# NGS and data analysis – Report

Filip Hajdyła

February 16, 2023

## Introduction

Next Generation Sequencing (NGS) is a set of techniques allowing to gather and analyse genomic and transcriptomic data. In the following report transcriptomic (RNA sequencing) data will be analysed. During the analysis a significance level of $\alpha = 0.05$ was assumed.

```r
sign.level <- 0.05
```

## Analysis

### Technical information

In the following analysis `R 4.1.2` was used along with the server version of `Rstudio`. In order to extend functionality of `R` additional libraries were used. Additionally `knitr` library was used to display tables and figures present in this report.

```r
library(Rsubread)    # data alignment
library(limma)       # DGE
library(edgeR)       # DGE
library(DESeq2)      # DGE
library(knitr)       # generating tables and general formatting
```

### Building index and mapping

At first `Rsubread` library was used for building a base-space index for reference sequence. `TAIR9.fa` file contents were used as a reference sequence and the index has been named `TAIR9g`.

```r
buildindex(
  basename = "TAIR9g",
  reference = "data/TAIR9.fa"
)
```

After building the index, reads were mapped onto the reference. As `.fastq` files contain data about reads, they have been used as input for `subjunc()` function. The outputs were `.bam` files later used for counting features.

```r
# mapping WT_R1
subjunc(
  index = "TAIR9g",
  "data/WT_R1.fastq",
  output_file = "WT_R1.bam",
  nthreads = 4,
  reportAllJunctions = TRUE,
)
```

```r
# mapping WT_R2
subjunc(
  index = "TAIR9g",
  "data/WT_R2.fastq",
  output_file = "WT_R2.bam",
  nthreads = 4,
  reportAllJunctions = TRUE,
)
# mapping OE_1_R1
subjunc(
  index = "TAIR9g",
  "data/OE_1_R1.fastq",
  output_file = "OE_1_R1.bam",
  nthreads = 4,
  reportAllJunctions = TRUE,
)
# mapping OE_1_R2
subjunc(
  index = "TAIR9g",
  "data/OE_1_R2.fastq",
  output_file = "OE_1_R2.bam",
  nthreads = 4,
  reportAllJunctions = TRUE,
)
```

### Counting genomic features

The reason for running `featureCounts()` is assigning sequence reads to genomic features (genomic region with some annotated function). Previously generated `.bam` files were used to count genomic features.

```r
fc <- featureCounts(
  c(
    "WT_R1.bam",
    "WT_R2.bam",
    "OE_1_R1.bam",
    "OE_1_R2.bam"
    ),
  annot.ext = "data/TAIR9.gtf",
  isGTFAnnotationFile = TRUE,
  juncCounts = TRUE,
  nthreads = 4
)
```

The output was an object variable containing data about reads assigned to features and assignment statistics such as seen below.

Table 1: Read counts for the first five loci in each sample.

| | WT_R1.bam | WT_R2.bam | OE_1_R1.bam | OE_1_R2.bam |
|---|---:|---:|---:|---:|
| AT1G01010 | 128 | 91 | 36 | 70 |
| AT1G01020 | 150 | 140 | 72 | 155 |
| AT1G01030 | 49 | 35 | 33 | 42 |
| AT1G01040 | 333 | 394 | 194 | 455 |
| AT1G01050 | 691 | 499 | 295 | 471 |

Table 2: Total counts in each sample sorted by status.

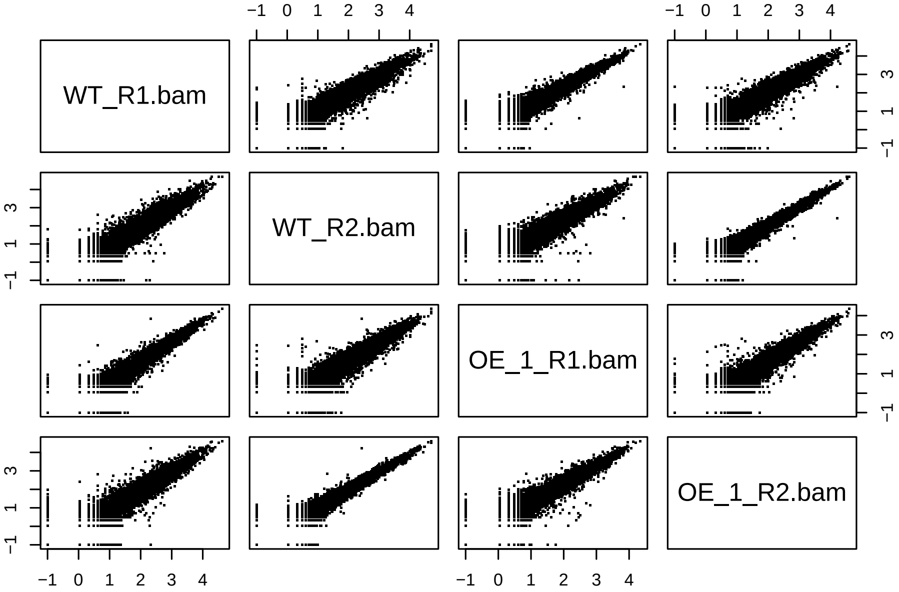
| Status | WT_R1.bam | WT_R2.bam | OE_1_R1.bam | OE_1_R2.bam |
|---|---|---|---|---|
| Assigned | 7822452 | 7869452 | 3897525 | 7358560 |
| Unassigned_Unmapped | 1784035 | 1744583 | 941327 | 2276256 |
| Unassigned_Read_Type | 0 | 0 | 0 | 0 |
| Unassigned_Singleton | 0 | 0 | 0 | 0 |
| Unassigned_MappingQuality | 0 | 0 | 0 | 0 |
| Unassigned_Chimera | 0 | 0 | 0 | 0 |
| Unassigned_FragmentLength | 0 | 0 | 0 | 0 |
| Unassigned_Duplicate | 0 | 0 | 0 | 0 |
| Unassigned_MultiMapping | 0 | 0 | 0 | 0 |
| Unassigned_Secondary | 0 | 0 | 0 | 0 |
| Unassigned_NonSplit | 0 | 0 | 0 | 0 |
| Unassigned_NoFeatures | 271136 | 257346 | 124273 | 246125 |
| Unassigned_Overlapping_Length | 0 | 0 | 0 | 0 |
| Unassigned_Ambiguity | 122377 | 128619 | 62475 | 119059 |

Majority of sequences in all of the samples were assigned to genomic features. However, there are some fragments that are either unmapped and therefor cannot be assigned, do not overlap any features, overlap two or more features or overlap so called meta-features.

## Visualisation

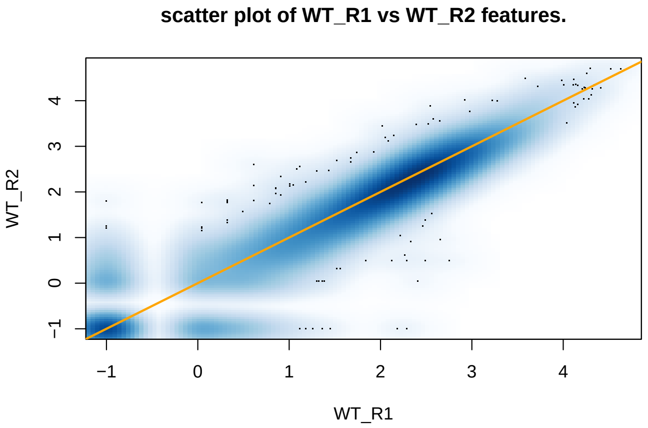
The first step in visualising data was creating batch plots in order to get an overview of relations between samples. Each point on each plot represents a single genomic site and has X and Y values equal to logarithm of relevant counts.

```
c <- data.frame(fc$counts)
pairs(log10(c + 0.1), pch=".")
```

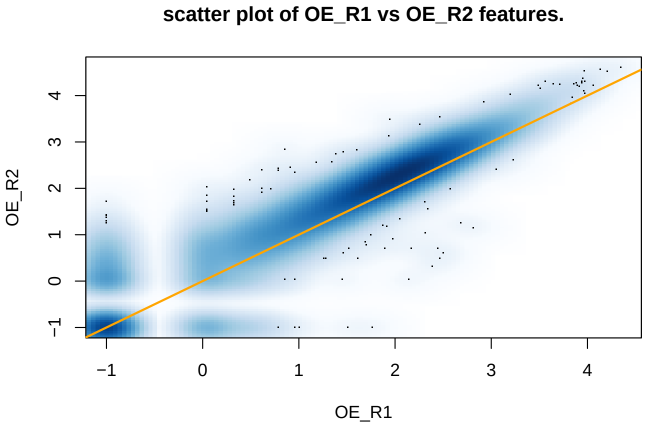


Next, smooth scatter (heatmap) plots comparing two WT samples as well as WT to OE were created. Heatmap was used in order to avoid overlapping points.

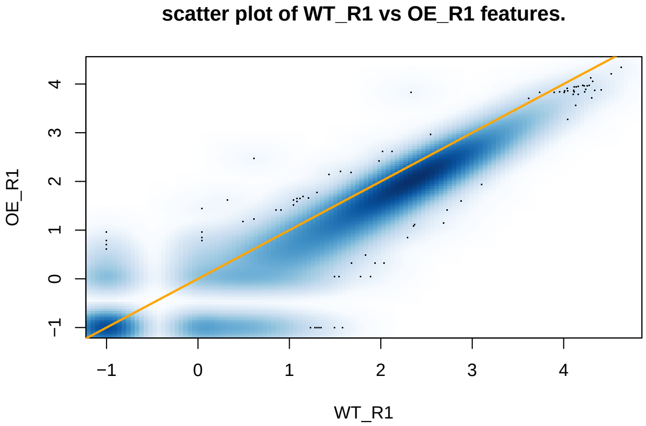
```
smoothScatter(
  x = log10(c$WT_R1.bam + 0.1),
  y = log10(c$WT_R2.bam + 0.1),
  xlab = "WT_R1",
  ylab = "WT_R2",
  main = "scatter plot of WT_R1 vs WT_R2 features.",
  pch = "."
)
abline(a=0,b=1,col="orange",lwd=2)
```



```
smoothScatter(
  x = log10(c$OE_1_R1.bam + 0.1),
  y = log10(c$OE_1_R2.bam + 0.1),
  xlab = "OE_R1",
  ylab = "OE_R2",
  main = "scatter plot of OE_R1 vs OE_R2 features.",
  pch = "."
)
abline(a=0,b=1,col="orange",lwd=2)
```



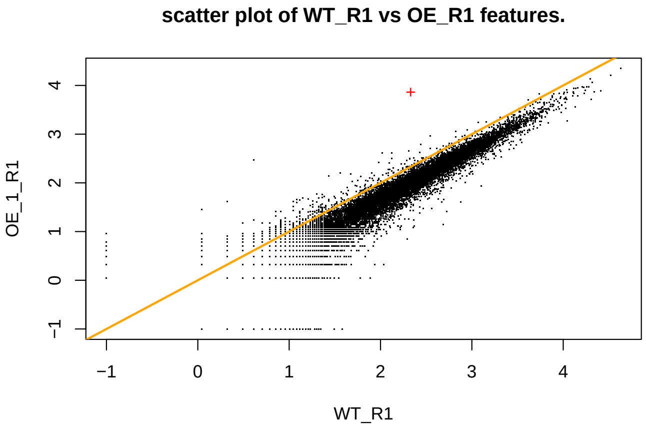
```
smoothScatter(
  x = log10(c$WT_R1.bam + 0.1),
  y = log10(c$OE_1_R1.bam + 0.1),
  xlab = "WT_R1",
  ylab = "OE_R1",
  main = "scatter plot of WT_R1 vs OE_R1 features.",
  pch = "."
)
abline(a=0,b=1,col="orange",lwd=2)
```



Then AT3G01150 locus was emphasized on the scatter plot as an example. It is depicted with a red x on the figure below.

```
gene.sel <- "AT3G01150"
colors <- rep("black", times = dim(c)[1])
pchs <- rep(".", times = dim(c)[1])
names(pchs) <- names(colors) <- rownames(c)
colors[gene.sel] <- "red"
pchs[gene.sel] <- "+"

plot(
  x = log10(c$WT_R1.bam + 0.1),
  y = log10(c$OE_1_R1.bam + 0.1),
  xlab = "WT_R1",
  ylab = "OE_1_R1",
  main = "scatter plot of WT_R1 vs OE_R1 features.",
  pch = pchs,
  col = colors
)
abline(a=0,b=1,col="orange",lwd=2)
```



All pairs show more or less linear relation. There is however a slight decrease of counts in OE samples compared to WT which is indicated by displacement of points relative to the orange line ($y = x$). Also, the expression in OE__R2 sample is generally a little bit higher than in OE__R1.

## Differential gene expression (DGE) analysis

### limma

The first step of this part of analysis was to construct a new data set `genes.merged` with defined column names. In further analyses the same data sets are used as for `limma` analysis.

```
genes.merged <- fc$counts[, c(3,4,1,2)] # change col order
colnames(genes.merged) <- c(
  "OE_1_R1",
  "OE_1_R2",
  "WT_R1",
  "WT_R2"
)
```

Then all samples were split into two categories: OE and WT and contrasts between the two were defined as `cm` variable using `limma::makeContrasts()`.

```
samples <- substr(colnames(genes.merged), 0, 2) # just like python slices
design <- data.frame(
  OEs = ifelse(samples == "OE", 1, 0),
  WTs = ifelse(samples == "WT", 1, 0)
)
cm <- makeContrasts(OEvsWT=OEs-WTs, levels=design)
print(cm)
```

```
##       Contrasts
## Levels OEvsWT
##    OEs      1
##    WTs     -1
```

Diferentially expressed genes were first assigned to `dge` variable. These were then normalized using `edgeR::calcNormFactors()` (TMM normalization). TMM normalization adjusts library sizes based on the assumption that most genes are not diferentially expressed. It ensures that the expression values are comparable between sequences. The `limma::voom()` transformation allows creation of multidimentional matrix containing weigth values. These values are then used by `limma::lmFit()` to create a linear model. After that `limma::contrasts.fit()` was used to calculate coefficients for a given matrix and design. Bayes correction (`limma::eBayes`) smoothed out standard errors. The `limma::topTable()` created a table with top rated genes. Benjamini-Hoechberg method was used in the means of $p$-value correction in order to get rid of potential false positives. At last, genes with adjusted $p$-value $< 0.05$ were selected and assigned to rows in the `sign.genes` data frame.

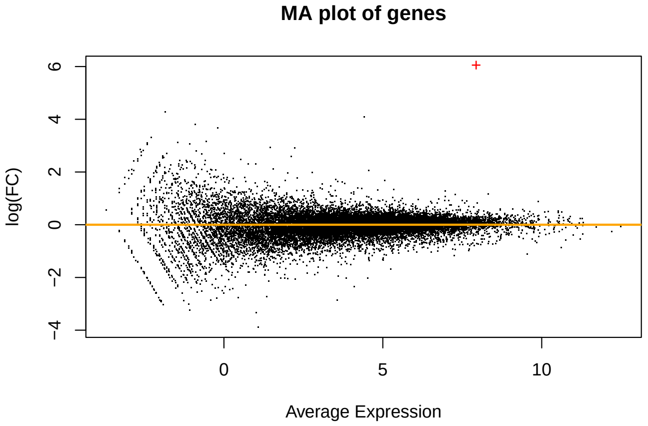
```
# TMM normalization
dge <- DGEList(counts = genes.merged)
dge <- calcNormFactors(dge)
# voom transformation
v <- voom(dge, design, plot=FALSE)
# linear model fit with limma
f.t <- lmFit(v, design)#, method="robust", maxit=9999
# contrasts fit
cf <- contrasts.fit(f.t, cm)
# Bayes corr
fe <- eBayes(cf, proportion = 0.01)
# Multiple testing corr
limma.countsTMMvoom.genes <- topTable(
  fe,
  number = Inf,
  adjust.method = "BH",
  sort.by = "none"
)
sign.genes <- limma.countsTMMvoom.genes[
  which(limma.countsTMMvoom.genes$adj.P.Val < sign.level),
]
```

Table 3: Signifficantly different ($P_{adj} < 0.05$) expression.

| | logFC | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| AT3G01150 | 5.999203 | 7.938799 | 16.26258 | 1.8e-06 | 0.0437794 | -4.147031 |

As seen above, only one gene (AT3G01150) is significantly differentiating. This gene is indicated using a red + on a MA plot of OE vs WT below.

```
pchs <- rep(".", dim(limma.countsTMMvoom.genes)[1])
colors <- rep("black", dim(limma.countsTMMvoom.genes)[1])
names(pchs) <- names(colors) <- rownames(limma.countsTMMvoom.genes)
pchs[gene.sel] <- "+"
colors[gene.sel] <- "red"
plot(
  limma.countsTMMvoom.genes$AveExpr,
  limma.countsTMMvoom.genes$logFC,
  col = colors,
  pch = pchs,
  xlab = "Average Expression",
  ylab = "log(FC)",
  main="MA plot of genes",
)
abline(h=0, col = "orange", lwd = 2)
```



Merged information about transcripts (`data/supplementary.RData` file) was then loaded into the environment. Previous steps in the means of normalization and transformations were repeated for this data set.

```
load(file = "data/supplementary.RData")
```

Table 4: First five rows of supplementary.RData

| | OE_1_R1 | OE_1_R2 | WT_R1 | WT_R2 |
|---|---|---|---|---|
| AT1G01010_ID1 | 83.673351 | 85.17584 | 142.173100 | 107.61582 |
| AT1G01020_ID8 | 8.750743 | 18.41270 | 21.360060 | 20.55019 |
| AT1G01020_ID9 | 37.910036 | 21.36570 | 22.675898 | 33.39722 |
| AT1G01020_ID4 | 9.650110 | 14.02244 | 8.363400 | 15.10687 |
| AT1G01020_ID5 | 11.817266 | 14.21590 | 8.250928 | 17.22066 |

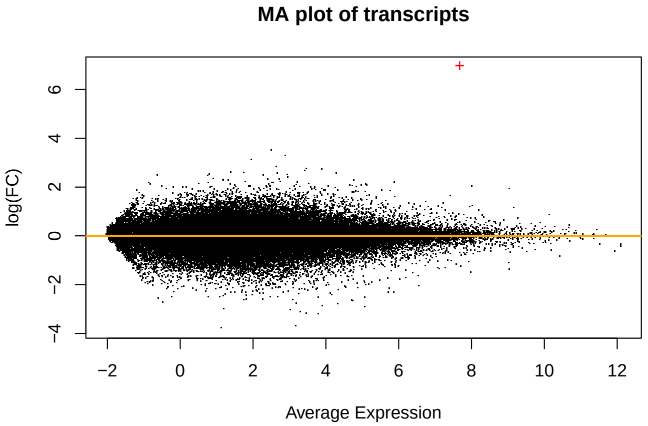
```
samples <- substr(colnames(trans.merged),0,2)
design <- data.frame(
  OEs=ifelse(samples=="OE",1,0),
  WTs=ifelse(samples=="WT",1,0)
)
rownames(design) <- colnames(trans.merged)
cm <- makeContrasts(OEvsWT=OEs-WTs, levels=design)
dge <- DGEList(counts=trans.merged)
dge <- calcNormFactors(dge)
v = voom(dge,design, plot = FALSE)
f.t <- lmFit(v,design)
cf <- contrasts.fit(f.t, cm)
fe <- eBayes(cf, proportion = 0.01)
limma.counts.TMMvoom.trans <- topTable(
  fe, number = Inf,
  adjust.method = "BH",
  sort.by = "none"
)
sign.trans <- limma.counts.TMMvoom.trans[
  which(limma.counts.TMMvoom.trans$adj.P.Val<sign.level),
]
```

Table 5: Significantly different transcripts.

| | logFC | AveExpr | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| AT3G01150_ID4 | 6.906772 | 7.675772 | 17.69561 | 0 | 0.00229 | -4.20909 |

As seen above, there is only one significantly differentiating transcript (AT3G01150_ID4). It is shown as a red + on a MA plot of transcripts below.

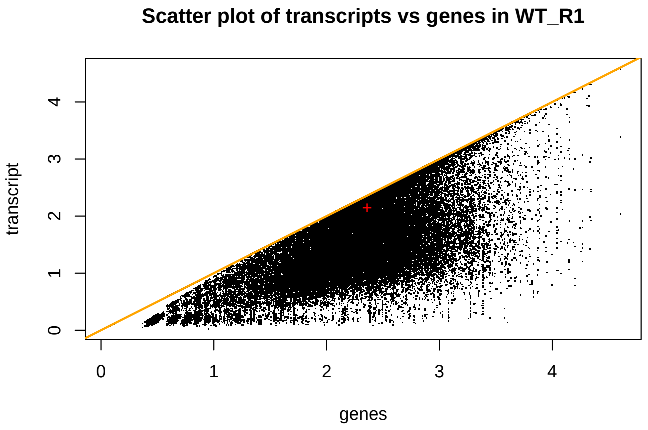
```
trans.sel <- "AT3G01150_ID4"
pchs <- rep(".", dim(limma.counts.TMMvoom.trans)[1])
cols <- rep("black", dim(limma.counts.TMMvoom.trans)[1])
names(pchs) <- names(cols) <- rownames(limma.counts.TMMvoom.trans)
pchs[trans.sel] <- "+"
cols[trans.sel] <- "red"
plot(
  limma.counts.TMMvoom.trans$AveExpr,
  limma.counts.TMMvoom.trans$logFC,
  xlab = "Average Expression",
  ylab="log(FC)",
  main="MA plot of transcripts",
  pch=pchs,
  col=cols
)
abline(h=0,col="orange", lwd=2)
```



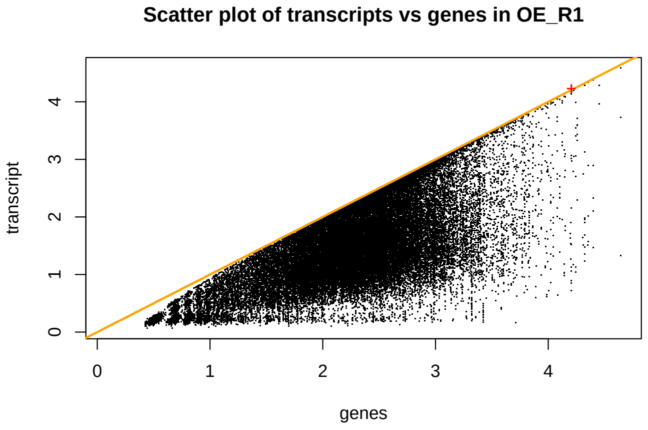
Scatter plots of transcripts vs genes were prepared for both WT_R1 and OE_R1 samples. Significantly differentiating transcripts were depicted with red +.

```
plot(
  log10(genes.merged[mapping[,"genes"],3]+0.1),
  log10(trans.merged[,3]+0.1),
  xlab="genes",
  ylab="transcript",
  main="Scatter plot of transcripts vs genes in WT_R1",
  pch=pchs,
  col=cols
)
abline(a=0,b=1, col="orange", lwd=2)
```

```
points(
  log10(genes.merged[mapping[trans.sel,"genes"],3]+0.1),
  log10(trans.merged[trans.sel,3]+0.1),
  pch=pchs[trans.sel],
  col=cols[trans.sel]
)
```



```
plot(
  log10(genes.merged[mapping[,"genes"],1]+0.1),
  log10(trans.merged[,1]+0.1),
  xlab="genes",
  ylab="transcript",
  main="Scatter plot of transcripts vs genes in OE_R1",
  pch=pchs, col=cols
)
abline(a=0,b=1, col="orange", lwd=2)
points(
  log10(genes.merged[mapping[trans.sel,"genes"],1]+0.1),
  log10(trans.merged[trans.sel,1]+0.1),
  pch=pchs[trans.sel],
  col=cols[trans.sel]
)
```



### edgeR

The next step was to construct a `DGEList` object and run a series of statistical tests using `edgeR`. Notice that this procedure is performed twice: for genes and transcripts. Both quasi-likelihood F-test and classic likelihood ratio test were used to determine *p*-values.

```
group <- factor(c(2,2,1,1))
y <- DGEList(counts=genes.merged ,group=group)
keep <- filterByExpr(y)
y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y)
design <- model.matrix(~group)
y <- estimateDisp(y,design)
```

Firstly, the quasi-likelihood F-test was performed:

```
#To perform quasi-likelihood F-tests:
fit <- glmQLFit(y,design)
qlf <- glmQLFTest(fit,coef=2)
genes.QLF.tt<- topTags(qlf, n=Inf)
sum(genes.QLF.tt$table$FDR<sign.level)
```

```
## [1] 0
```

Secondly, the classic likelihood ratio test:

```
#To perform likelihood ratio tests:
fit <- glmFit(y,design)
lrt <- glmLRT(fit,coef=2)
genes.LR.tt<- topTags(lrt, n=Inf)
sum(genes.LR.tt$table$FDR<sign.level)
```

```
## [1] 5
```

Then, the previous steps were repeated for transcripts.

```
group <- factor(c(2,2,1,1))
y <- DGEList(counts=trans.merged ,group=group)
keep <- filterByExpr(y)
y <- y[keep,,keep.lib.sizes=FALSE]
y <- calcNormFactors(y)
design <- model.matrix(~group)
y <- estimateDisp(y,design)
```

Quasi-likelihood F-tests:

```
fit <- glmQLFit(y,design)
qlf <- glmQLFTest(fit,coef=2)
trans.QLF.tt<- topTags(qlf, n=Inf)
sum(trans.QLF.tt$table$FDR<sign.level)
```

```
## [1] 0
```

Classic likelihood ratio test:

```
fit <- glmFit(y,design)
lrt <- glmLRT(fit,coef=2)
trans.LR.tt<- topTags(lrt, n=Inf)
sum(trans.LR.tt$table$FDR<sign.level)
```

```
## [1] 9
```

From the results above it can be deduced that there are no genes or transcripts that pass a quasi-likelihood test, there are however 5 genes and 9 transcripts that pass classic likelihood test.

### DESeq2

In `DESeq2` Wald test is used to determine *p*-values.

```
condition <- factor(c("OE", "OE", "WT", "WT"))
coldata.genes <- data.frame(row.names = colnames(genes.merged), condition)
dds.genes <- DESeqDataSetFromMatrix(
  countData = round(genes.merged),
  colData = coldata.genes,
  design = ~condition
)
dds.genes <- DESeq(dds.genes)
res.genes <- results(dds.genes)
```

Table 6: Significantly differentiating genes (DESeq2).

| | baseMean | log2FoldChange | lfcSE | stat | pvalue | padj |
|---|---:|---:|---:|---:|---:|---:|
| AT1G43800 | 156.73939 | 3.468741 | 0.6349943 | 5.462633 | 0.0e+00 | 0.0005607 |
| AT3G01150 | 8962.58903 | -6.121532 | 0.4033756 | -15.175762 | 0.0e+00 | 0.0000000 |
| AT3G12500 | 205.87802 | 2.195659 | 0.4515427 | 4.862573 | 1.2e-06 | 0.0092329 |
| AT5G35935 | 41.38681 | 3.123976 | 0.6662341 | 4.689006 | 2.7e-06 | 0.0164069 |

```
coldata.trans <- data.frame(row.names = colnames(trans.merged), condition)
dds.trans <- DESeqDataSetFromMatrix(
  countData = round(trans.merged),
  colData = coldata.trans,
  design = ~condition
)
dds.trans <- DESeq(dds.trans)
res.trans <- results(dds.trans)
```

Table 7: Significantly differentiating genes (DESeq2).

| | baseMean | log2FoldChange | lfcSE | stat | pvalue | padj |
|---|---:|---:|---:|---:|---:|---:|
| AT1G43800_ID1 | 158.61563 | 3.485717 | 0.7376612 | 4.725364 | 2.30e-06 | 0.0109561 |
| AT2G36530_ID120 | 129.87219 | -2.793770 | 0.6010623 | -4.648055 | 3.40e-06 | 0.0109561 |
| AT3G01150_ID4 | 8822.40283 | -6.912164 | 0.5415206 | -12.764359 | 0.00e+00 | 0.0000000 |
| AT3G10970_ID10 | 105.73625 | 2.772445 | 0.6013669 | 4.610239 | 4.00e-06 | 0.0109561 |
| AT3G51370_ID7 | 228.52462 | -2.630701 | 0.6175238 | -4.260080 | 2.04e-05 | 0.0342560 |
| AT3G52220_ID3 | 121.06964 | -2.707565 | 0.5906088 | -4.584362 | 4.60e-06 | 0.0110261 |
| AT4G01800_ID1 | 307.58196 | 2.680729 | 0.5700741 | 4.702423 | 2.60e-06 | 0.0109561 |
| AT4G14880_ID12 | 188.11808 | 3.091633 | 0.6534722 | 4.731086 | 2.20e-06 | 0.0109561 |
| AT4G14880_ID19 | 406.87575 | 2.868622 | 0.5782748 | 4.960655 | 7.00e-07 | 0.0076551 |
| AT4G19410_ID16 | 158.65729 | 3.579034 | 0.7726093 | 4.632399 | 3.60e-06 | 0.0109561 |
| AT4G24440_ID7 | 49.96931 | -3.101744 | 0.6994112 | -4.434794 | 9.20e-06 | 0.0200836 |
| AT5G46210_ID7 | 103.39655 | 2.929204 | 0.6720951 | 4.358318 | 1.31e-05 | 0.0259654 |
| AT5G52650_ID2 | 298.24973 | 2.569473 | 0.5968891 | 4.304774 | 1.67e-05 | 0.0303555 |

As seen above, when using `DESeq2` the results are 4 significantly differentiating genes and 13 transcripts. In the tables above $log_2(FC)$ is a indicator of both down and up regulation.