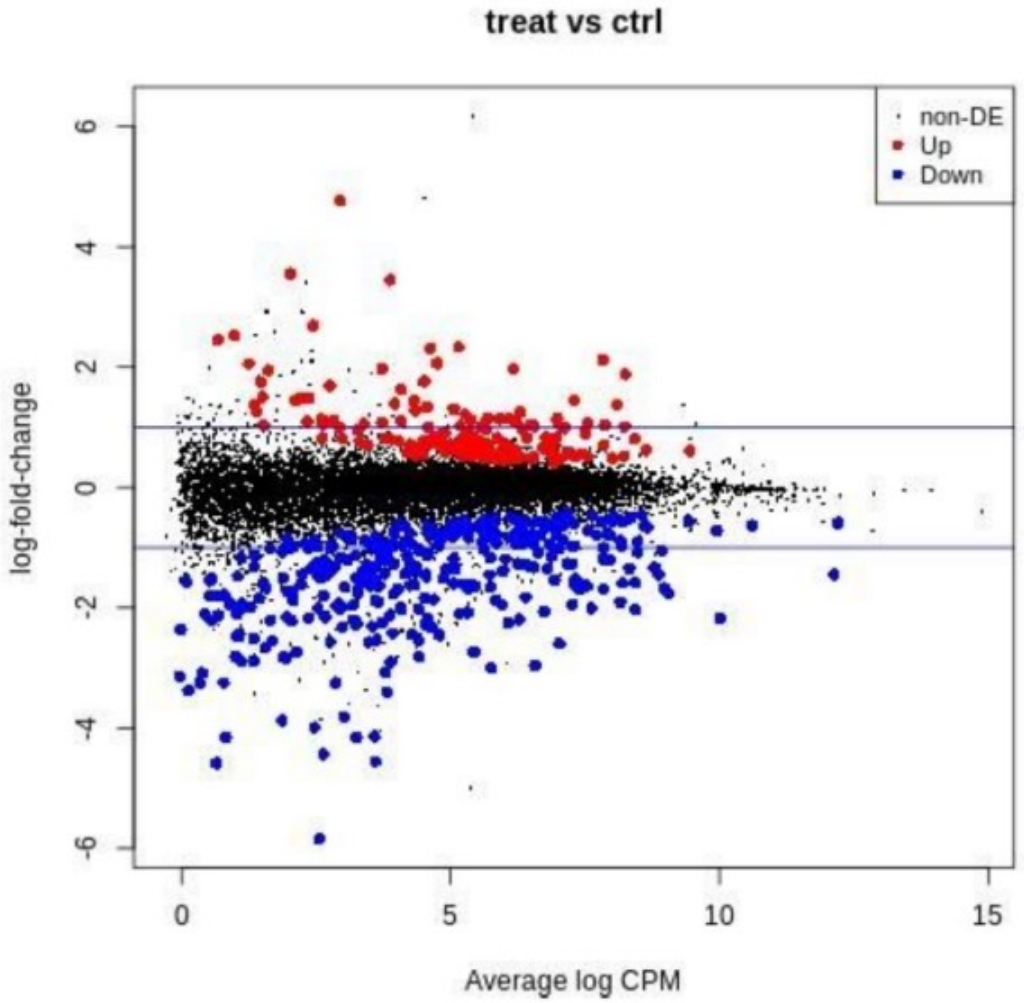
Exact tests often are a good place to start with differential expression analysis of genomic data sets.



Example mean difference (MD) plot of exact test results for the E05 *Daphnia* genotype.

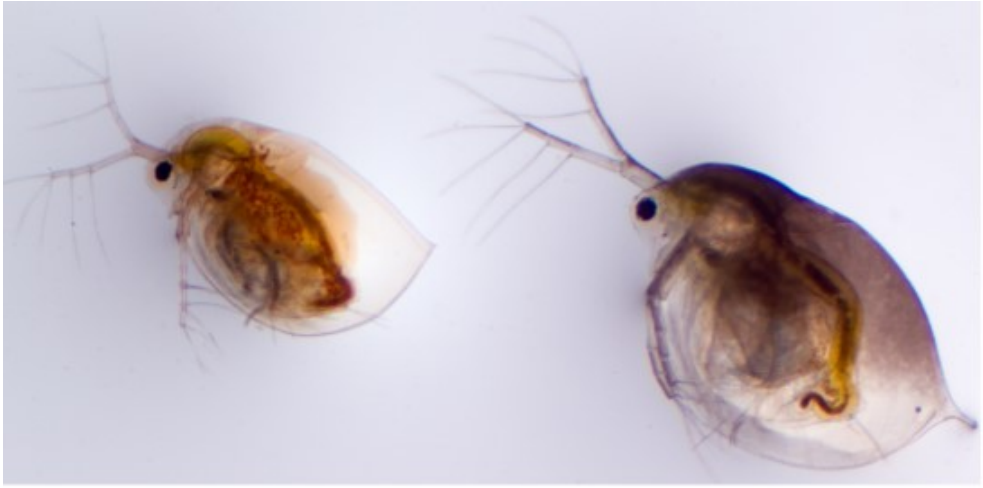
As usual, the types of contrasts you can make will depend on the design of your study and data set. In the following example we will use the raw counts of differentially expressed (DE) genes to compare the following *Daphnia* genotypes.

Genotype	Control	Treatment	Replicates
Olympic Y023 melanica	VIS	UV	3
Olympic R2 melanica	VIS	UV	3
Olympic Y05 melanica	VIS	UV	3
Olympic E05 melanica	VIS	UV	3
Sierra melanica	VIS	UV	3
PA pulex	VIS	UV	3

The design of our experiment is described by three replicates of ultra-violet radiation (UV) treatment, and three replicates of visible light (VIS) control for each of the *Daphnia* genotypes.

After normalization of raw counts we will perform genewise exact tests for differences in the means between two groups of gene counts. Specifically, the two experimental groups of treatment and control for the E05

Daphnia genotype.



D. melanica from the Olympic mountains in WA (left) and the Sierra Nevada mountains in CA (right).

Exact Test and Plotting Script – exactTest_edgeR.r

The following R script will be used to normalize raw gene counts produced by a quantification program such as Hisat2 or Tophat2. Then, we will perform exact tests for a specified genotype data set and plot both intermediate and final results.

We will use several different plotting functions to generate informative plots of our input data sets and exact test results with edgeR.

- **barplot** – Bar plots of sequence read library sizes
- **plotMDS** – Multi-dimensional scaling (MDS) scatterplot of distances between the samples
- **heatmap** – Heatmaps of moderated log counts per million (logcpm)
- **plotBCV** – Genewise biological coefficient of variation (BCV) plots against gene abundance
- **plotMD** – Mean difference (MD) plots of all log fold change (logFC) against average count size
- **plotSmear** – MA plots of log-intensity ratios (M-values) versus log-intensity averages (A-values)

Script Inputs

A csv formatted file of raw gene counts is expected as an input to the script. Below is an example raw gene count table where columns are samples, rows are gene IDs, and raw counts are represented by “N#” for simplicity here.

Additionally, the column numbers for the range of samples you wish to perform exact tests on need to be specified. So for example, the column numbers 1 and 6 should be input to perform an exact test on the E05 *Daphnia* genotype for the example raw gene count table above. This selects all replicates of the UV treatment and VIS control for the E05 genotype.

R Script

First, we should add some comments to the top of the R script describing how to run the script. We should also add a comment with a reminder of the purpose of the script.

```
#!/usr/bin/env Rscript
#Usage: Rscript exactTest_edgeR.r rawGeneCountsFile startColumn endColumn
#Usage Ex: Rscript exactTest_edgeR.r daphnia_rawGeneCounts_htseq.csv 1 6
#R script to perform exact test DE analysis of raw gene counts using edgeR
```

Before proceeding with plotting and exact tests we will need to import the edgeR library.

```
#Load the edgeR library
library("edgeR")
```

Next, we retrieve the input path for our raw gene counts and the column range for the exact test. The number of input arguments to the script are verified to help ensure that the required data is input to the script.

```
#Retrieve inputs to the script
args = commandArgs(trailingOnly=TRUE)
#Test if there is one input argument
if (length(args)!=3) {
  stop("One file name and a range of columns must be supplied.n", call.=FALSE)
}
```

Now we can import our data with the input path to our raw gene counts and the column range for the sample we want to test.

```
#Import gene count data
countsTable <- read.csv(file=args[1], row.names="gene")[ ,args[2]:args[3]]
head(countsTable)
```

The samples that represent our treatment and control need to be specified as a grouping factor to perform the exact tests with edgeR. In our example input data set the samples are ordered as:

1. Treatment replicate 1
2. Treatment replicate 2
3. Treatment replicate 3
4. Control replicate 1
5. Control replicate 2
6. Control replicate 3

```
#Add grouping factor
group <- factor(c(rep("treat",3),rep("ctrl",3)))
#Create DGE list object
list <- DGEList(counts=countsTable,group=group)
```

This is a good point to generate some interesting plots of our input data set before we begin preparing the raw gene counts for the exact test.

First, we will plot the library sizes of our sequencing reads before normalization using the **barplot** function. The resulting plot is saved to the **plotBarsBefore** jpg file in your working directory.

```
#Plot the library sizes before normalization
jpeg("plotBarsBefore.jpg")
barplot(list$samples$lib.size*1e-6, names=1:6, ylab="Library size (millions)")
dev.off()
```

Next, we will use the **plotMDS** function to display the relative similarities of the samples and view batch and treatment effects before normalization. The resulting plot is saved to the **plotMDSBefore** jpg file in your working directory.

```
#Draw a MDS plot to show the relative similarities of the samples
jpeg("plotMDSBefore.jpg")
plotMDS(list, col=rep(1:3, each=3))
dev.off()
```

It can also be useful to view the moderated log-counts-per-million before normalization using the **cpm** function results with **heatmap**. The resulting plot is saved to the **plotHeatMapBefore** jpg file in your working directory.

```
#Draw a heatmap of individual RNA-seq samples
jpeg("plotHeatMapBefore.jpg")
logcpm <- cpm(list, log=TRUE)
heatmap(logcpm)
dev.off()
```

There is no purpose in analyzing genes that are not expressed in either experimental condition (treatment or control), so raw gene counts are first filtered by expression levels. The normalized gene counts are output to the **stats_normalizedCounts** csv file.

```
#Filter raw gene counts by expression levels
keep <- filterByExpr(list)
table(keep)
list <- list[keep, , keep.lib.sizes=FALSE]
#Calculate normalized factors
list <- calcNormFactors(list)
#Write normalized counts to file
normList <- cpm(list, normalized.lib.sizes=TRUE)
write.table(normList, file="stats_normalizedCounts.csv", sep=",",
row.names=TRUE)
#View normalization factors
list$samples
dim(list)
```

Now that we have normalized gene counts for our samples we should generate the same set of previous plots for comparison.

First, we plot the library sizes of our sequencing reads after normalization using the **barplot** function. The resulting plot is saved to the **plotBarsAfter** jpg file in your working directory.

```
#Plot the library sizes after normalization
jpeg("plotBarsAfter.jpg")
barplot(list$samples$lib.size*1e-6, names=1:6, ylab="Library size (millions)")
dev.off()
```

Next, we will use the **plotMDS** function to display the relative similarities of the samples and view batch and treatment effects after normalization. The resulting plot is saved to the **plotMDSAfter** jpg file in your working directory.

```
#Draw a MDS plot to show the relative similarities of the samples
jpeg("plotMDSAfter.jpg")
plotMDS(list, col=rep(1:3, each=3))
dev.off()
```

It can also be useful to view the moderated log counts per million (logcpm) after normalization using the **cpm** function results with **heatmap**. The resulting plot is saved to the **plotHeatMapAfter** jpg file in your working directory.

```
#Draw a heatmap of individual RNA-seq samples
jpeg("plotHeatMapAfter.jpg")
logcpm <- cpm(list, log=TRUE)
heatmap(logcpm)
dev.off()
```

With the normalized gene counts we can also produce a matrix of pseudo-counts to estimate the common and tagwise dispersions. This allows us to use the **plotBCV** function to generate a genewise biological coefficient of variation (BCV) plot of dispersion estimates. The resulting plot is saved to the **plotBCVResults** jpg file in your working directory.

```
#Produce a matrix of pseudo-counts
list <- estimateDisp(list)
```

```
list$common.dispersion
#View dispersion estimates and biological coefficient of variation
jpeg("plotBCV.jpg")
plotBCV(list)
dev.off()
```

Finally, we are ready to perform exact tests with edgeR using the **exactTest** function. The resulting table of differentially expressed (DE) genes are written to the **stats_exactTest** csv file to your working directory.

```
#Perform an exact test for treat vs ctrl
tested <- exactTest(list, pair=c("ctrl", "treat"))
topTags(tested)
#Create results table of DE genes
resultsTbl <- topTags(tested, n=nrow(tested$table))$table
#Output resulting table
write.table(resultsTbl, file="stats_exactTest.csv", sep=",", row.names=TRUE)
```

Using the resulting DE genes from the exact test we can view the counts per million for the top genes of each sample.

```
#Look at the counts per million in individual samples for the top genes
o <- order(tested$table$PValue)
cpm(list)[o[1:10],]
#View the total number of differentially expressed genes at 5% FDR
summary(decideTests(tested))
```

We can also generate a mean difference (MD) plot of the log fold change (logFC) against the log counts per million (logcpm) using the **plotMD** function. DE genes are highlighted and the blue lines indicate 2-fold changes. The resulting plot is saved to the **plotMDResults** jpg file in your working directory.

```
#Make a MD plot of logFC against logcpm
jpeg("plotMDResults.jpg")
plotMD(tested)
abline(h=c(-1, 1), col="blue")
dev.off()
```

As a final step, we will produce a MA plot of the libraries of count data using the **plotSmear** function. There are smearing points with very low counts, particularly those counts that are zero for one of the columns. The resulting plot is saved to the **plotMAResults** jpg file in your working directory.

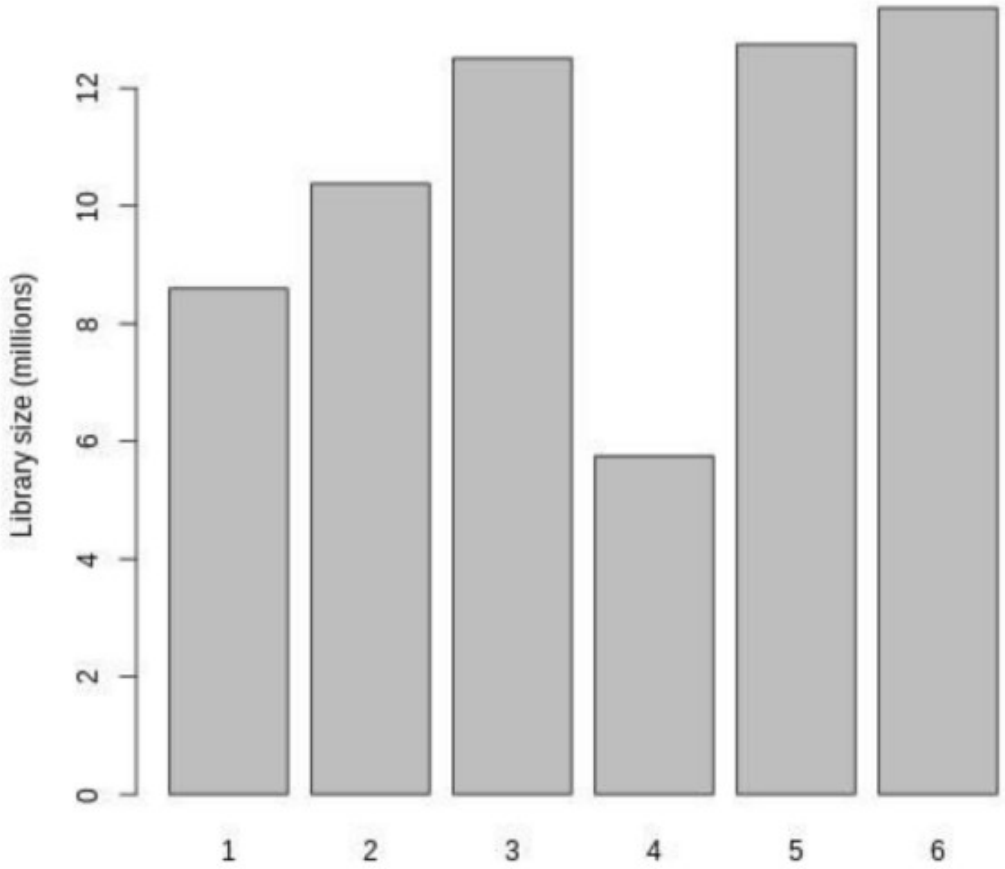
```
#Make a MA plot of the libraries of count data
jpeg("plotMAResults.jpg")
plotSmear(tested)
dev.off()
```

exacttest_edger.rDownload

Example Outputs – exactTest_edgeR.r

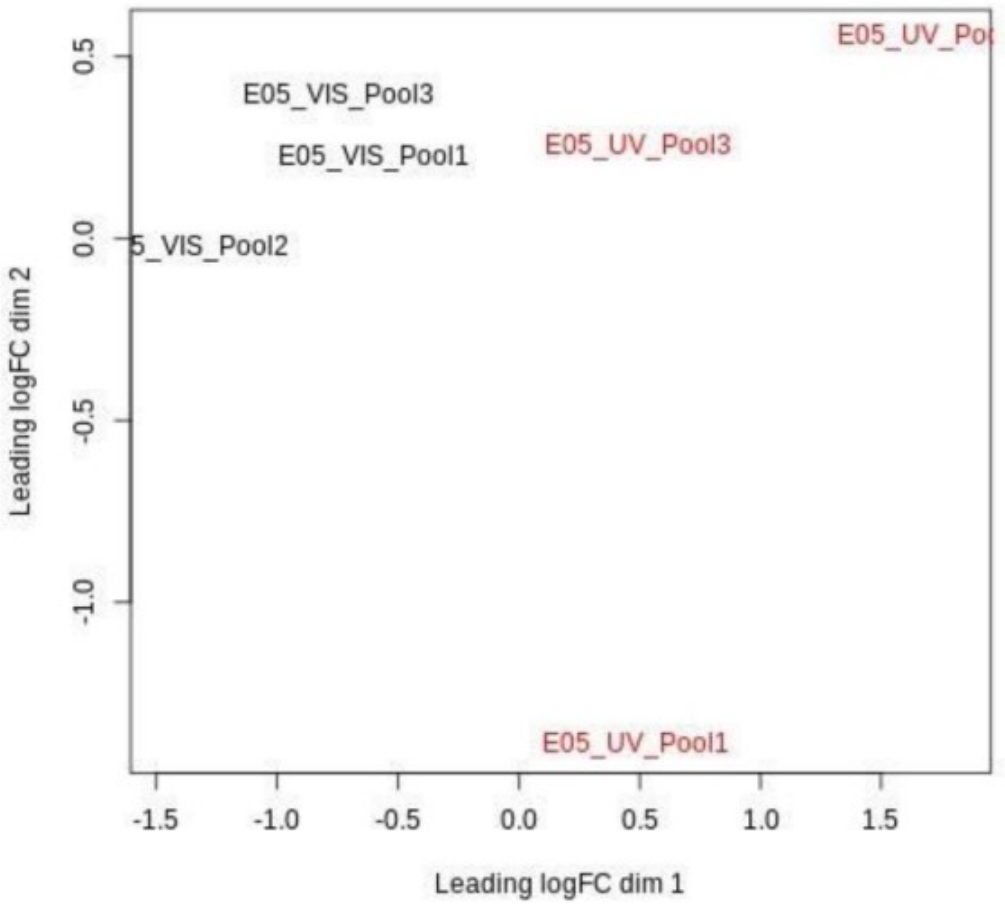
The following plots are example outputs from the above R script.

plotBarsAfter.jpg



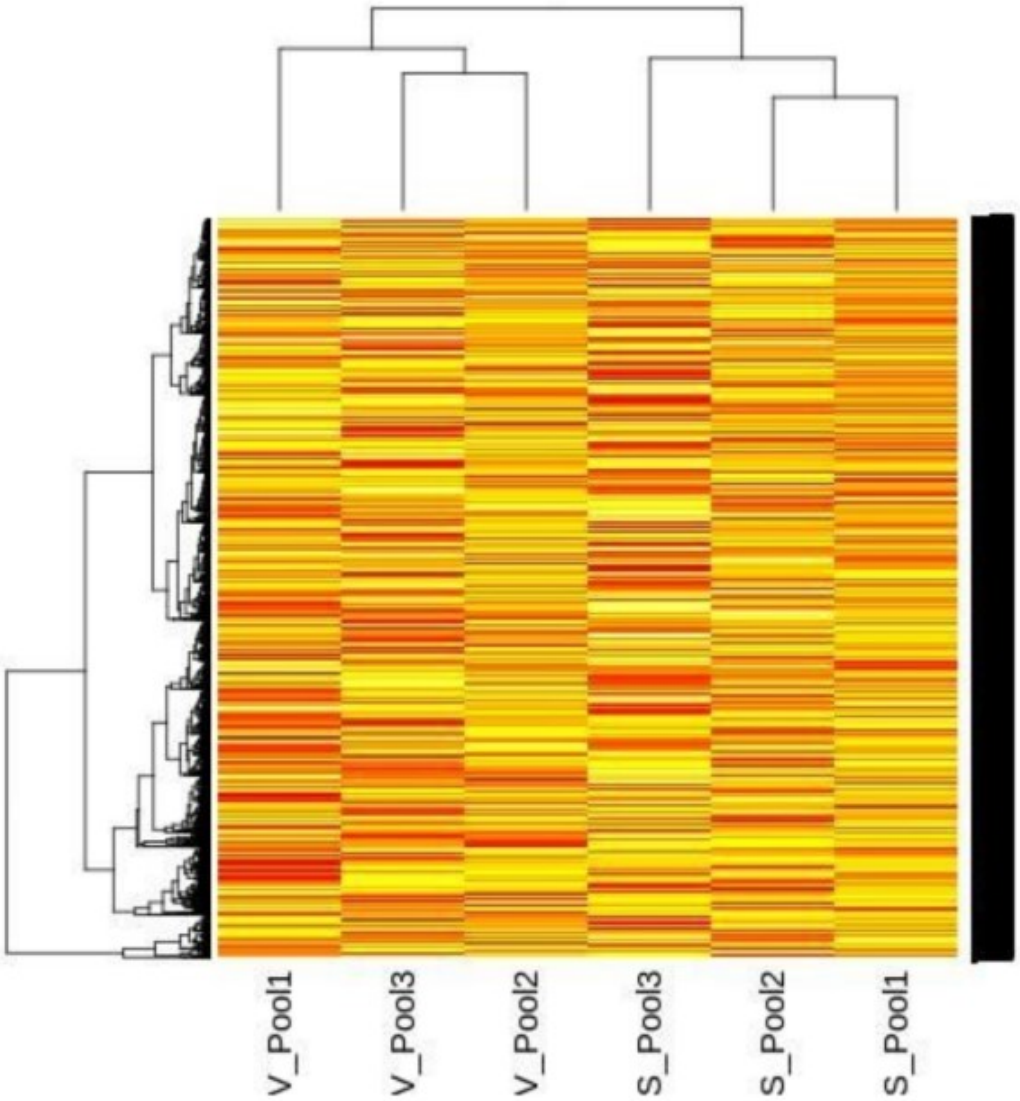
Bar plot the library sizes of E05 sequencing reads after normalization.

plotMDSAfter.jpg



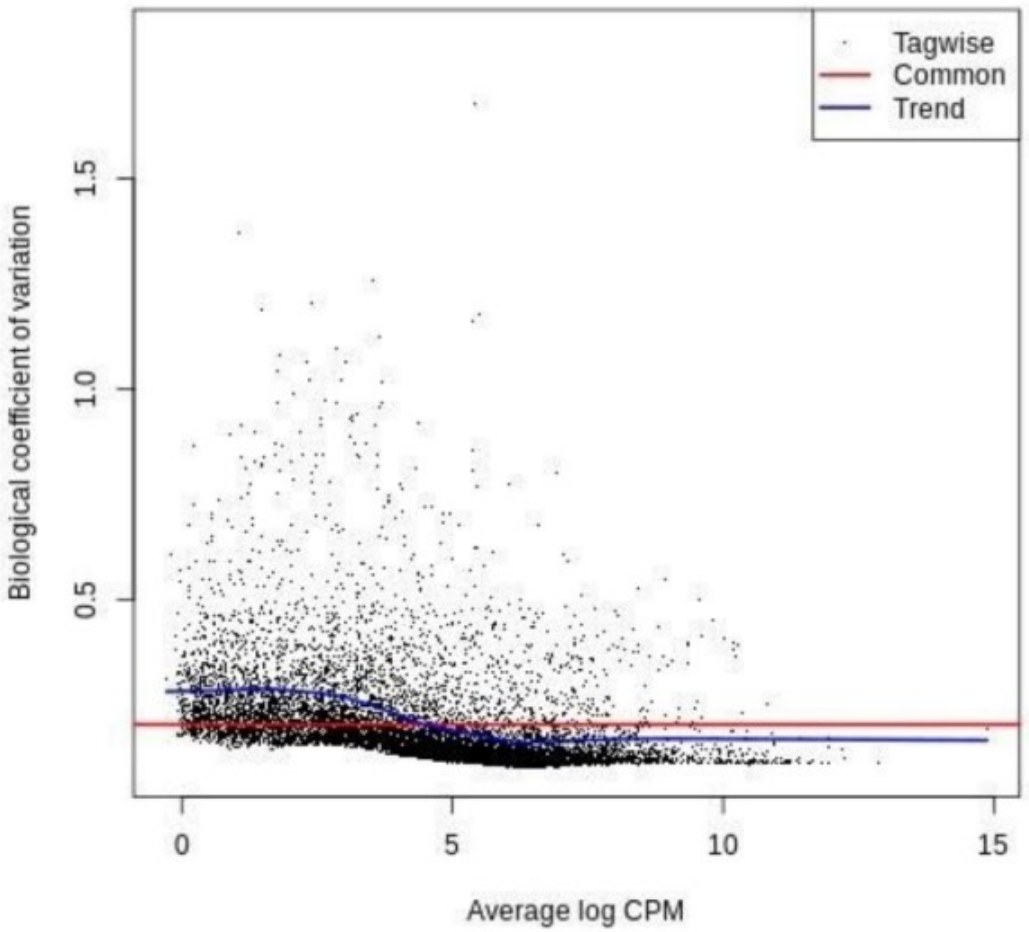
MDS plot of the relative similarities of E05 samples after normalization.

plotHeatMapAfter.jpg



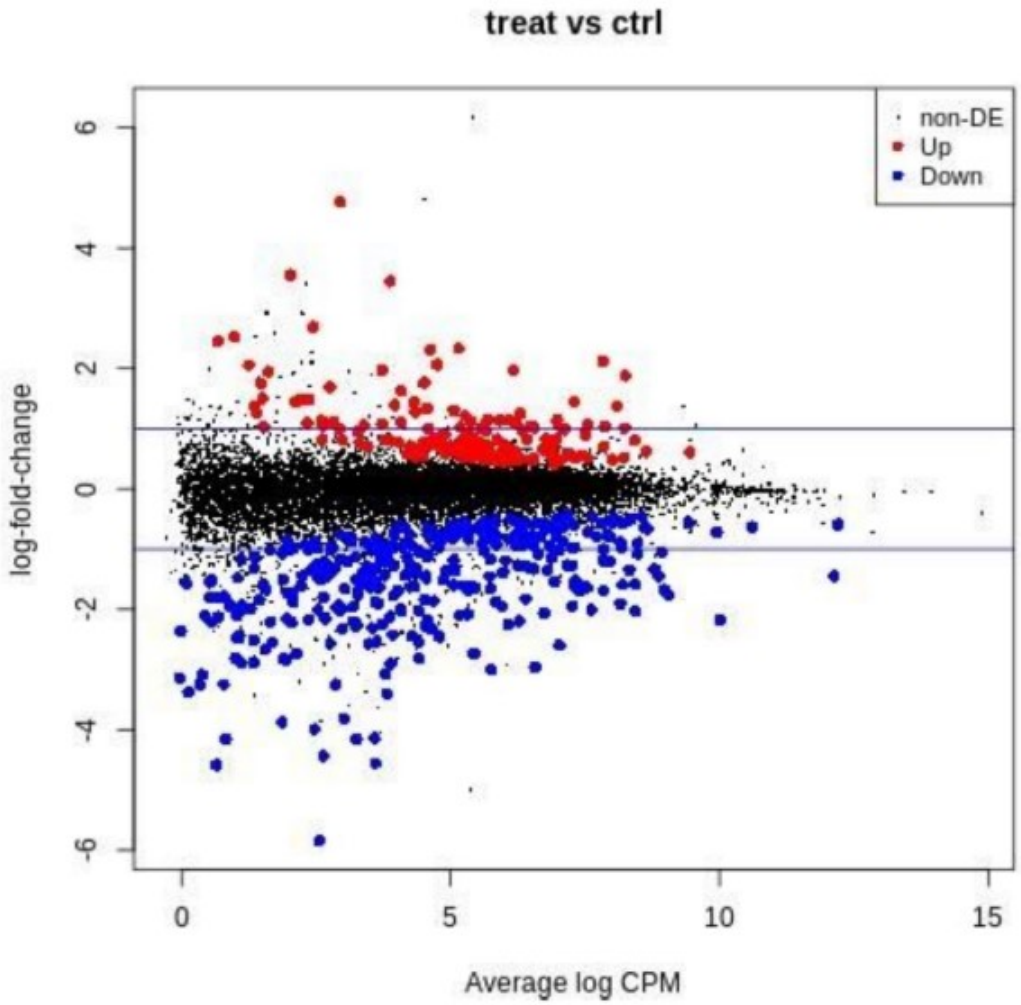
Heat map of moderated log counts per million (logcpm) for the E05 genotype after normalization.

plotBCVResults.jpg



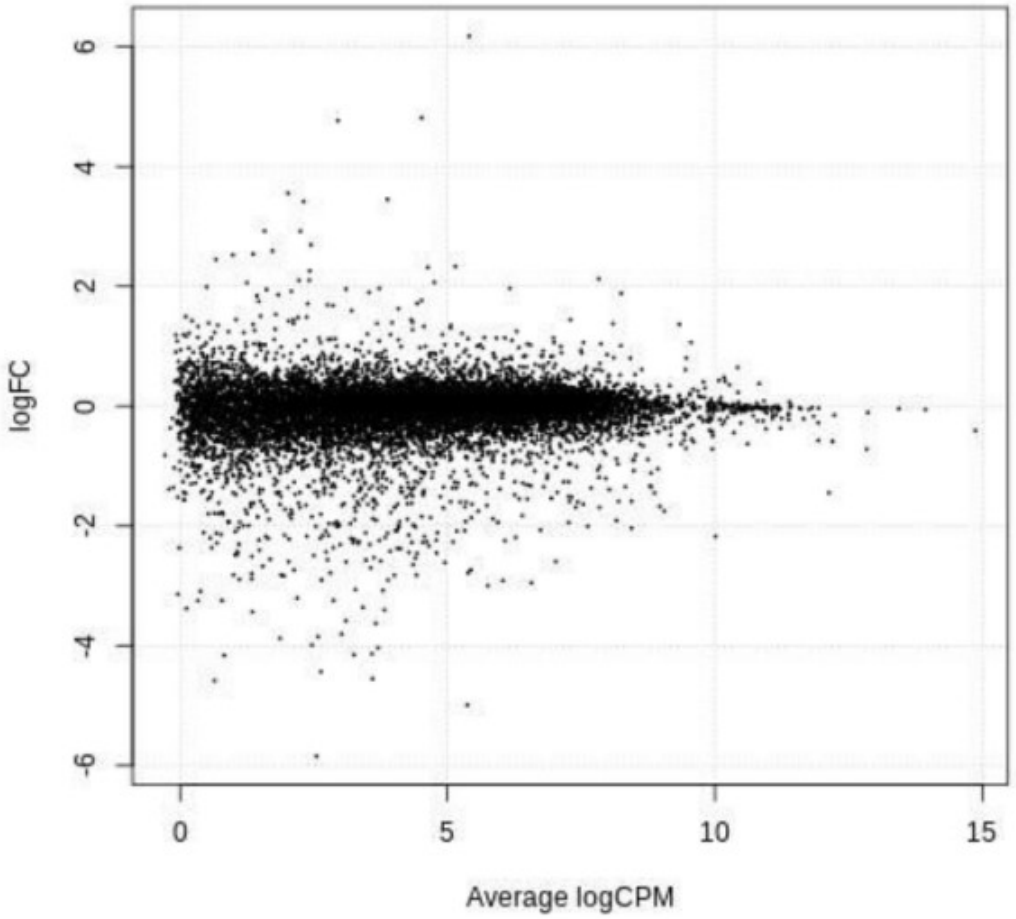
Biological coefficient of variation (BCV) plot of dispersion estimates for the E05 genotype.

plotMDResults.jpg



Mean difference (MD) plot of the log fold change (logFC) against the log counts per million (logcpm) for the E05 genotype.

plotMAResults.jpg



MA plot of the libraries of count data for the E05 genotype.