table((gene_symbiont_DGE$genes == (symbiont_annotations)))
```

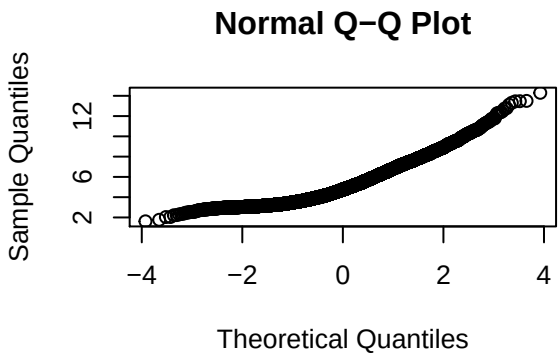
```
##
##  TRUE
## 83840
```

```
gene_symbiont_filt <- filterByExpr(gene_symbiont_DGE,
    design = model.matrix(~Strain, data = gene_symbiont_DGE$samples),
    min.count = 40)

gene_symbiont_filtered1 <- gene_symbiont_DGE[gene_symbiont_filt,
    , keep.lib.sizes = F]

gene_symbiont_filtered2 <- calcNormFactors(gene_symbiont_filtered1)


# Check filter cut off before and after
par(mfrow = c(2, 2))
mean_log_cpm3 <- aveLogCPM(gene_symbiont_DGE$counts)
filter_threshold <- 0.5
hist(mean_log_cpm3, breaks = 200, ylim = c(0,
    2500))
abline(v = filter_threshold)
qqnorm(mean_log_cpm3)
abline(h = filter_threshold)
#
mean_log_cpm4 <- aveLogCPM(gene_symbiont_filtered1$counts)
filter_threshold <- 0.5
hist(mean_log_cpm4, breaks = 200, ylim = c(0,
    2500))
abline(v = filter_threshold)
qqnorm(mean_log_cpm4)
abline(h = filter_threshold)
```





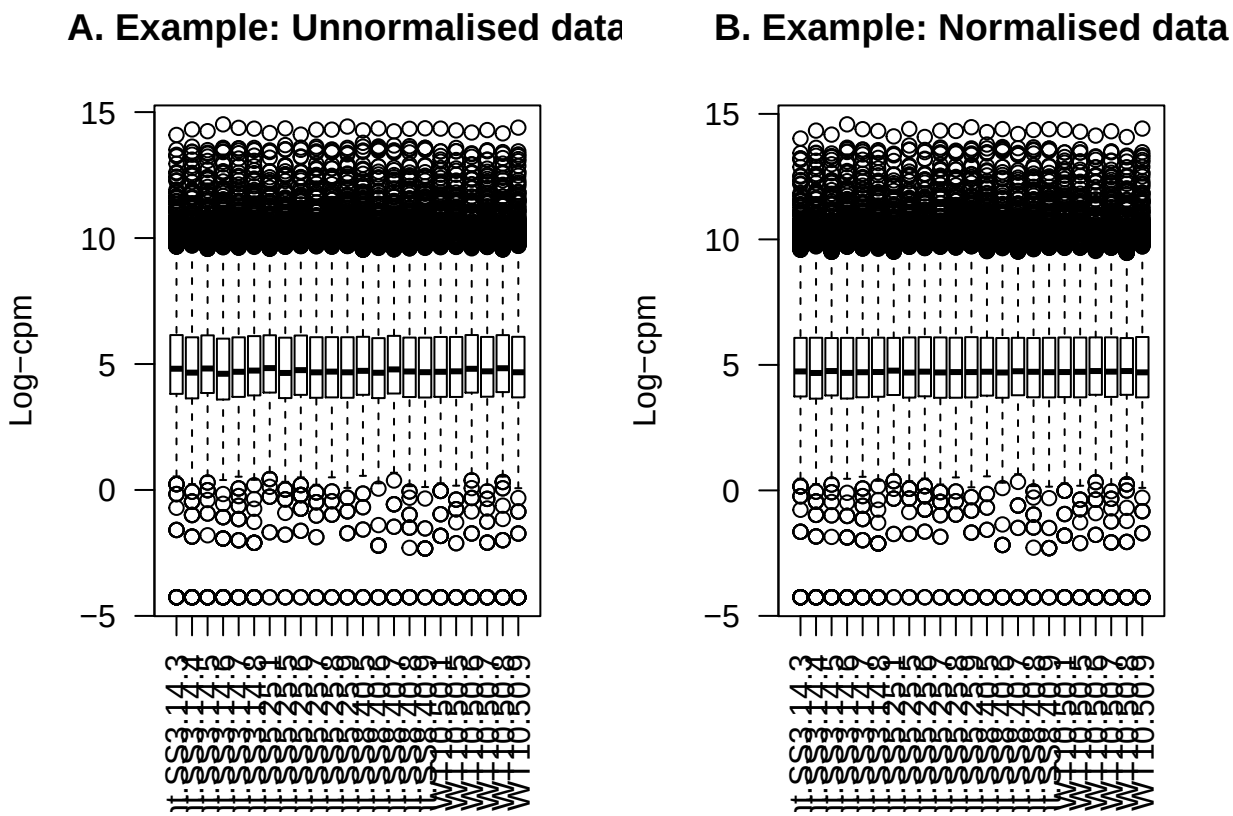





```
# Check normalised and unnormalised data
par(mfrow = c(1, 2))
Symbiont_lcpm1 <- cpm(gene_symbiont_filtered1,
    log = TRUE)
boxplot(Symbiont_lcpm1, las = 2, main = "")
title(main = "A. Example: Unnormalised data",
    ylab = "Log-cpm")
#
gene_symbiont_filtered2$samples$norm.factors
```

```
##  [1] 1.0520850 0.9862067 1.0477746 0.9530875 0.9894508 1.0164701 1.0470959
##  [8] 0.9661660 1.0188871 0.9795719 0.9888226 0.9706820 1.0008197 0.9707695
## [15] 1.0274005 0.9857248 0.9769038 0.9846052 0.9894032 1.0387732 0.9852348
## [22] 1.0551194 0.9792173
```

```
Symbiont_lcpm2 <- cpm(gene_symbiont_filtered2,
    log = TRUE)
boxplot(Symbiont_lcpm2, las = 2, main = "")
title(main = "B. Example: Normalised data",
    ylab = "Log-cpm")
```




```
dev.off()
```

```
## null device
##           1
```

```
# check that data is still all matching
# up
table(symbiont_samples$SampleName == colnames(gene_symbiont_filtered2))
```

```
##
## TRUE
##   23
```

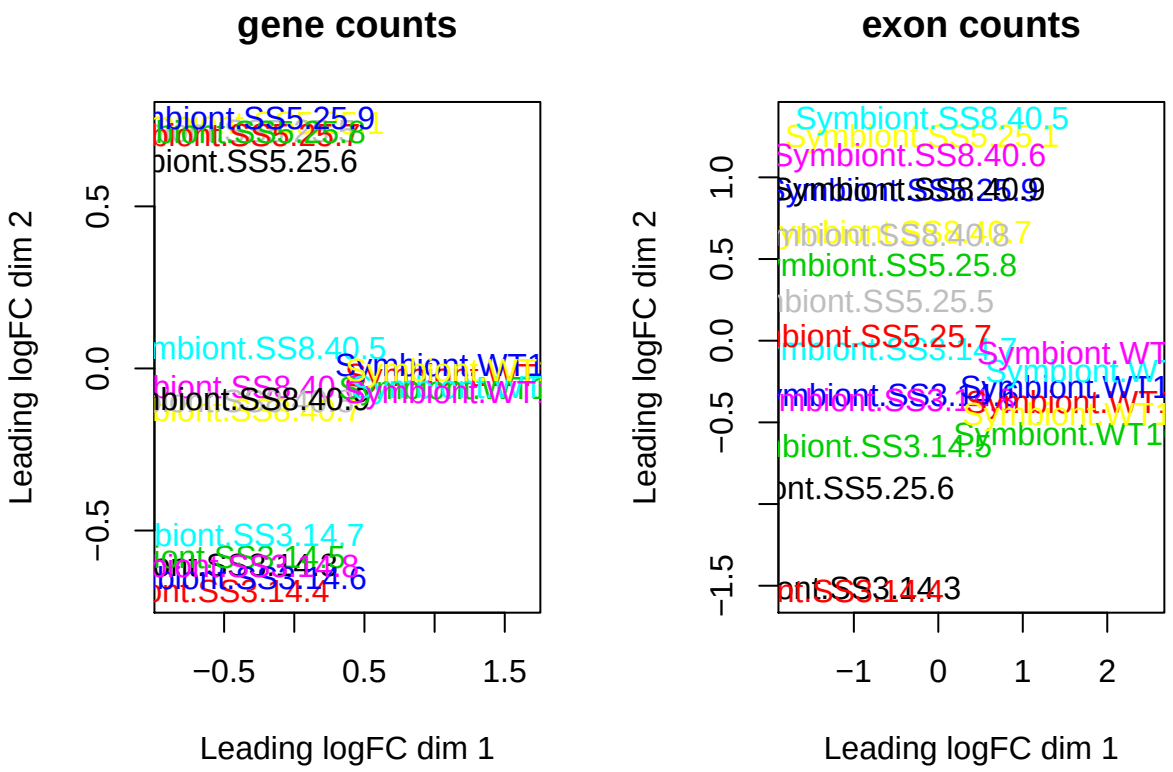
```
gene_symbiont_v <- voom(gene_symbiont_filtered2,
    design = model.matrix(~Strain, data = gene_symbiont_filtered2$samples),
    plot = T)
```

Fig. 1 Limma voom model.

MDS to check overall expression differences

```
# par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))
par(mfrow = c(1, 2))
plotMDS(gene_symbiont_v, main = "gene counts",
    labels = symbiont_v$targets$FileName %>%
        str_remove("symbiont\\."), col = factor(symbiont_v$targets$FileName %>%
```

```
        str_replace("symbiont\\.(.*?)\\..*",
            "\\1")) %>% as.integer())

plotMDS(symbiont_v, main = "exon counts",
    labels = symbiont_v$targets$FileName %>%
        str_remove("symbiont\\."), col = factor(symbiont_v$targets$FileName %>%
        str_replace("symbiont\\.(.*?)\\..*",
            "\\1")) %>% as.integer())
```




```
dev.off()
```

```
## null device
##           1
```

Fig. 2 MDS plot. Left side - estimate according to whole gene counts. Right side - estimate according to exon expression.

Fit expression model and check stats

```
gene_symbiont_fit <- lmFit(gene_symbiont_v,
    design = model.matrix(~Strain, data = gene_symbiont_filtered2$samples)) %>%
    eBayes()

# Summary
summary(decideTests(gene_symbiont_fit))
```

```
##        (Intercept) StrainStrain3 StrainStrain5 StrainStrain8
```

```
## Down            14          1576           987          1008
## NotSig          10          8929          9889          9833
## Up           11571          1090           719           754
```

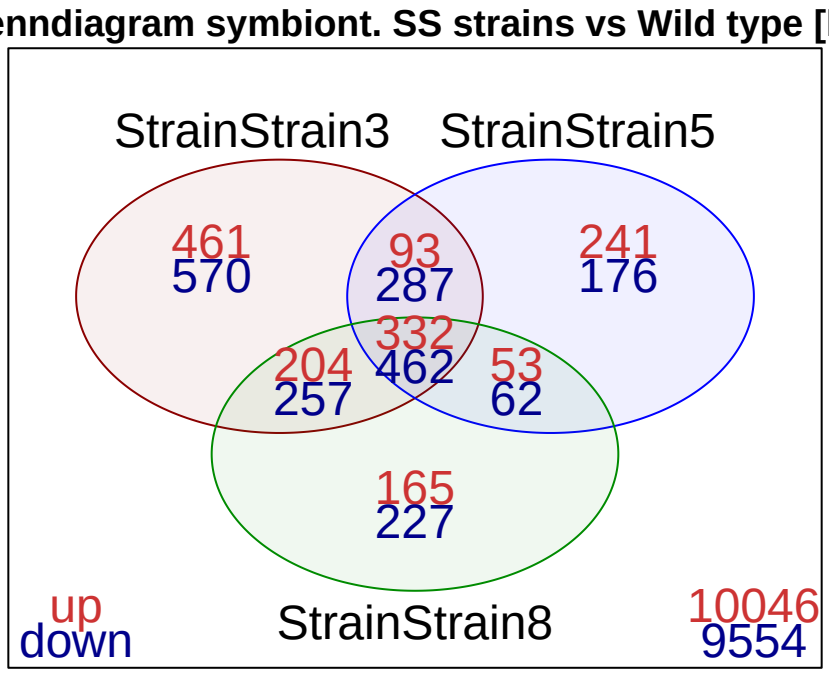
```
# view fit stats
topTable(gene_symbiont_fit)
```

```
## Removing intercept from test coefficients
```

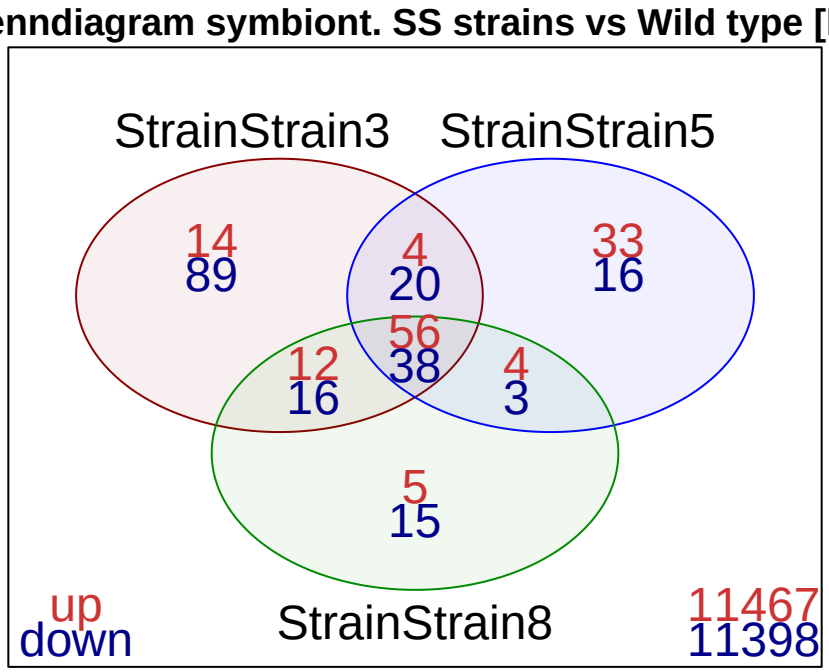
```
##                             gene
## 30200  SymbC1.scaffold6975.1.m1
## 35644  SymbC1.scaffold9776.2.m1
## 8350  SymbC1.scaffold1742.11.m1
## 14619  SymbC1.scaffold2440.6.m1
## 27075  SymbC1.scaffold5602.4.m1
## 23196  SymbC1.scaffold4366.3.m1
## 31775  SymbC1.scaffold7583.2.m1
## 32233  SymbC1.scaffold7866.3.m1
## 29442   SymbC1.scaffold662.3.m1
## 14616  SymbC1.scaffold2440.2.m1
##                                                Annotation
## 30200                        Reticulocyte-binding 2-like a
## 35644                                        Nipped-B B
## 8350                           Tip elongation aberrant 1
## 14619             hypothetical protein AK812_SmicGene34950
## 27075                          Nucleoside triphosphatase I
## 23196 Retrovirus-related Pol poly from transposon TNT 1-94
## 31775               LINE-1 retrotransposable element ORF2
## 32233                                              <NA>
## 29442          tccd-inducible-parp-like domain-containing
## 14616             hypothetical protein AK812_SmicGene2381
##                                GOterms StrainStrain3 StrainStrain5
## 30200                             <NA>     -4.145621    -3.3553461
## 35644                             <NA>      2.402863     1.9781435
## 8350                        GO:0005515     -3.615268    -3.2993406
## 14619                             <NA>      2.042499     2.0585331
## 27075                             <NA>      5.814926     4.6739216
## 23196                             <NA>     -3.018912     5.5228236
## 31775                       GO:0008168      1.346338    -3.6330684
## 32233                             <NA>      2.209306     2.5836103
## 29442 GO:0005515|GO:0016891|GO:0003950     -1.975775    -0.6995472
## 14616                             <NA>      2.522405     2.4315404
##       StrainStrain8  AveExpr        F      P.Value    adj.P.Val
## 30200    -2.9681419 6.003198 525.5683 6.314527e-27 4.381264e-23
## 35644     2.2023186 9.626758 519.4718 7.557161e-27 4.381264e-23
## 8350     -4.6470339 5.212733 429.6654 1.395646e-25 5.394171e-22
## 14619     2.1662159 7.076648 373.1342 1.208509e-24 3.503167e-21
## 27075     5.0916654 5.588741 364.2280 1.747561e-24 3.940822e-21
## 23196    -0.9367589 2.158743 360.5625 2.039235e-24 3.940822e-21
## 31775    -0.6071650 5.952907 356.2273 2.452492e-24 4.062378e-21
## 32233     2.0646506 6.977934 345.4077 3.924866e-24 5.688603e-21
## 29442    -2.0079593 6.160791 327.5203 8.819579e-24 1.136256e-20
## 14616     2.4644820 6.452094 303.2486 2.841374e-23 3.294573e-20
```

```
# call genes as DE for different
# coefficients Limiting to a 2X change or
```

```
# greater (1 log2 fold change.. lfc = 1 =
# 2 fold change..)
gene_symbiont_coef_sig <- decideTests(gene_symbiont_fit,
    lfc = 1) %>% as.data.frame() %>% bind_cols(gene_symbiont_fit$genes)

# recreate venn diagram from prior work
# par(mar=c(5,6,3,1))
# layout(matrix(c(1,2), 2, 2, byrow =
# TRUE))

decideTests(gene_symbiont_fit)[, -1] %>%
    vennDiagram(include = c("up", "down"),
        counts.col = c("brown3", "darkblue"),
        circle.col = c("darkred", "blue",
            "green4"))
title("Venndiagram symbiont. SS strains vs Wild type [lfc > 1]",
    line = 1)
```




```
decideTests(gene_symbiont_fit, lfc = 1)[,
    -1] %>% vennDiagram(include = c("up",
    "down"), counts.col = c("brown3", "darkblue"),
    circle.col = c("darkred", "blue", "green4"))
title("Venndiagram symbiont. SS strains vs Wild type [lfc > 1]",
    line = 1)
```




```
# dev.off()
```

Fig. 3 Venndiagram showing genes that are significantly different expressed between the wild type and respective selected strains. For example: With lfc > 1 Strain 5 has 207 gene significantly upregulated and 16 genes downregulated compared to the wild type.

Exploratory analysis 1

Heatmaps coming soon.

Filter genes from Venndiagram. Which are the genes that all selected strains have significantly expressed compared to the wild type?

DATA EXPORT

Code silenced and hidden. See DGE table: symbiont_DGE_expression_FULLlist.xlsx.

Exploratory analysis 2

PLOTTING NEEDS MORE WORK,

Get HEAT STRESS ASSOCIATED GENES FROM A PARTICULAR STRAIN Filter genes according to annotation. Which are heat stress related genes that are differentially expressed between the wild type and selected strains?

```
# filter based on gene
# symbiont_annotations
head(topTable(gene_symbiont_fit, number = Inf,
```

```
    p.value = 0.05) %>% filter(Annotation %>%
    str_detect("[H|h]eat|[G|g]lutathione|[C|c]haperone|[D|d]ismutase")))
```

```
## Removing intercept from test coefficients
```

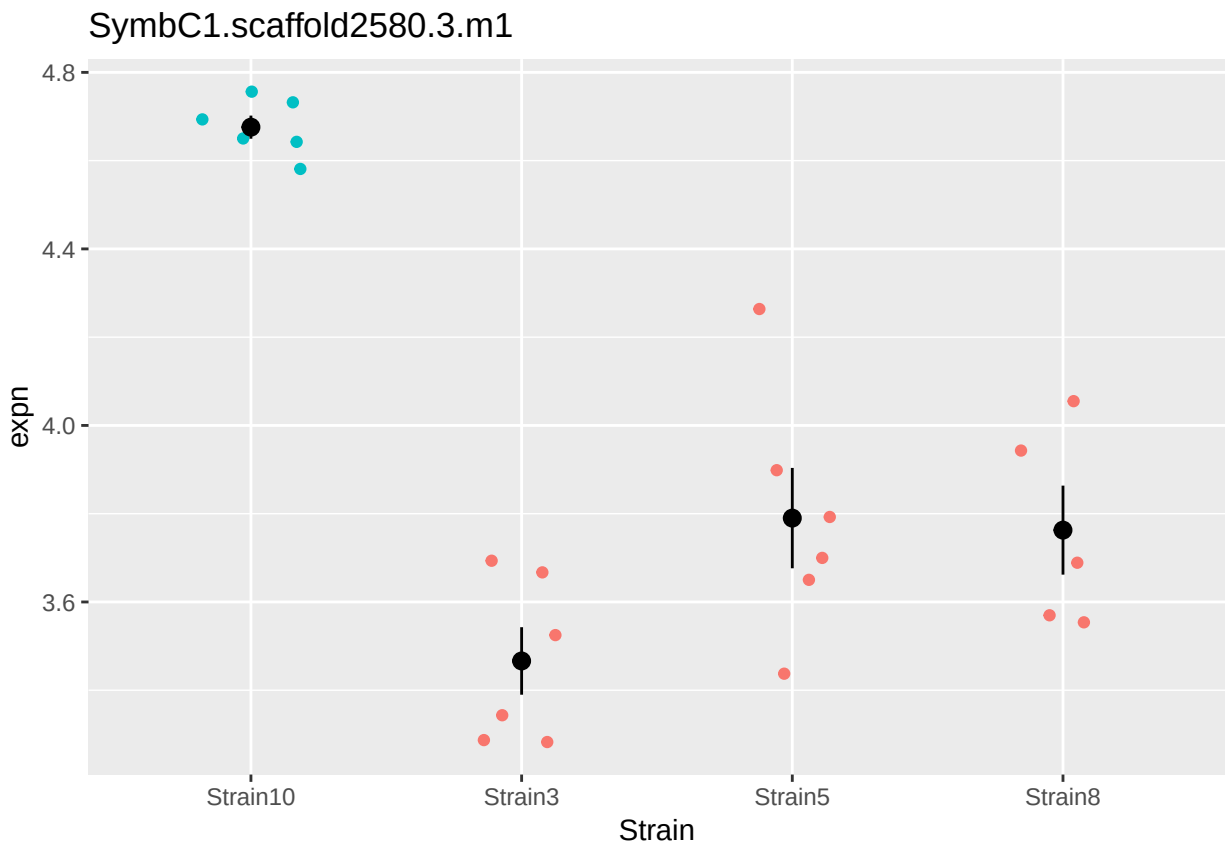
```
##                         gene                                Annotation
## 1 SymbC1.scaffold2580.3.m1                        molecular chaperone
## 2 SymbC1.scaffold4974.6.m1                  Glutathione S-transferase
## 3 SymbC1.scaffold170.11.m1                    Chaperone chloroplastic
## 4 SymbC1.scaffold1708.3.m1                  Glutathione S-transferase
## 5 SymbC1.scaffold5833.1.m1                Glutathione S-transferase 1
## 6 SymbC1.scaffold1454.8.m1 glutathione S-transferase domain-containing
##                             GOterms StrainStrain3 StrainStrain5
## 1                              <NA>    -1.1945313   -0.87998306
## 2                        GO:0046872     0.8007086    0.71803429
## 3 GO:0016887|GO:0005524|GO:0019538     0.5250186    0.02695672
## 4            GO:0005515|GO:0046872     0.5793043    0.48268085
## 5                              <NA>    -0.4119162   -0.41792019
## 6                        GO:0005515     0.4828147    0.43430512
##   StrainStrain8  AveExpr        F      P.Value    adj.P.Val
## 1    -0.9329820 3.930776 41.19501 5.295670e-11 2.537326e-09
## 2     0.7210122 5.333209 40.98902 5.636087e-11 2.678296e-09
## 3     0.3360369 8.541804 32.02519 1.111507e-09 4.184391e-08
## 4     0.4277791 5.892650 23.09141 4.347191e-08 1.169505e-06
## 5    -0.4382063 6.621592 21.85403 7.781417e-08 1.957170e-06
## 6     0.4638536 8.535494 17.51624 7.227806e-07 1.408511e-05
```

Plotting, plot by gene and by exon.

ISSUE MARKED IN GITHUB - Gene expression does not seem to match with exon expression. Exon plots dont seem to make sense??

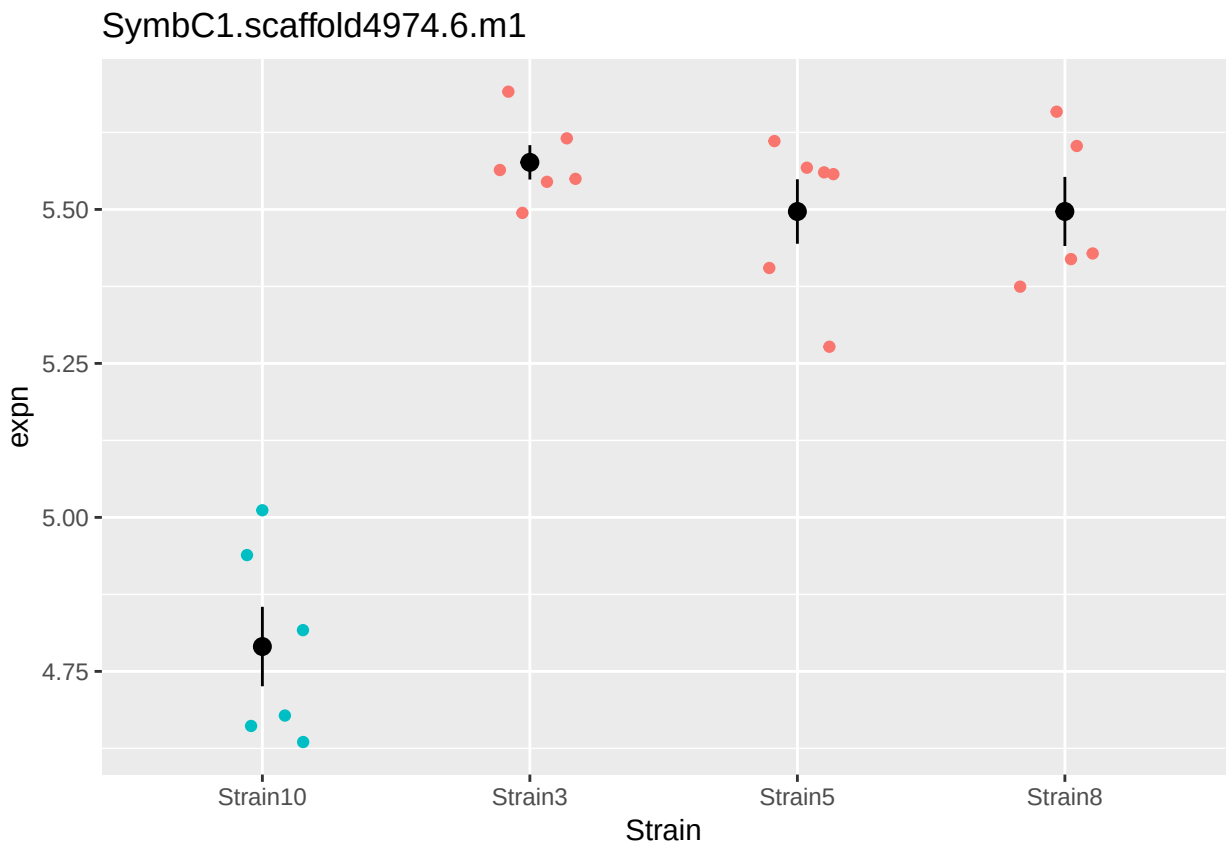
```
## Plot by gene, according to annotation
plot_symbiont_gene <- function(gene) {
    # expn <-
    # gene_symbiont_v$E[gene_symbiont_v$genes$Annotation
    # == gene, ]
    expn <- gene_symbiont_v$E[gene_symbiont_v$genes$gene ==
        gene, ]
    gene_symbiont_v$targets %>% mutate(expn = expn) %>%
        ggplot(aes(x = Strain, y = expn,
            colour = CellType)) + geom_point(position = position_jitter(width = 0.2)) +
        stat_summary(colour = "black") +
        ggtitle(gene) + theme(legend.position = "none")
}
# gene of choice
par(mar = c(5, 6, 3, 1))
layout(matrix(c(1, 2), 2, 2, byrow = TRUE))
plot_symbiont_gene("SymbC1.scaffold2580.3.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```



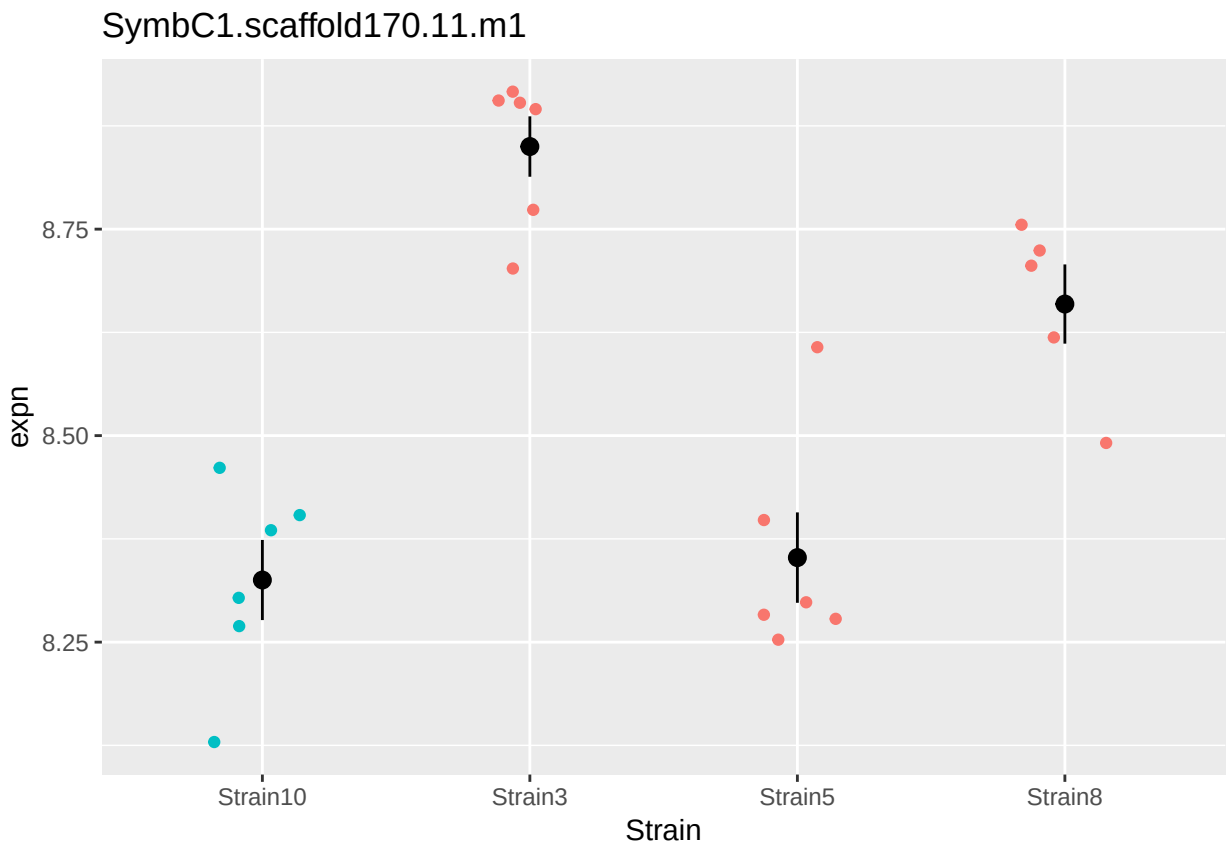

```
plot_symbiont_gene("SymbC1.scaffold4974.6.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```



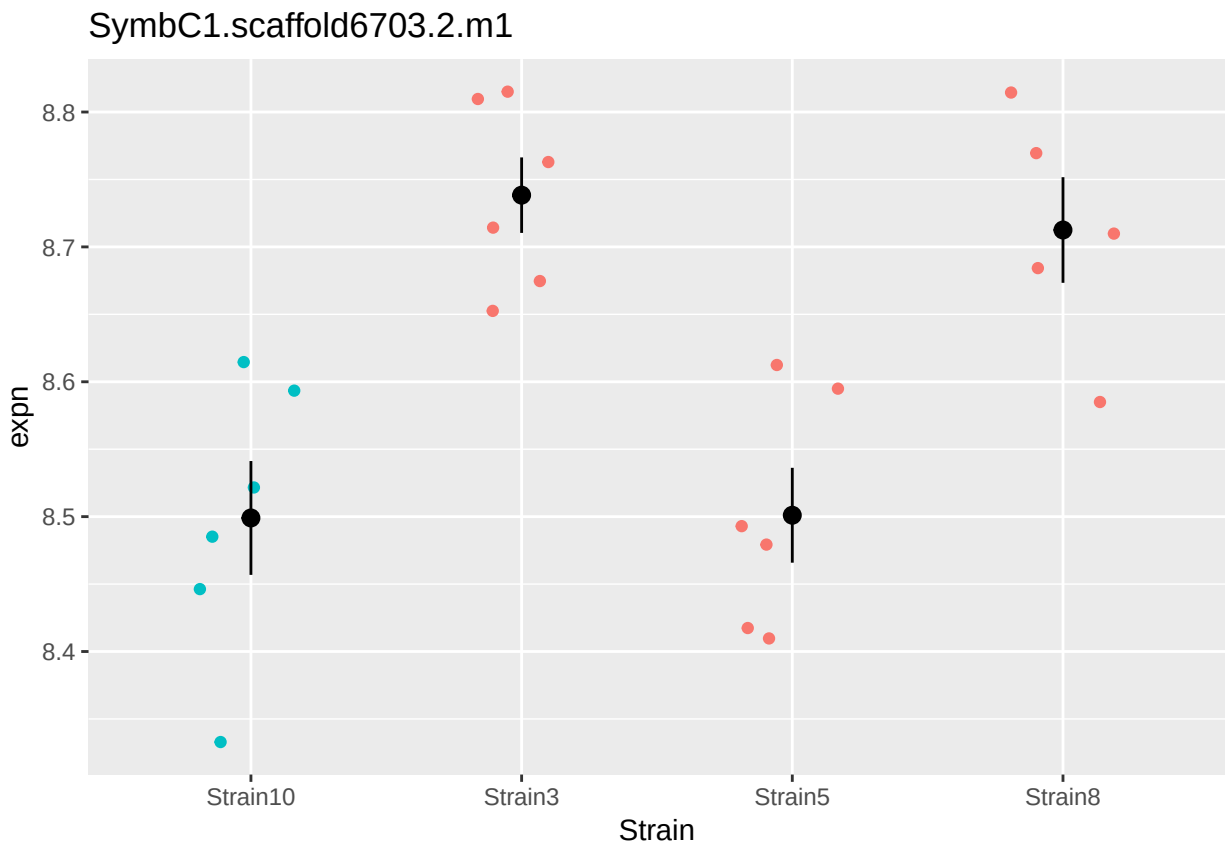

```
plot_symbiont_gene("SymbC1.scaffold170.11.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```




```
plot_symbiont_gene("SymbC1.scaffold6703.2.m1")
```

```
## No summary function supplied, defaulting to `mean_se()
```




```
dev.off()
```

```
## null device
##           1
```

```
# Does not work...
# plot_symbiont_gene('molecular
# chaperone')


## Plot by exon, DOES NOT WORK AT THE
## MOMENT
plot_symbiont_gene_by_exon <- function(gene) {
    symbiont_v$E[str_detect(symbiont_v$genes$EntrezGeneID,
        "SymbC1.scaffold1.1.m1"), ] %>% as.data.frame() %>%
        mutate(exon = symbiont_v$genes$EntrezGeneID[str_detect(symbiont_v$genes$EntrezGeneID,
            "SymbC1.scaffold1.1.m1")]) %>%
        gather(sample, expn, -exon) %>% left_join(symbiont_v$targets %>%
        as.data.frame() %>% rownames_to_column("sample")) %>%
        ggplot(aes(x = Strain, y = expn,
            colour = CellType)) + geom_point(position = position_jitter(width = 0.2)) +
        stat_summary(colour = "black") +
        ggtitle(gene) + facet_wrap(~exon) +
        theme(legend.position = "none")
}
# exons of gene of choice
```

```
par(mar = c(5, 6, 3, 1))
layout(matrix(c(1, 2), 2, 2, byrow = TRUE))
# plot_symbiont_gene_by_exon('SymbC1.scaffold1.1.m1')
# plot_symbiont_gene_by_exon('aten_0.1.m1.4452.m1')
# plot_symbiont_gene_by_exon('aten_0.1.m1.24908.m1')
# plot_symbiont_gene_by_exon('aten_0.1.m1.17022.m1')

dev.off()
```

```
## null device
##           1
```

GO-Stats in ErmineJ

```
# write table for GO-stats
# GeneExpressionProfiles-Symbiont.txt
write.table(cbind(gene_symbiont_filtered2$genes[,
    -2], gene_symbiont_filtered2$counts),
    file = "GeneExpressionProfiles-Symbiont.txt",
    row.names = F, sep = "\t", quote = F)
#
write.table(cbind(gene_symbiont_filtered2$genes[,
    -2], gene_symbiont_v$E), file = "GeneExpressionProfiles-Symbiont-lcpm.txt",
    row.names = F, sep = "\t", quote = F)

# GeneScores-Symbiont.txt
GeneScores_Symbiont <- topTable(gene_symbiont_fit,
    n = Inf)
write.table(cbind(GeneScores_Symbiont$gene,
    GeneScores_Symbiont$adj.P.Val), file = "GeneScores-Symbiont.txt",
    sep = "\t", row.names = F, quote = F,
    col.names = F)

# GeneAnnotations-Symbiont.txt
GeneScores_Symbiont2 <- topTable(gene_symbiont_fit,
    n = Inf)
write.table(cbind(GeneScores_Symbiont2$gene,
    GeneScores_Symbiont$Annotation), file = "GeneAnnotations-Symbiont.txt",
    sep = "\t", row.names = F, quote = F,
    col.names = F)


# GeneExpressionProfiles-Host.txt
write.table(cbind(gene_host_filtered2$genes[,
    -2], gene_host_filtered2$counts), file = "GeneExpressionProfiles-Host.txt",
    row.names = F, sep = "\t", quote = F)
#
write.table(cbind(gene_host_filtered2$genes[,
    -2], gene_host_v$E), file = "GeneExpressionProfiles-Host-lcpm.txt",
    row.names = F, sep = "\t", quote = F)

# GeneScores-Host.txt
```

```
GeneScores_Host <- topTable(gene_host_fit,
    n = Inf)
write.table(cbind(GeneScores_Host$gene, GeneScores_Host$adj.P.Val),
    file = "GeneScores-Host.txt", sep = "\t",
    row.names = F, quote = F, col.names = F)

# GeneAnnotations-Host.txt
GeneScores_Host2 <- topTable(gene_host_fit,
    n = Inf)
write.table(cbind(GeneScores_Host2$gene,
    GeneScores_Host$Annotation), file = "GeneAnnotations-Host.txt",
    sep = "\t", row.names = F, quote = F,
    col.names = F)
```

---

3 Next steps

- Summarise to gene level counts and redo DE analysis
  - pretty much almost done, all data exported into big table
- Investigate DEXSeq for alternate exon usage
  - needs to be done
- Annotate with GO categories for functional analysis
  - progressing, currently computing InterPro scan for symbiont, host still waiting.
- Cluster expression and plot cluster patterns
  - needs to be done, Stephen?
- Do all the above with symbiont
  - pretty much almost done
- Symbiont/host interactions
  - MixOmics package, needs to be done
- Exon expression plots dont seem to make sense. Summarised in "Issues" tab.
  - Needs work
- WT does produce a lot of ROS... SS dont produce ROS. What can we find in the data to support this.. looking for SOD and other ROS scavengers. May be hypothesis testing by pulling out stress related genes? What was the code again for it please?
  - Code is there, needs to be done.
- Looking at the heatmaps, there are signature blocks of genes that are either highly upregulated or downregul for the respective genes. What are these genes and what are their function?
  - Code is there, needs to be done.
- We have SS8 strain (resistant) and WT10 strain (susceptible). Minimal evolution of three years between them, yet different thermal tolerance. We should be able to pinpoint the mutations (probably SNP analysis) and modulation of pathways that contribute to the differential resistance. We really have here for the first time an apple VS apple comparison (no apple vs orange any more).
  - Not there yet

---

4 Construction site

Heatmap of most significant genes across samples

The code below is a bit messy, sorry, I could not clean it up later. If you have a shortcut how to do the same, that would be great of course.. many thanks. Sometimes the error comes up when running the heatmap and

it fails to produce the graph (did not google it yet, will do tomorrow). But the heatmap usually works..:

Error in (function (filename = "Rplot%03d.png", width = 480, height = 480, : unable to start png() device

Oh, I think, I just realised that the heatmap is not on the logcounts, but looks like absolute values. I still need to fix that..

It may be nice to order them according to their average counts from high to low, so we would get a top block of all expressed genes for WT and a bottom block for all expressed genes for the SS strains. Will have another look into it. ***

Heatmap of most significant genes across samples (should be discriminant analysis, what genes distinguish WT10 from SS8?, at the moment it is across all samples).

```
# Heatmap of most significant genes across samples
# Colour definitions
library(RColorBrewer)
mypalette <- brewer.pal(11, "RdYlBu")
morecols <- colorRampPalette(mypalette)
col.cell <- c("purple", "orange")[host_samples$CellType]
column_labels <- paste(host_samples$CellType, host_samples$Strain, sep = ".")
column_labels <- paste(host_samples$SampleName)

# A)
# Extract top 500 most significant genes
# Extract labels to match them later to table with logcounts
top500signf <- topTable(gene_host_fit, n=500)
select_topsignf <- as.character(top500signf$Annotation)
select_topsignf <- as.character(top500signf$gene)
head(select_topsignf)

# B)
# NOT WORKING
# Extract genes of interest that are significantly different
# between SS8 and WT10, sort according to logvalues
gene_host_counts2 <- host_annotations %>% bind_cols(gene_host_counts[,-1])
design <- model.matrix(~Strain, data = gene_host_filtered2$samples)
colnames(design)
cont.matrix <- makeContrasts(SS8.vs.WT10 = (Intercept) - StrainStrain8, levels=design)
cont.matrix
names(gene_host_fit)
fit.cont <- contrasts.fit(gene_host_fit, cont.matrix)
fit.cont <- eBayes(fit.cont)
dim(fit.cont)
summa.fit <- decideTests(fit.cont)
summary(summa.fit)
SelectedGenesSignf <- topTable(fit.cont,coef="SS8.vs.WT10", number = Inf, p.value = 0.05,
                               sort.by="logFC") %>% filter(Annotation %>%
                               str_detect("[G|g]lutathione"))
select_topsignf2 <- as.character(SelectedGenesSignf$Annotation)

# C)
# NOT WORKING
# Extract genes of interest across all genes # NOT WORKING
gene_host_counts2 <- host_annotations %>% bind_cols(gene_host_counts[,-1])
SelectedGenesSignf <- topTable(gene_host_fit, number = Inf, p.value = 0.05) %>%
```

```
  filter(Annotation %>% str_detect("[H|h]eat|[G|g]lutathione|[C|c]haperone|[D|d]ismutase"))
#
select_topsignf3 <- as.character(SelectedGenesSignf$Annotation)
head(select_topsignf3)
###


## Subsetting table according to the new labels
# got some error messages that matrix is required, fixed like this
host_counts2 <- as.matrix(gene_host_counts, mode='any', row.names=1)
# had an error that first column should have been labels, fixed like this
host_counts3 <- data.frame(host_counts2[,-1], row.names=host_counts2[,1])
# subsetting of matrix worked
highly_signf_lcpm <- host_counts3[select_topsignf, ]
# had to convert matrix into numeric matrix
highly_signf_lcpm2 <- sapply(highly_signf_lcpm, as.numeric)
dim(highly_signf_lcpm2)

# make heatmap
heatmap.2(
  highly_signf_lcpm2,
  Rowv = FALSE,
  Colv = FALSE,
  col = rev(morecols(50)),
  trace = "none",
  main = "Selected genes",
  ColSideColors = col.cell,
  scale = "row",
  labCol = column_labels,
  #fix stuff that falls off the image with the following line
  margins = c(10,5)
  # remove junk on side
  # labRow = ""
)
```

```
# Packages
library(limma)
library(edgeR)
library(Glimma)
library(gplots)
library(org.Mm.eg.db)
library(RColorBrewer)
library(BiasedUrn)
library(openxlsx)
library(vegan)
library(rgl)
library(ape)
library(openxlsx)
library(ggplot2)

# Parameters
SampleGroup<-"Host"
#SampleGroup<-"Host-STAR"
time<-format(Sys.time(),"%d-%m-%Y")
```

```
#
# LOAD MATRIX
# Only heat stress realted genes
highly_variable_lcpm_NEW <- read.delim("data/Heatmaps/Host-NewFilter-Top500.txt", row.names=1)
highly_variable_lcpm_NEWNEW <- as.matrix(highly_variable_lcpm_NEW, mode='any')
#
#
# SELECTION IN R DOES NOT WORK - see var_genes file is a value with attributes, not sure how to convert
#library(dplyr)
#TEST <- topTable(fit.cont, n=Inf, coef="SS8.vs.WT10")
#TEST2 <- subset(TEST, select = c(5))
#head(TEST2)
#TEST2NEW <- as.matrix(TEST2, mode='numeric')
#select_var_NEW <- names(sort(TEST2NEW, decreasing = TRUE)) [1:500]
#head(select_var_NEW)
#
# Make new heat map
TitleFile <- paste(SampleGroup,"Top 500")
Title <- paste(SampleGroup,"- Top 500 most significant genes across samples")
FileName <- paste(TitleFile,time,".jpg")
jpeg(file=FileName, width = 4000, height = 3280, res = 450)
par(mfrow = c(1,1))
#
lmat = rbind(0:3,2:1,4:4)
#lmat = rbind(c(0,3),c(2,1),c(0,4))
lwid = c(1.5,4)
lhei = c(1.5,4,1)
# lmat
#
mypalette <- brewer.pal(11, "RdYlBu")
morecols <- colorRampPalette(mypalette)
col.cell <- c("purple", "orange")[host_samples$CellType]
column_labels <- paste(host_samples$CellType, host_samples$Strain, sep = ".")
column_labels <- paste(host_samples$SampleName)
#
heatmap.2(
  highly_variable_lcpm_NEWNEW,
  #Rowv = FALSE,
  #Colv = FALSE,
  col = rev(morecols(50)),
  trace = "none",
  key.title = NA,
  keysize = 1,
  key.par=list(mgp=c(1, 0.5, 0), mar=c(5, 2, 2, 1)),
  #lmat = lmat,
  #lmat=rbind(c(0, 4, 2), c(0, 1, 3)), lhei=c(2.5, 10), lwid=c(1, 10, 1),
  main = " ",
  ColSideColors = col.cell,
  scale = "row",
  labCol = column_labels,
  ## fix stuff that falls off the image with the following line
  margins = c(8,32),
  ## remove junk on side
```

```
  labRow = ""
)
title(Title, line = +3.0, adj = +0.4)

dev.off()
```

Heatmap trial

```
library(gplots)

# IT SHOULD WORK, TAKING THE logcpm
# values from the vomm model from $E but
# it does seem to be meaingful..
basal.vs.lp.topgenes <- host_strain_3_up$gene[1:5]
i <- which(gene_host_v$genes$gene %in% basal.vs.lp.topgenes)
mycol <- colorpanel(1000, "blue", "white",
    "red")
heatmap.2(lcpm[i, ], scale = "row", labRow = gene_host_v$genes$Annotation[i],
    col = mycol, trace = "none", density.info = "none",
    margin = c(8, 15), lhei = c(2, 10), dendrogram = "column")
```