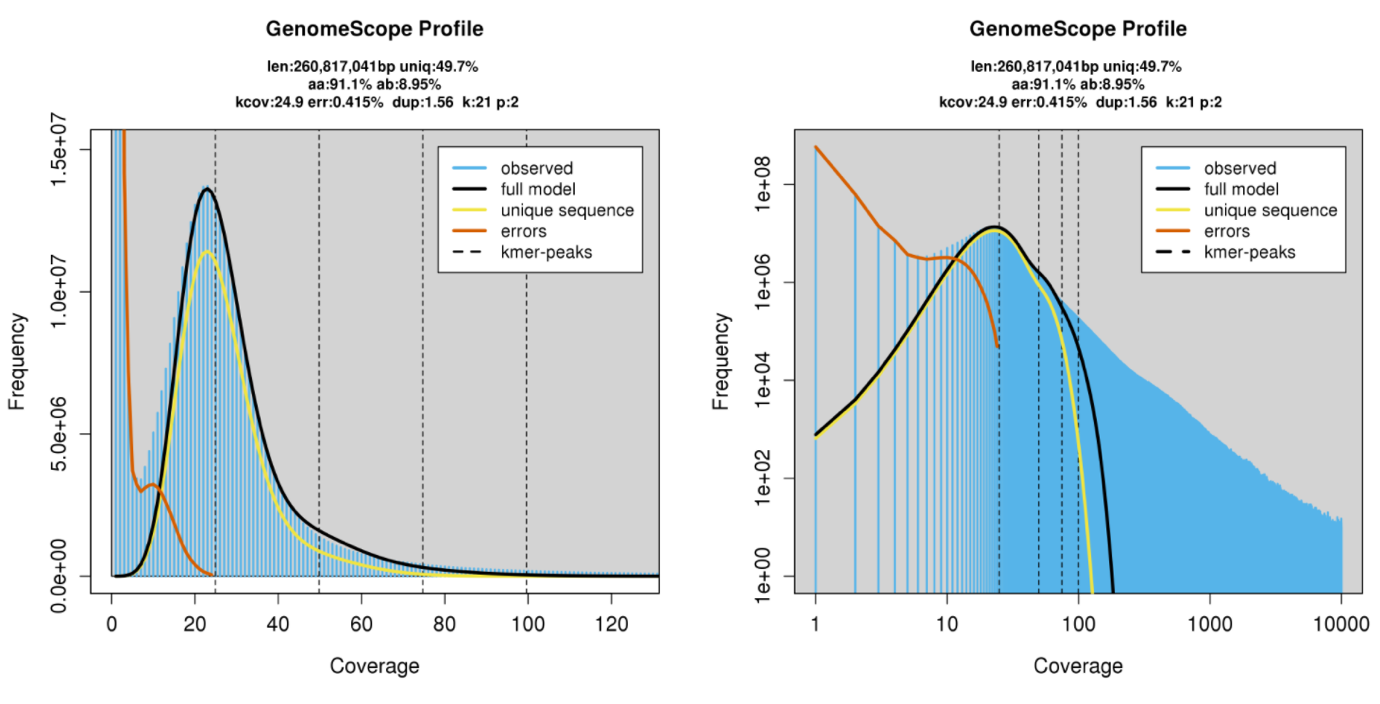
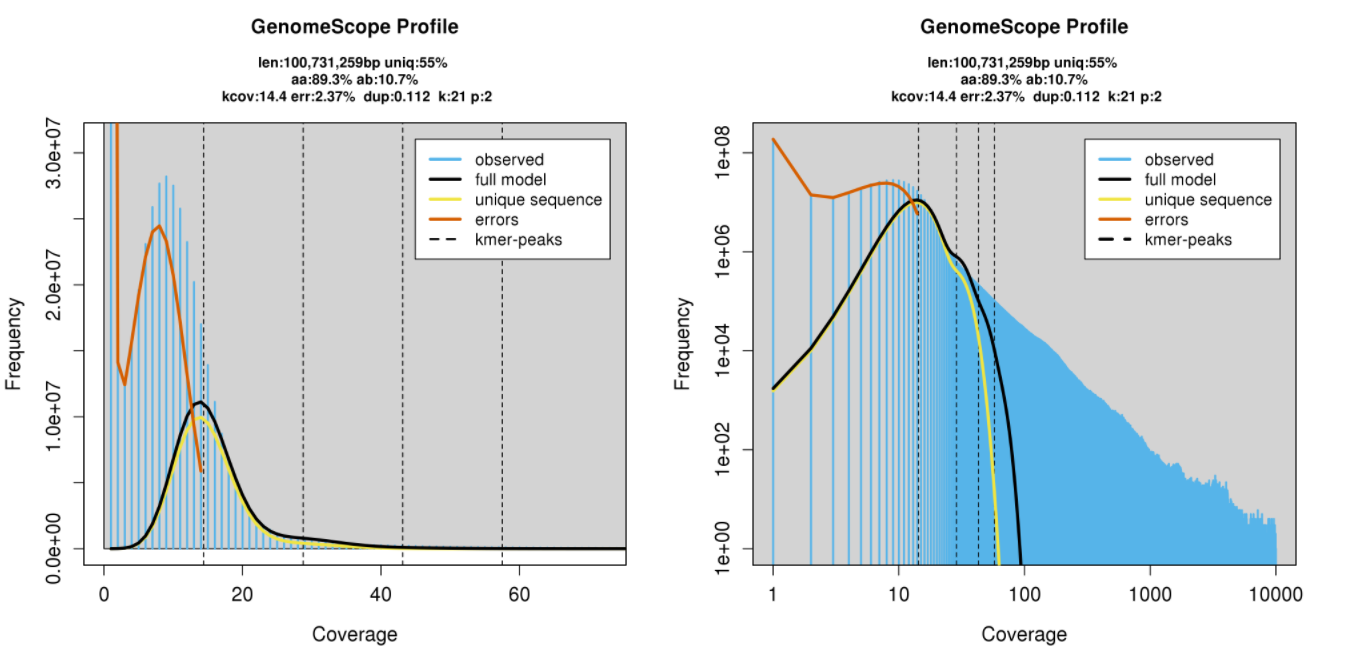
**Supporting Information**

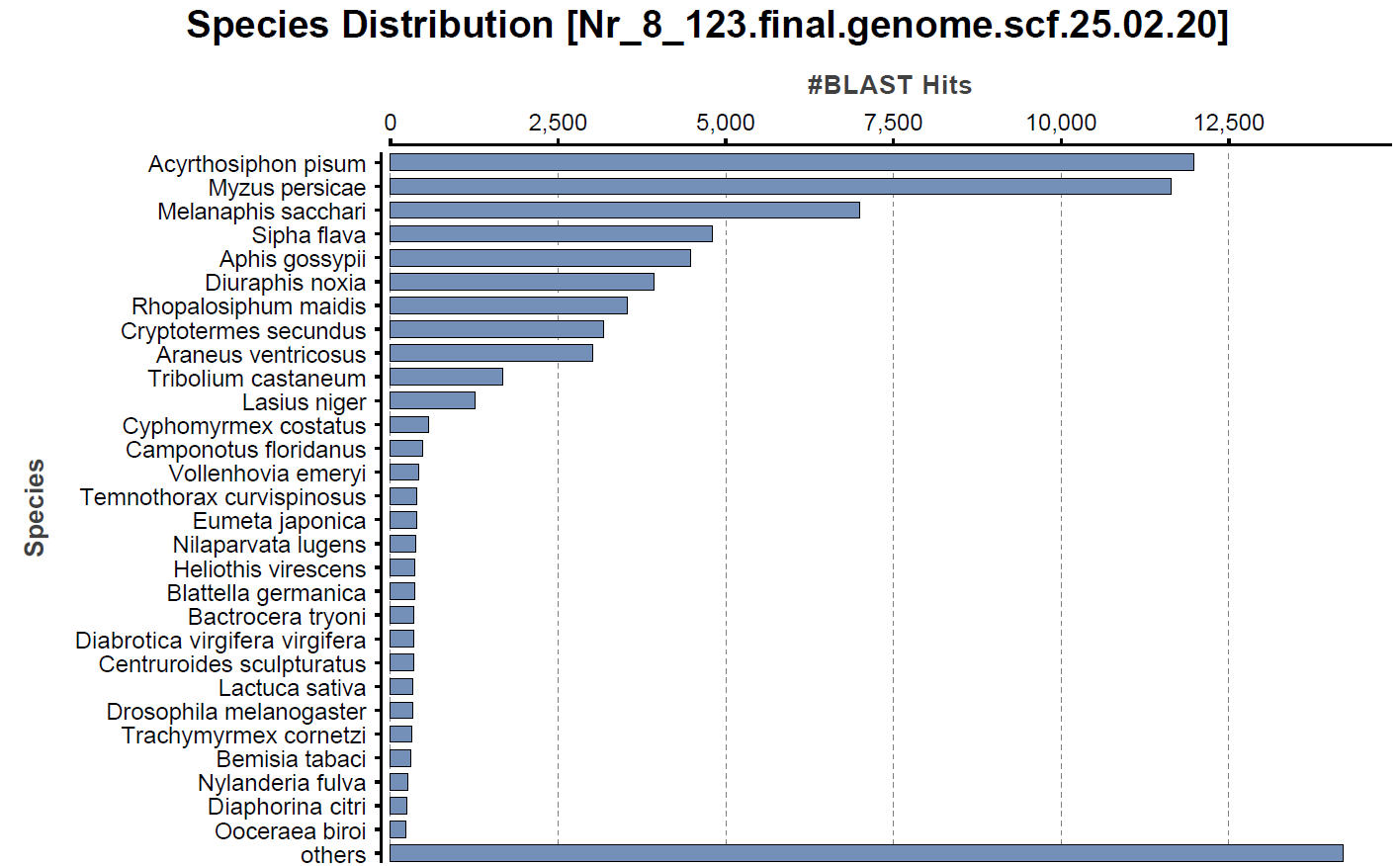


**A)**

**B)**



**S1.** **k-mer graphs for the *N. ribisnigri* genomes**. GenomeScope v2.0 profile plots of A: a transformed linear plot of k=mer frequency and B: a transformed log plot of k-mer coverage, at a k-mer length of 21 and a maximum k-mer coverage of 10000. A) Wild-type *N. ribisnigri* (Nr8\_123) k-mer plots (Illumina data only). B) Resistant *N. ribisnigri* k-mer plots (Illumina data only).

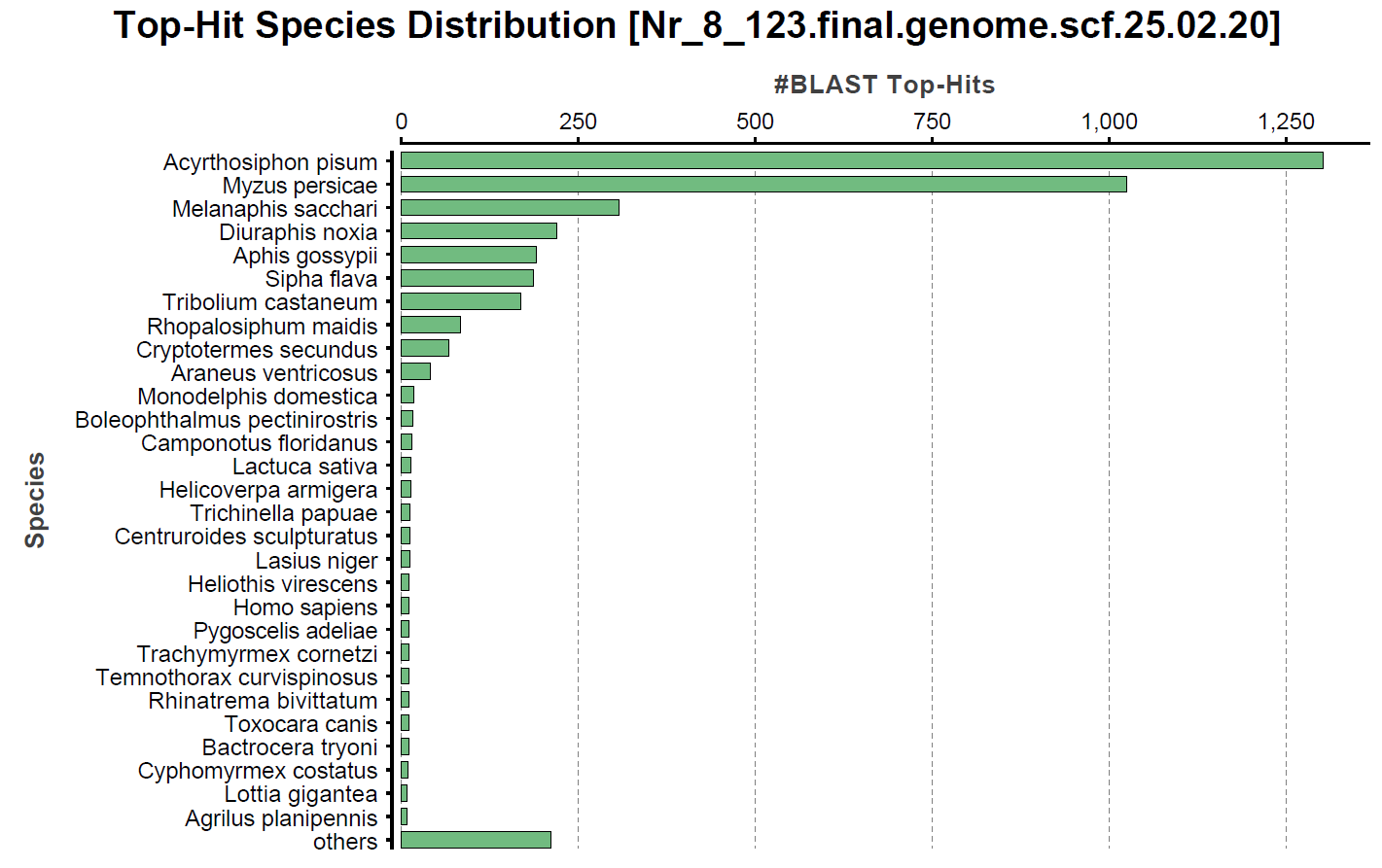


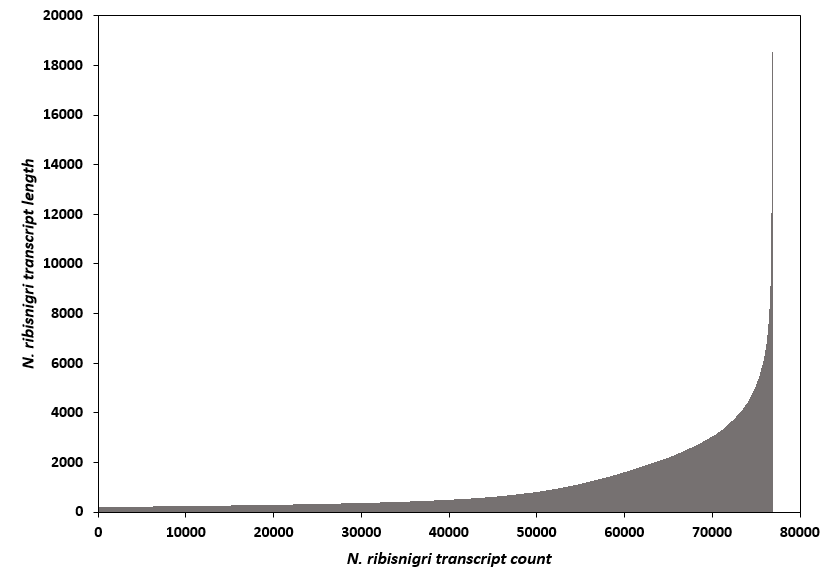
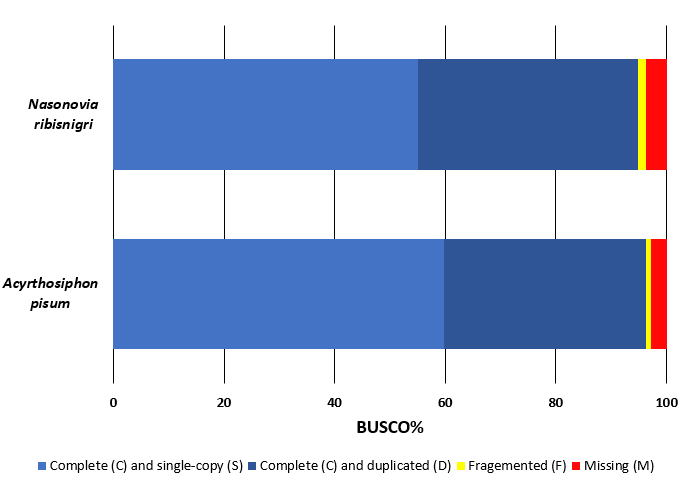
**B)**

**A)**

**A)**

**A)**

**S2. A)** **Species distribution for the wild-type genome (Nr\_8)** highlighting the number of shared genes with other organisms from BLAST using OmicsBox (v. 3.1.8).   
**B) Top-hit species distribution for the wild-type genome (Nr\_8)** highlighting the number of shared top-hit genes with other organisms from a BLAST search using OmicsBox (v. 3.1.8).



**A)**

**A)**

**B)**

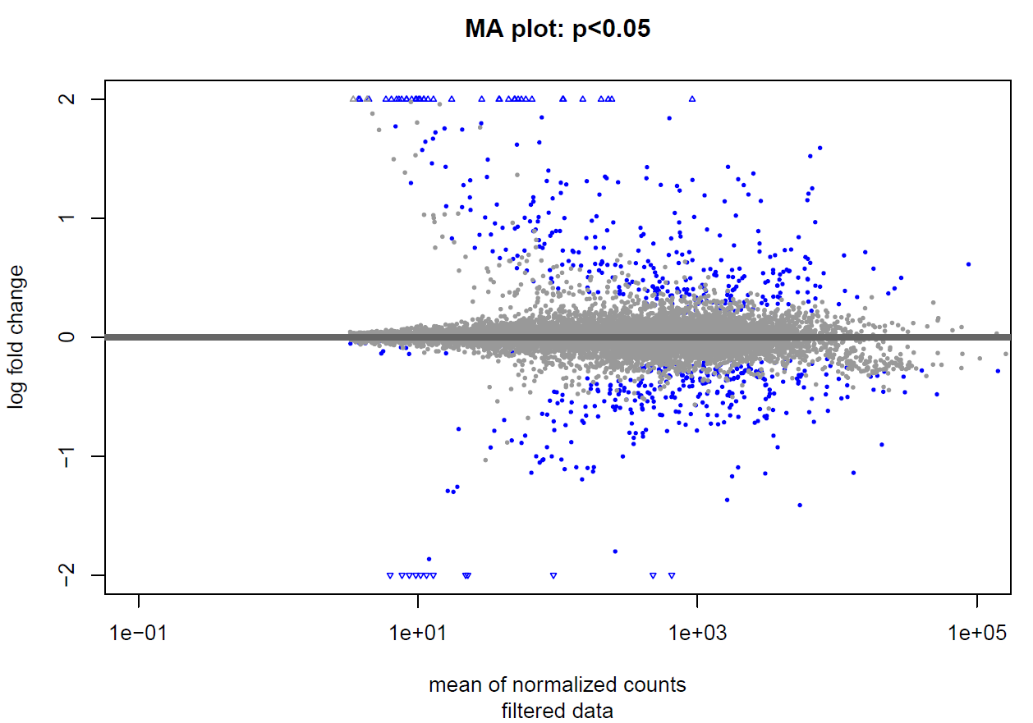
Figure **Error! No text of specified style in document.**.: A) Nasonovia ribisnigri transcript length distribution of 76,782 contigs of de novo assembled transcriptome. Individual contigs are ordered on X-axis based on increasing size. B) BUSCO analysis score of the N. ribisnigri transcriptome and the pea aphid (Acyrthosiphon pisum) transcriptome (obtained from BIPAA genomic resources) using an Arthropod gene set (n= 1562). C: Complete, S: Single-copy, D: Duplicate, F: Fragmented, M: Missing.**B)**

**S3. A)** ***Nasonovia ribisnigri* transcript length distribution** of 76,782 contigs of de novo assembled transcriptome. Individual contigs are ordered on X-axis based on increasing size. **B) BUSCO analysis score of the *N. ribisnigri* transcriptome** and the pea aphid (*A. pisum*) transcriptome (obtained from BIPAA genomic resources) using an Arthropod gene set (n= 1562). C: Complete, S: Single-copy, D: Duplicate, F: Fragmented, M: Missing.

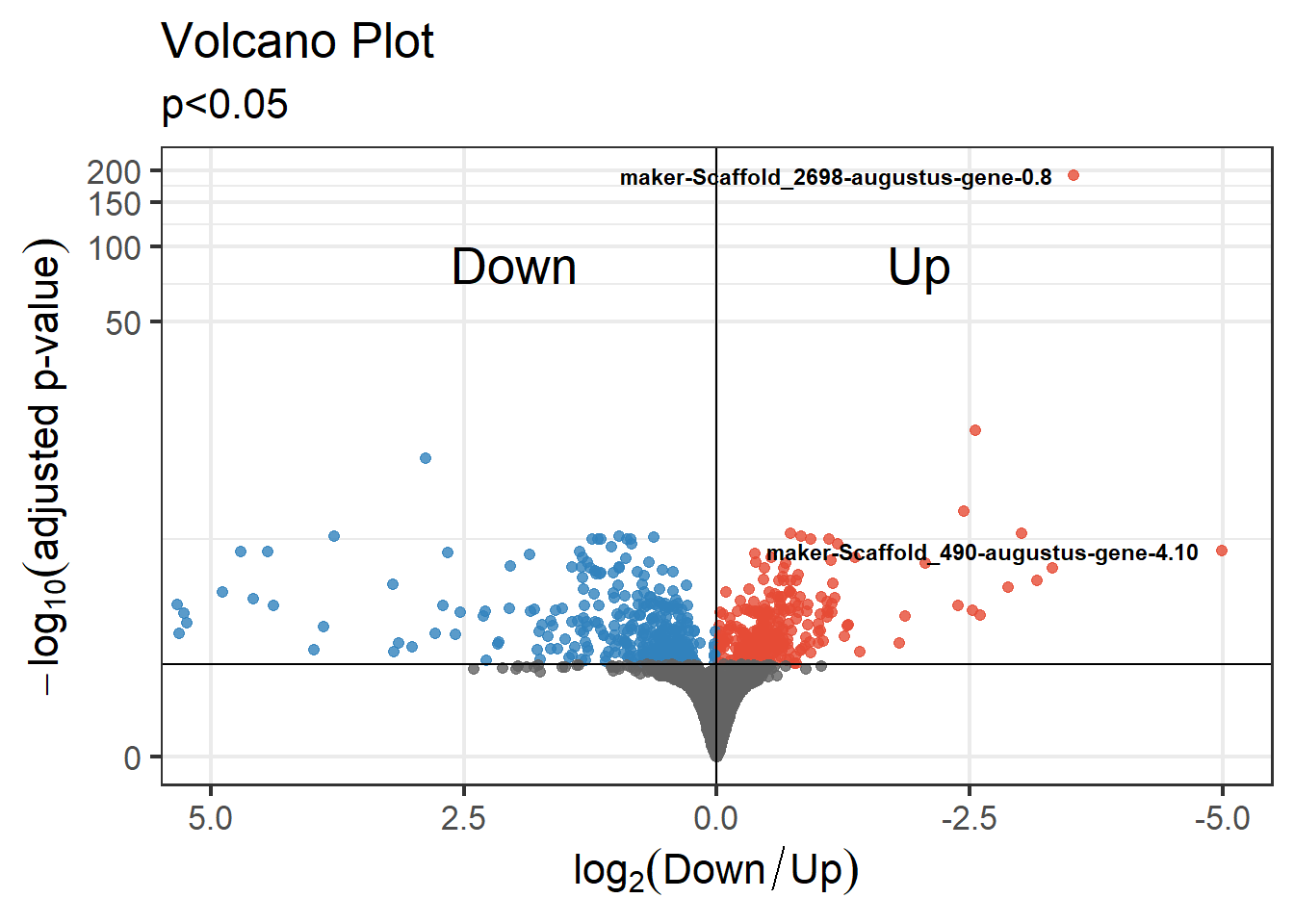
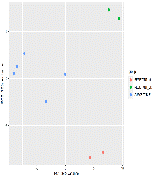
**S4.** **HISAT2 results** of number of reads which aligned to the Nr\_8 WT genome annotation using feature counts and mapping statistics.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Nasonovia ribisnigri* culture** | **Biotype** | **Host plant** | **No. of HiSeq reads** | **Assigned** | **Unmapped** | **Mapping quality** | **Chimera** |
| Nr4 | Nr:0, Pyrethroid R | Pinokkio (S) | 37,726,652 | 30225819 | 788040 | 33682 | 62599 |
| Nr8 | Nr:0, Insecticide R | Pinokkio (S) | 11,418,030 | 9119987 | 257288 | 9339 | 21991 |
| Nr29 | Nr:0, Insecticide R | Pinokkio (S) | 11,779,986 | 8174203 | 1231914 | 50176 | 68931 |
| 4850a | Nr: 0 | Pinokkio (S) | 17,018,669 | 1253797 | 317543 | 21170 | 36128 |
| WT Kent | Nr: 0 | Pinokkio (S) | 37,227,165 | 28894399 | 2041638 | 49777 | 95444 |
| Kent CL | Nr: 1 | Eluarde (R) | 18,270,893 | 13918495 | 538177 | 28144 | 48192 |
| Kent CL | Nr: 1 | Pinokkio (S) | 19,054,666 | 14008966 | 516558 | 24536 | 45233 |
| UK631 | Nr: 1 | Eluarde (R) | 18,356,475 | 9618800 | 219916 | 13510 | 27106 |
| Uk631 | Nr: 1 | Pinokkio (S) | 12,634,191 | 14452372 | 479047 | 22803 | 40625 |

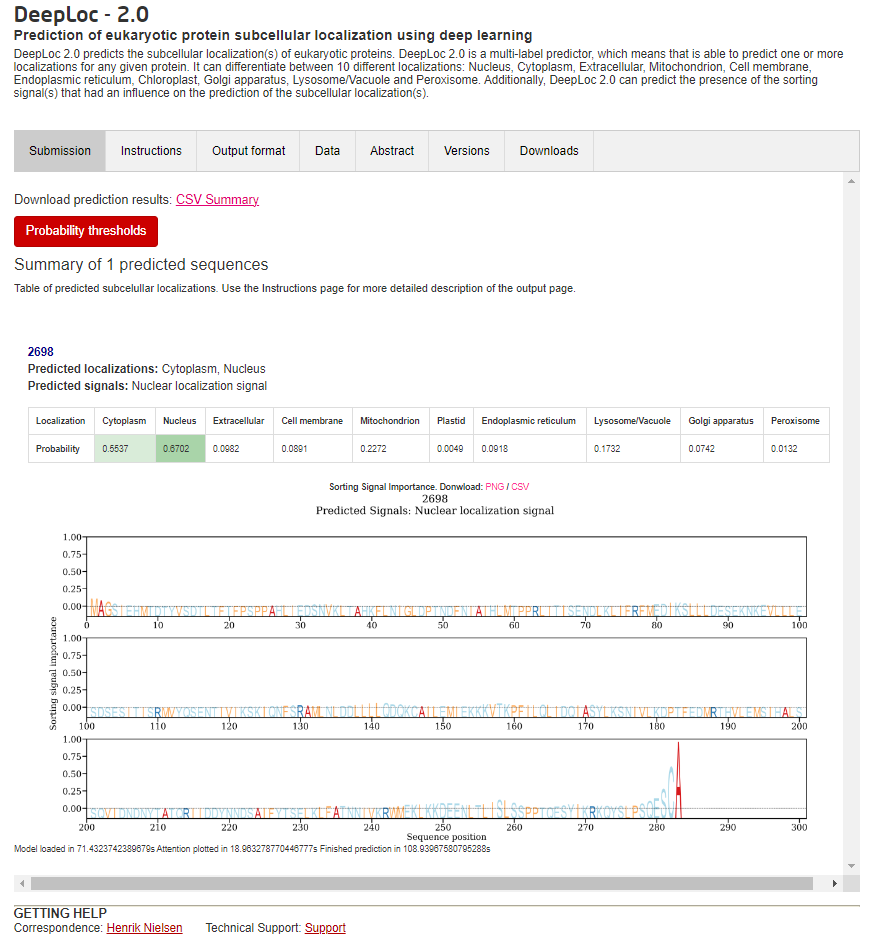
Nr:0 = *N. ribisnigri* unable to feed to lettuce containing the Nr-gene (susceptible); Nr:1 *N. ribisnigri* able to break the host plant resistance and feed on lettuce containing the Nr-gene (resistant).

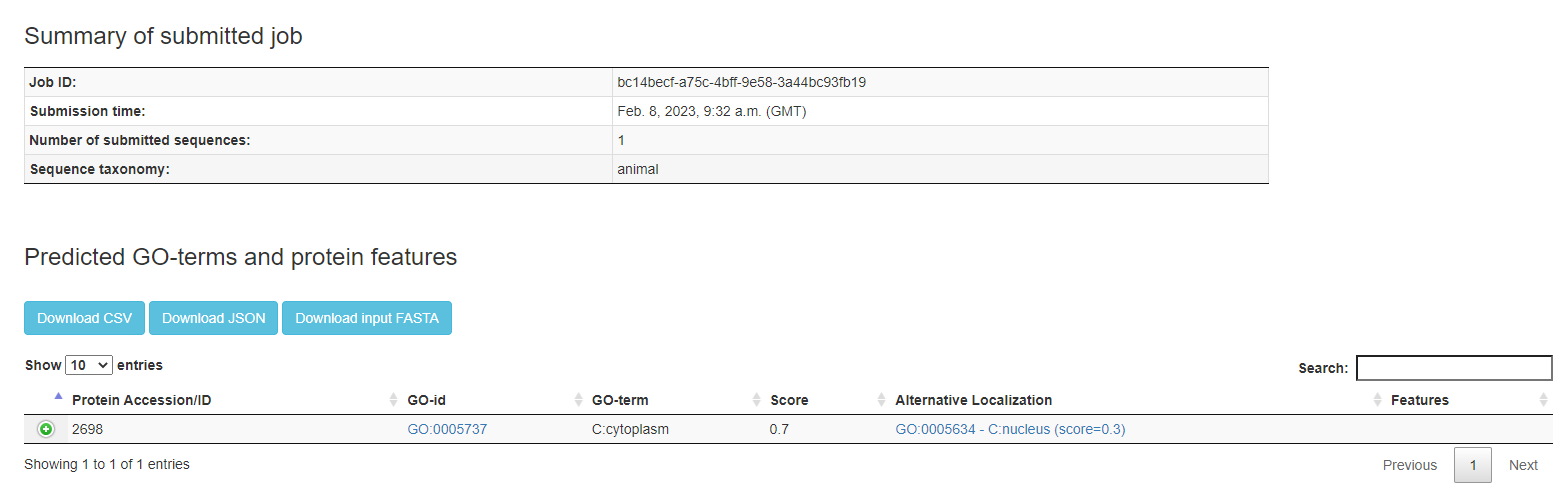


**S5.** **Gene expression change in *N. ribisnigri* biotypes** feeding on resistant (Nr-gene) and susceptible (Nr-gene absent) lettuce host plants. The shrunken log2 fold change (removing the noise associated with log2 fold changes from low count genes) of each gene between *N. ribisnigri* able to feed on resistant lettuce (Nr:1) and *N. ribisnigri* unable to feed on resistant lettuce (Nr:0). Fold changes were considered significant if P values were <0.05. Differentially expressed (DE) genes are coloured as blue circles and DE genes which fall out of the window are indicated by open triangles. The remaining grey circles are genes which are not significantly expressed between the resistance-breaking and susceptible biotype.

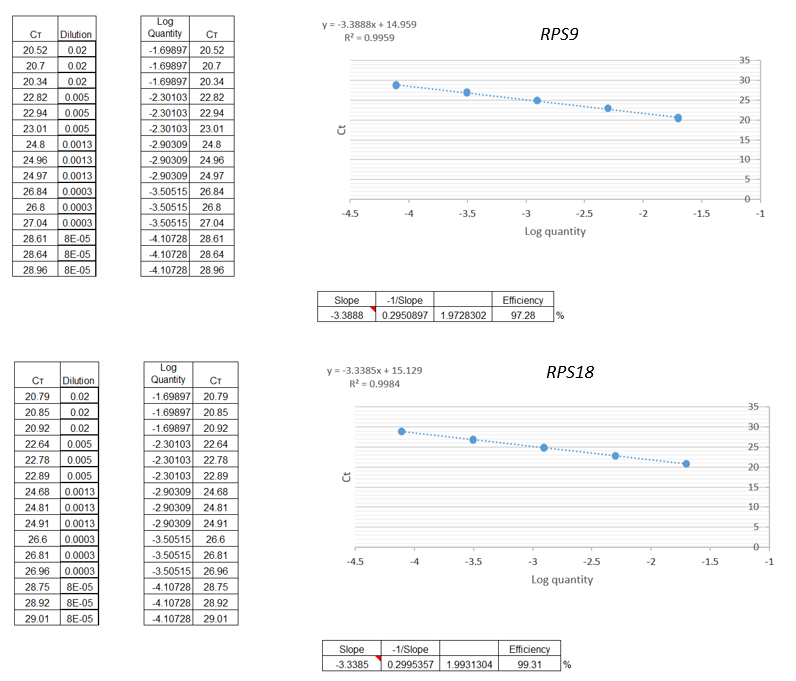


**S6.** **Volcano plot** of the 18,872 genes in both resistance-breaking (Nr:1) and susceptible (Nr:0) *N. ribisnigri* biotypes. 689 differentially expressed (DE) genes were identified with a P value of <0.05, highlighted in either blue or red. The volcano plot used the susceptible (Nr:0) *N. ribisnigri* biotype as a foundation for the plot and therefore DE genes are shown to be down regulated in the Nr:0 biotype and up regulated in the resistance-breaking (Nr:1) biotype.

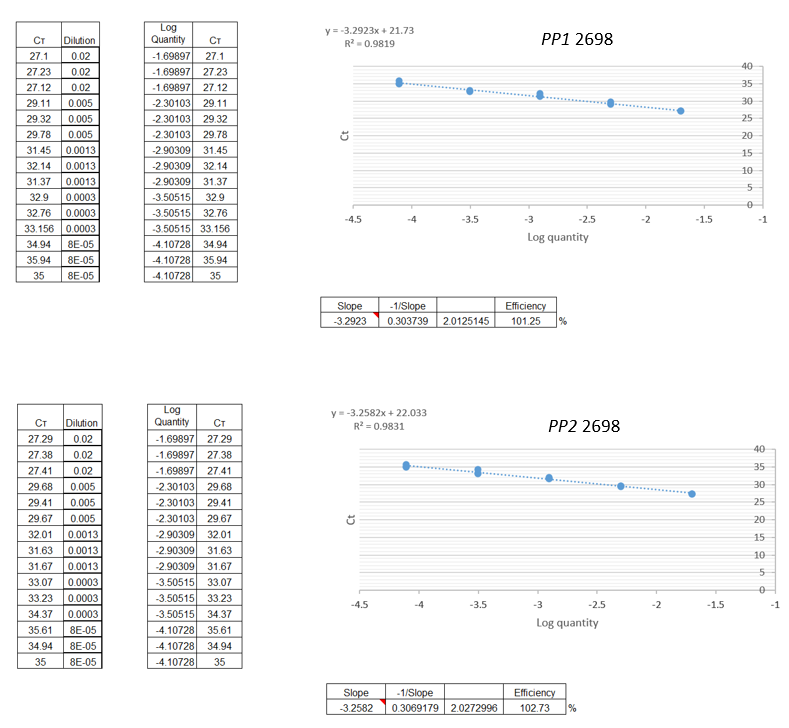




**S7. Protein prediction outputs** Both DeepLoc 2.0 (top) and BUSCA (below) predict localisation in both the cytoplasm and nucleus.



**S8. Primer efficiencies of *RPS9* and *RPS18***



**S9. Primer efficiencies of *PP1* and *PP2***

**S10. Conditions used for qRT-PCR experiment** for *N. ribisnigri*. Nr:0 = susceptible biotype (unable to feed on resistant lettuce containing the Nr-gene.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Treatment** | ***Nasonovia ribisnigri* culture** | **Biotype** | **Host plant variety** | **Biological replicate** |
| 1 | WT Kent | Nr: 0 | Pinokkio (S) | 1 |
| 1 | WT Kent | Nr: 0 | Pinokkio (S) | 2 |
| 1 | WT Kent | Nr: 0 | Pinokkio (S) | 3 |
| 2 | Kent CL | Nr: 1 | Eluarde (R) | 1 |
| 2 | Kent CL | Nr: 1 | Eluarde (R) | 2 |
| 2 | Kent CL | Nr: 1 | Eluarde (R) | 3 |
| 3 | UK631 | Nr: 1 | Eluarde (R) | 1 |
| 3 | UK631 | Nr: 1 | Eluarde (R) | 2 |
| 3 | UK631 | Nr: 1 | Eluarde (R) | 3 |
| 4 | Uk631 | Nr: 1 | Pinokkio (S) | 1 |
| 4 | Uk631 | Nr: 1 | Pinokkio (S) | 2 |
| 4 | Uk631 | Nr: 1 | Pinokkio (S) | 3 |

Nr:1 = resistant biotype (able to feed and reproduce on resistant lettuce containing the Nr-gene). UK631 were cultured on both susceptible (Pinokkio) and resistant (Eluarde) lettuce lines. Susceptible *N. ribisnigri* cultures were only cultured on susceptible (Pinokkio) lettuce lines.

**S11: Accession numbers for all genomes** are available on the NCBI (BioProject ID: PRJNA857679).

|  |  |  |  |
| --- | --- | --- | --- |
| **Accession Number** | **Genome** | **Organism** | **Tax ID** |
| SAMN29667034 | Nr\_8\_123 | Nasonovia ribisnigri | 269403 |
| SAMN29667035 | WT\_Kent\_123 | Nasonovia ribisnigri | 269403 |
| SAMN29667036 | UK631\_1 | Nasonovia ribisnigri | 269403 |
| SAMN29667037 | UK631\_2 | Nasonovia ribisnigri | 269403 |
| SAMN29667038 | UK631\_12 | Nasonovia ribisnigri | 269403 |
| SAMN29667039 | Kent\_CL\_1 | Nasonovia ribisnigri | 269403 |
| SAMN29667040 | Kent\_CL\_2 | Nasonovia ribisnigri | 269403 |
| SAMN29667041 | Kent\_CL\_12 | Nasonovia ribisnigri | 269403 |
| SAMN29667042 | Ely\_1 | Nasonovia ribisnigri | 269403 |
| SAMN29667043 | Ely\_2 | Nasonovia ribisnigri | 269403 |
| SAMN29667044 | Ely\_12 | Nasonovia ribisnigri | 269403 |

**S12. Accession numbers** **for all sequence reads** which are available on the NCBI (BioProject ID: PRJNA857679).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Accession Number** | **Sample Name** | **Organism** | **Tax ID** | **Breed** |
| SAMN29632255 | Ely\_1\_F1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632256 | Ely\_1\_R1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632257 | Ely\_2\_F2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632258 | Ely\_2\_R2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632259 | Kent\_CL\_1\_F1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632260 | Kent\_CL\_1\_R1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632261 | Kent\_CL\_2\_F2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632262 | Kent\_CL\_2\_R2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632263 | Nr\_8\_123\_F1 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632264 | Nr\_8\_123\_R1 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632265 | UK\_631\_1\_F1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632266 | UK\_631\_1\_R1 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632267 | UK\_631\_2\_F2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632268 | UK\_631\_2\_R2 | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |
| SAMN29632269 | wt\_Kent\_1\_F1 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632270 | wt\_Kent\_1\_R1 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632271 | wt\_Kent\_2\_F2 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632272 | wt\_Kent\_2\_R2 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632273 | wt\_Kent\_3\_F3 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632274 | wt\_Kent\_3\_R3 | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632275 | Nr\_8\_nanopore | Nasonovia ribisnigri | 269403 | Nr-0 (susceptible) |
| SAMN29632276 | Kent\_CL\_nanpore | Nasonovia ribisnigri | 269403 | Nr-1 (resistant) |

**S13.** **Primer sequences** and amplicon characteristics of resistance genes and house-keeping genes for qRT-PCR validation experiment for *N. ribisnigri*.

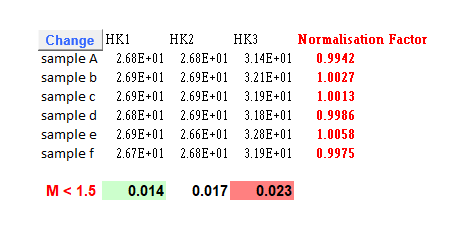
|  |  |  |  |
| --- | --- | --- | --- |
| **Gene** | **Primer sequence** | **Amplicon length (bp)** | **Product temp (°C)** |
| ***2698\_1*** | F: GTTAGTCGGATCGAGTCCAATG R: CGTGAGCGATACTTTGACATTTAC | 115 | 62 |
| ***2698\_2*** | F:  ACGTTGAGTGGCAGTATAGTTATC R: AAGATATGAGAACCCACGTGTTAG | 89 | 62 |
| ***RPS9*** | F: CTGTTGACCCTCGAAGAGAAG R: GCCCTCGTCTAATACTCCAATAC | 93 | 62 |
| ***RPS18*** | F: CGTATCCTCAGCACCAACAT R: CGGTACATTCTCCAGCTCTTT | 142 | 62 |
| ***RPL13*** | F: TCAAATACGATGCAAACCTTCAC R: ACGCACTCCTCATTCCTTAAC | 102 | 62 |

**S14. qRT-PCR validation methodology**

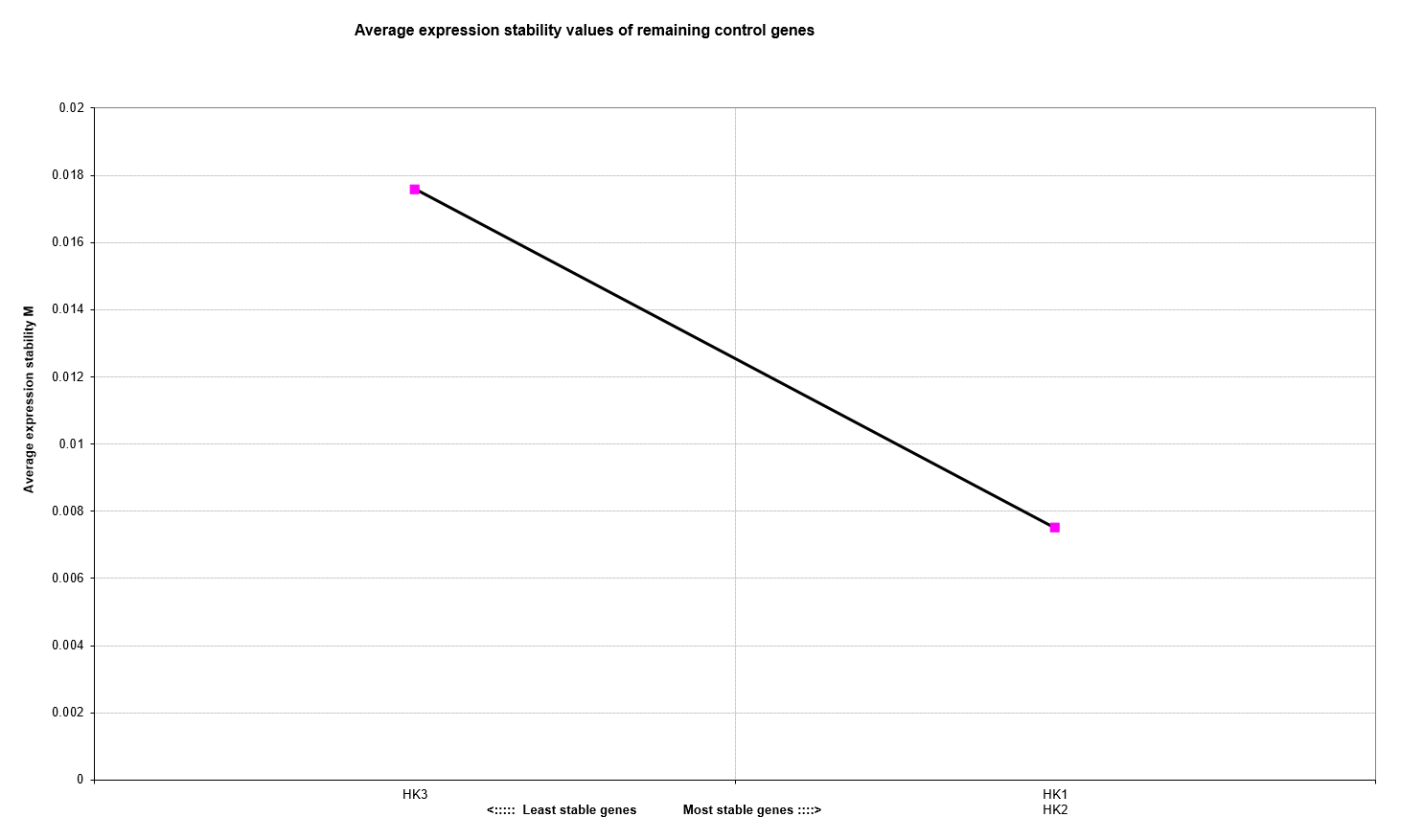
For RNA-seq data validation, aphid samples were collected as described previously. Four *N.ribisnigri* cultures were used for the qRT-PCR experiment; a WT strain feeding on susceptible lettuce (WT Kent), two resistant strains feeding on resistant lettuce (containing the Nr-locus) (Kent CL and UK631) and finally, a resistant strain feeding on susceptible lettuce (UK631). Total RNA was extracted and treated with DNase I using a RNeasy Micro Kit (50) (Qiagen), following the manufacturer’s protocol. All RNA samples were checked for quality and quantity using a nanodrop ND-1000 spectrophotometer. First-strand cDNA was prepared with 200 ng RNA (DNA free) for all samples using A High-Capacity cDNA reverse Transcription Kit (Life Technologies) according to the manufacturer’s instructions using 200 ng of DNA-free RNA for all samples.

The qRT-PCR was conducted using the 7500 Fast Real-Time PCR system (Applied Biosystems) with PowerUp SYBR Green Master Mix (ThermoFisher Scientific) according to the manufacturer’s instructions using 200 ng of DNA-free RNA for all samples. qRT-PCR reactions were conducted with the following conditions: Uracil-DNA glycosylases (UDG) activation at 50°C for 2 min, followed by Taq DNA polymerase step of 95°C for 2 min, 40 cycles of denature at 95°C for 15 s, annealing of 55°C for 15 s and extension at 72°C for 1 min. Melt curve analysis was performed by heating the PCR samples from 55 to 95°C (1°C per cycle of 10 secs) with SYBR Green I signal intensity measurements.

Each reaction was performed in triplicate (**S10. Conditions used for qRT-PCR experiment**) using three technical replicates for each biological replicate and no-template controls were included in 96-well optical-grade PCR plates (ThermoFisher Scientific) sealed with optical sealing tape. qRT-PCR primers were designed to obtain 89-142 bp products for the resistance and reference genes using PrimerQuest (**S12. Primer sequences**). Each reaction was carried out with 2 µl of cDNA, 0.4 µl of each primer pair (100 ng/µl), 2.2µl RNase-free water and 5 µl of PowerUp SYBR Green Master Mix in a total volume of 10 µl. qRT-PCR reactions were conducted with the following conditions: Uracil-DNA glycosylases (UDG) activation at 50°C for 2 min, followed by Taq DNA polymerase step of 95°C for 2 min, 40 cycles of denature at 95°C for 15 s, annealing of 55°C for 15 s and extension at 72°C for 1 min. Melt curve analysis was performed by heating the PCR samples from 55 to 95°C (1°C per cycle of 10 secs) with SYBR Green I signal intensity measurements. Relative transcript levels were calculated by using the comparative Ct method (2-∆∆Ct) using *RPS9* and *RPS18* as reference genes. Statistical analysis was performed using t-test through the MeV package, version 4.9 available at www.tm4.org. A Bonferroni corrected *P*-value of >0.008 or less was considered significant.



**A)**



**B)**

**S15.** **Housekeeping (HKGs) stability (GeNorm)** A) Normalised expression levels for *RPS9* (HK1), *RPS18* (HK2) and *RPL13* (HK3). A M value of < 1.5 suggests a more stable gene expression or low variation, and an M-score > 1.5 indicating high variation and therefore not a suitable HKG. Most stable and least stable HKG’s are highlighted in green and red, respectively. B) Average stability of HK1, HK2 and HK3. Least stable on the left (HK3) and most stable on the right (HK1 and HK2).

**S16. Power analysis code**

**#Set working directory**

setwd("”)

**##Load counts table and define treatment groups.**  
data <- read.csv("combined\_feature\_counts\_table\_ALL2.tab", row.names=1, header=TRUE, stringsAsFactors=FALSE, sep='\t', colClasses=c("character",rep("numeric",9)))

head(data) ; dim(data)

**#Assigning susceptible and resistant \*N. ribisnigri\* biotype columns**  
SUS <-1:5 ; RES <-6:9   
data <-data[,c(SUS, RES)]  
colnames(data)<- c("SUS1","SUS2","SUS3","SUS4","SUS5","RES1","RES2","RES3","RES4")

rbtype <- c(rep("SUS",5),rep("RES",4))   
colData <- data.frame(rbtype, row.names=colnames(data))  
colData

**##Use DSS package to estimate dispersion and baseline expression.**  
library(DSS)  
library(edgeR)  
  
counts = as.matrix(data)  
counts<-counts[apply(counts,1,sum)>9,]  
design = data.frame(gender=rbtype)  
X = model.matrix(~gender, data=design)  
rownames(X)<-colnames(counts)  
seqData = newSeqCountSet(counts, as.data.frame(X))  
seqData = estNormFactors(seqData)  
seqData = estDispersion(seqData)  
dispersion = dispersion(seqData)  
fit.edgeR<-glmFit(counts, X, lib.size=normalizationFactor(seqData), dispersion=dispersion(seqData)) lrt.edgeR<-glmLRT(fit.edgeR, coef=2)

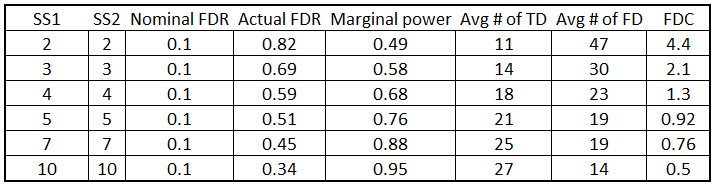
**##Simulate counts using PROPER package: 50,000 transcripts (‘ngenes’)**  
**##Simulate for 2, 3, 5, 7 and 10 replicates per condition (‘Nreps’), testDE with DESeq** **(‘DEmethod’), over 100 iterations (‘nsims’).**library(PROPER)  
library(DESeq2)

sz = 50000  
zz = sample(1:nrow(counts),sz)  
sim.opts = RNAseq.SimOptions.2grp (ngenes = sz, p.DE=0.01, lOD = log(dispersion(seqData))[zz], lBaselineExpr = apply(log(counts+0.5),1,mean)[zz], lfc = lrt.edgeR$table$logFC[zz], sim.seed = 11111)

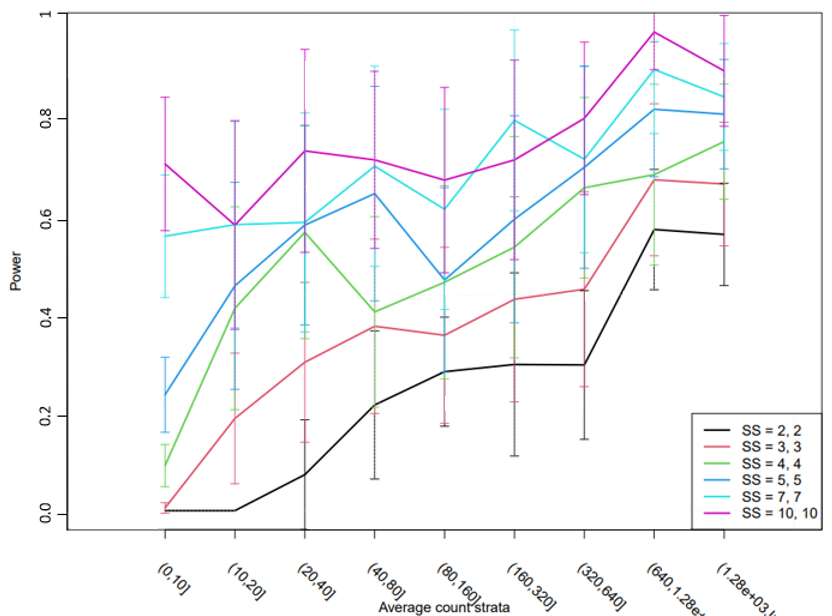
simres = runSims(Nreps = c(2, 3, 4, 5, 7, 10), sim.opts=sim.opts, DEmethod="DESeq2", nsims=100)

**##From simulated data, calculate power at FDR 0.05, for a 1.5 fold-change (equiv. 0.585 log2 fold-change) (‘delta’).**powers = comparePower(simres, alpha.type="fdr", alpha.nominal=0.1, stratify.by="expr", delta=0.585)

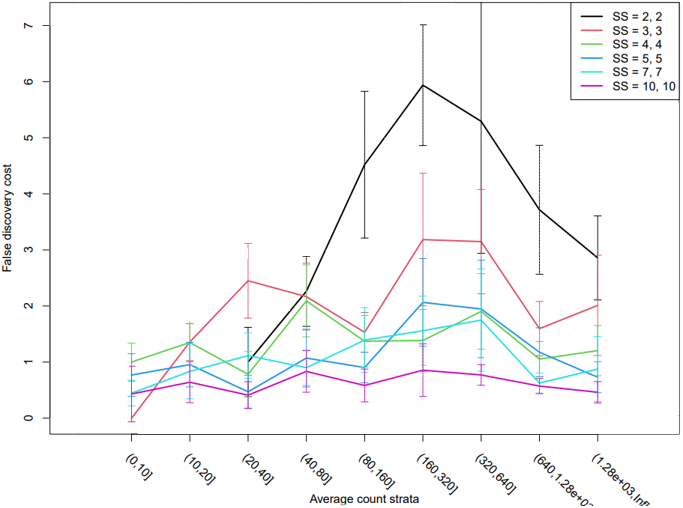
summaryPower(powers)



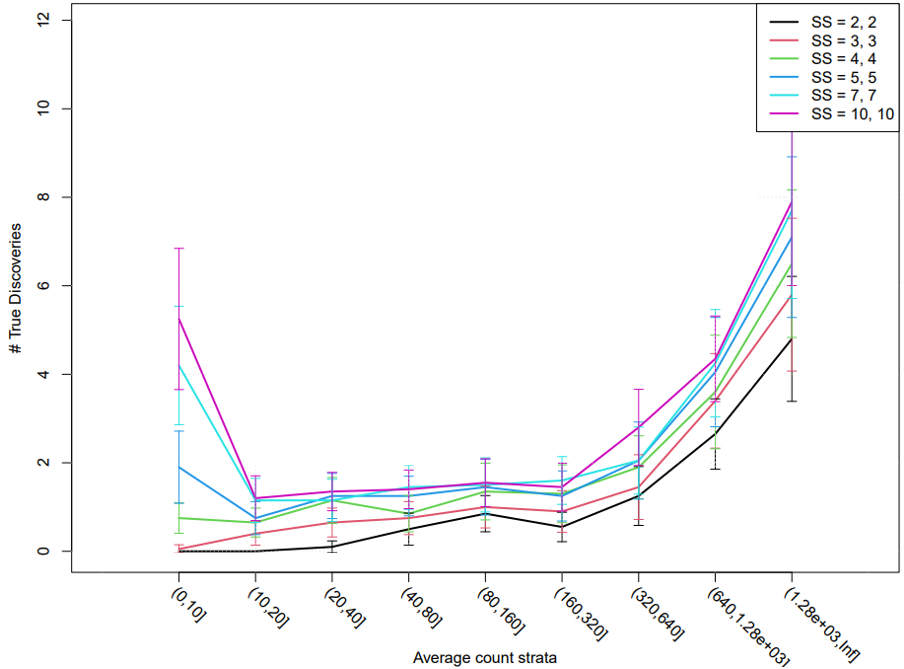
**#Plot result**  
plotPower(powers, main="Susceptible vs. Resistant", lty=1,1:ncol(powers$power))



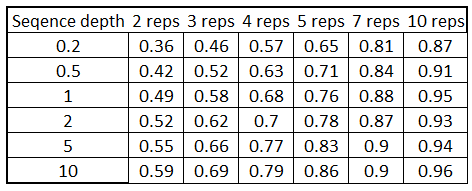
**#plotFDcosts for RNAseq data**  
plotFDcost(powers, main="Susceptible vs. Resistant", lty=1,1:ncol(powers$power))



**#plot true discoveries]**  
plotPowerTD(powers, main="Susceptible vs. Resistant", lty=1,1:ncol(powers$power))



**#power and sequencing depth**  
power.seqDepth(simres, powers)



**S17 DeSeq2 analysis code**

RNA-seq analysis used herein was adapted from the course material provided by Rothamsted Research bioinformatics team - Planning & analysing an RNA-seq experiment – 2018. Cheers guys.

Data available on request.

DE - differentially expressed.

Required libraries

**library**(DESeq2)

**library**(ggplot2)

**library**(pheatmap)

Working directory

setwd("//salt/aphid\_s2156/RRes/Rothamsted\_Research/Laboratory\_work/Sequences/Novogene\_assembly/RNA-seq\_analysis")

getwd()

## [1] "\\\\salt/aphid\_s2156/RRes/Rothamsted\_Research/Laboratory\_work/Sequences/Novogene\_assembly/RNA-seq\_analysis"

Import count matrix

data <- read.csv("combined\_feature\_counts\_table\_ALL2.tab", row.names=1, header=TRUE, stringsAsFactors=FALSE, sep='\t',

colClasses=c("character",rep("numeric",9)))

Header

head(data) ; dim(data)

## SU1 SU2 SU3 SU4 SU5 RE1 RE2 RE3 RE4

## augustus-Scaffold\_1-processed-gene-0.0 0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.0 0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.1 0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.7 0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.2 20 7 8 1 25 2 7 1 3

## augustus-Scaffold\_1-processed-gene-3.0 25 8 6 0 33 3 7 7 4

## [1] 38389 9

Assigning susceptible and resistant *N. ribisnigri* biotype columns

SUS <-1:5 ; RES <-6:9

data <-data[,c(SUS, RES)]

colnames(data)<- c("SUS1","SUS2","SUS3","SUS4","SUS5","RES1","RES2","RES3","RES4")

Checking to make sure it’s worked

head(data) ; dim(data)

## SUS1 SUS2 SUS3 SUS4 SUS5 RES1 RES2 RES3

## augustus-Scaffold\_1-processed-gene-0.0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.0 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.1 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.7 0 0 0 0 0 0 0 0

## augustus-Scaffold\_1-processed-gene-1.2 20 7 8 1 25 2 7 1

## augustus-Scaffold\_1-processed-gene-3.0 25 8 6 0 33 3 7 7

## RES4

## augustus-Scaffold\_1-processed-gene-0.0 0

## augustus-Scaffold\_1-processed-gene-1.0 0

## augustus-Scaffold\_1-processed-gene-1.1 0

## augustus-Scaffold\_1-processed-gene-1.7 0

## augustus-Scaffold\_1-processed-gene-1.2 3

## augustus-Scaffold\_1-processed-gene-3.0 4

## [1] 38389 9

Making a dataframe that describes the data

rbtype <- c(rep("SUS",5),rep("RES",4))

colData <- data.frame(rbtype, row.names=colnames(data))

Checking data

colData

## rbtype

## SUS1 SUS

## SUS2 SUS

## SUS3 SUS

## SUS4 SUS

## SUS5 SUS

## RES1 RES

## RES2 RES

## RES3 RES

## RES4 RES

Constructing a DeqSeq dataset and removing rows with 0 or 1 count in total

dds <- DESeqDataSetFromMatrix(countData=data, colData=colData, design=~rbtype)

dds <- dds[rowSums(counts(dds))>1,]

Checking dataset

dds

## class: DESeqDataSet

## dim: 18872 9

## metadata(1): version

## assays(1): counts

## rownames(18872): augustus-Scaffold\_1-processed-gene-1.2

## augustus-Scaffold\_1-processed-gene-3.0 ...

## maker-Scaffold\_998-augustus-gene-0.0

## augustus-Scaffold\_999-processed-gene-0.3

## rowData names(0):

## colnames(9): SUS1 SUS2 ... RES3 RES4

## colData names(1): rbtype

Log transform data for ease of viewing

rld <- rlog(dds, blind=FALSE)

rld

## class: DESeqTransform

## dim: 18872 9

## metadata(1): version

## assays(1): ''

## rownames(18872): augustus-Scaffold\_1-processed-gene-1.2

## augustus-Scaffold\_1-processed-gene-3.0 ...

## maker-Scaffold\_998-augustus-gene-0.0

## augustus-Scaffold\_999-processed-gene-0.3

## rowData names(7): baseMean baseVar ... dispFit rlogIntercept

## colnames(9): SUS1 SUS2 ... RES3 RES4

## colData names(2): rbtype sizeFactor

head(assay(rld))

## SUS1 SUS2 SUS3

## augustus-Scaffold\_1-processed-gene-1.2 2.755547 2.7595630 2.7906589

## augustus-Scaffold\_1-processed-gene-3.0 3.033816 3.0283387 3.0157852

## augustus-Scaffold\_1-processed-gene-3.7 5.418481 5.5533021 5.4976565

## augustus-Scaffold\_1-processed-gene-3.6 -0.953468 -0.9259157 -0.9495109

## augustus-Scaffold\_1-processed-gene-3.9 -1.801843 -1.8033043 -1.8029351

## maker-Scaffold\_1-augustus-gene-3.2 7.965819 7.9562191 7.8821325

## SUS4 SUS5 RES1

## augustus-Scaffold\_1-processed-gene-1.2 2.6677391 2.782008 2.6778780

## augustus-Scaffold\_1-processed-gene-3.0 2.9104657 3.073921 2.9446747

## augustus-Scaffold\_1-processed-gene-3.7 5.2168133 5.571040 5.2786479

## augustus-Scaffold\_1-processed-gene-3.6 -0.9519402 -0.953463 -0.9400544

## augustus-Scaffold\_1-processed-gene-3.9 -1.8037713 -1.805309 -1.8039255

## maker-Scaffold\_1-augustus-gene-3.2 8.0416578 7.904672 8.2774507

## RES2 RES3 RES4

## augustus-Scaffold\_1-processed-gene-1.2 2.7323701 2.6640431 2.701823

## augustus-Scaffold\_1-processed-gene-3.0 2.9868966 2.9840345 2.971527

## augustus-Scaffold\_1-processed-gene-3.7 5.2488767 5.3186358 5.343757

## augustus-Scaffold\_1-processed-gene-3.6 -0.9403563 -0.9526872 -0.950746

## augustus-Scaffold\_1-processed-gene-3.9 -1.8040135 -1.7998480 -1.803360

## maker-Scaffold\_1-augustus-gene-3.2 8.0900516 8.0803817 8.054818

head(assays(dds)[["counts"]])

## SUS1 SUS2 SUS3 SUS4 SUS5 RES1 RES2 RES3

## augustus-Scaffold\_1-processed-gene-1.2 20 7 8 1 25 2 7 1

## augustus-Scaffold\_1-processed-gene-3.0 25 8 6 0 33 3 7 7

## augustus-Scaffold\_1-processed-gene-3.7 100 46 34 19 143 27 25 34

## augustus-Scaffold\_1-processed-gene-3.6 0 2 0 0 0 1 1 0

## augustus-Scaffold\_1-processed-gene-3.9 1 0 0 0 0 0 0 1

## maker-Scaffold\_1-augustus-gene-3.2 525 164 124 229 482 330 273 280

## RES4

## augustus-Scaffold\_1-processed-gene-1.2 3

## augustus-Scaffold\_1-processed-gene-3.0 4

## augustus-Scaffold\_1-processed-gene-3.7 26

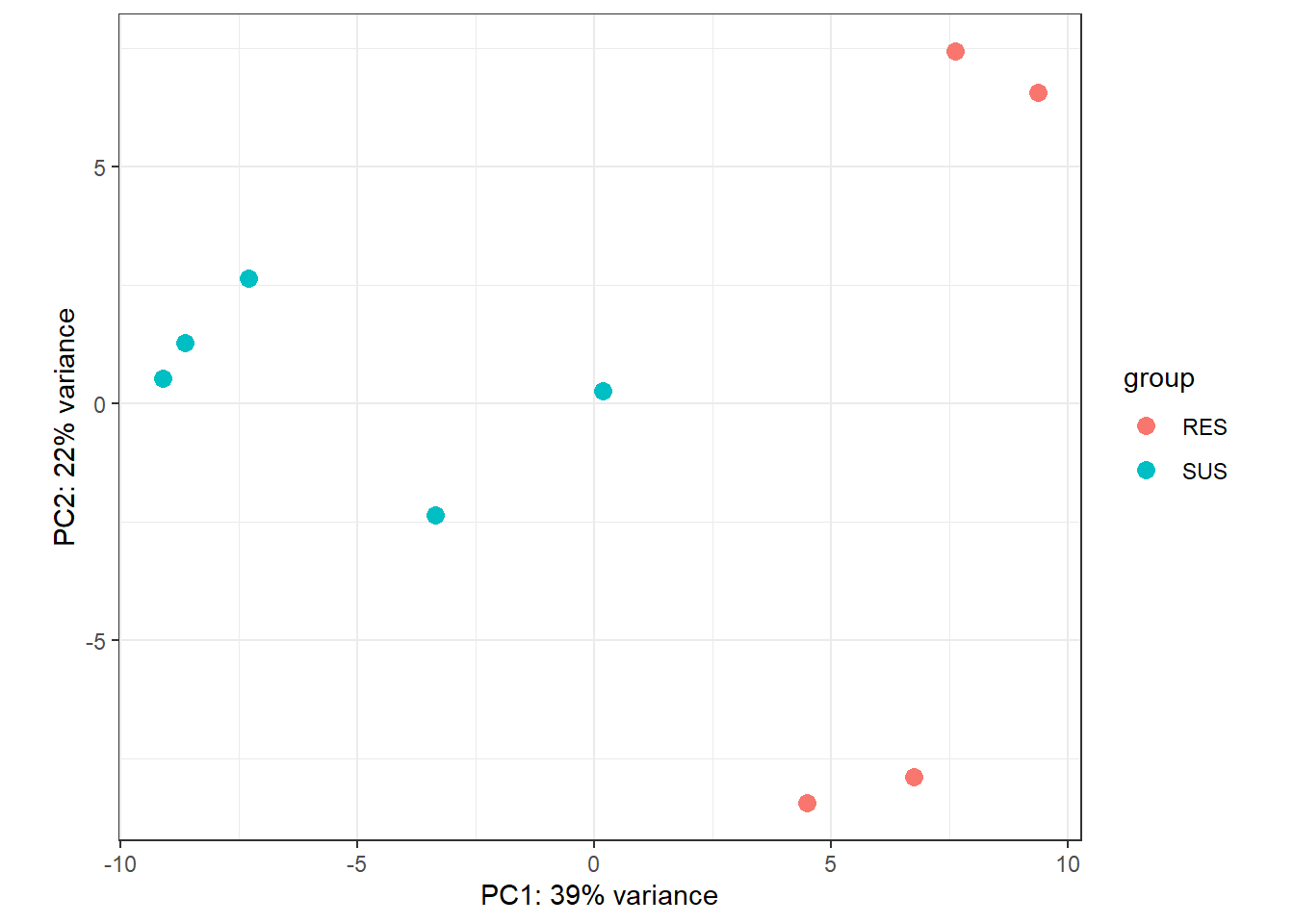
## augustus-Scaffold\_1-processed-gene-3.6 0

## augustus-Scaffold\_1-processed-gene-3.9 0

## maker-Scaffold\_1-augustus-gene-3.2 192

Using ggplot2 to make PCA

plotPCA(rld, intgroup="rbtype") + theme\_bw()



df.PCA <- plotPCA(rld, intgroup="rbtype", returnData=TRUE)

df.PCA

## PC1 PC2 group rbtype name

## SUS1 -8.6451925 1.2837997 SUS SUS SUS1

## SUS2 -7.2914072 2.6477619 SUS SUS SUS2

## SUS3 -9.1130319 0.5148459 SUS SUS SUS3

## SUS4 0.1759198 0.2642074 SUS SUS SUS4

## SUS5 -3.3585805 -2.3691001 SUS SUS SUS5

## RES1 6.7496877 -7.8950568 RES RES RES1

## RES2 4.4838921 -8.4494928 RES RES RES2

## RES3 9.3650327 6.5695534 RES RES RES3

## RES4 7.6336799 7.4334815 RES RES RES4

DeSeq2 Analysis First the size factors need to be estimated

dds <- estimateSizeFactors(dds)

dds

## class: DESeqDataSet

## dim: 18872 9

## metadata(1): version

## assays(1): counts

## rownames(18872): augustus-Scaffold\_1-processed-gene-1.2

## augustus-Scaffold\_1-processed-gene-3.0 ...

## maker-Scaffold\_998-augustus-gene-0.0

## augustus-Scaffold\_999-processed-gene-0.3

## rowData names(0):

## colnames(9): SUS1 SUS2 ... RES3 RES4

## colData names(2): rbtype sizeFactor

sizeFactors(dds)

## SUS1 SUS2 SUS3 SUS4 SUS5 RES1 RES2 RES3

## 2.1762529 0.6917661 0.5813352 0.8619969 2.1620382 0.9269670 0.9661969 1.0033169

## RES4

## 0.7102451

Secondly, the dispersion is estimated

dds <- estimateDispersions(dds)

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

dds

## class: DESeqDataSet

## dim: 18872 9

## metadata(1): version

## assays(2): counts mu

## rownames(18872): augustus-Scaffold\_1-processed-gene-1.2

## augustus-Scaffold\_1-processed-gene-3.0 ...

## maker-Scaffold\_998-augustus-gene-0.0

## augustus-Scaffold\_999-processed-gene-0.3

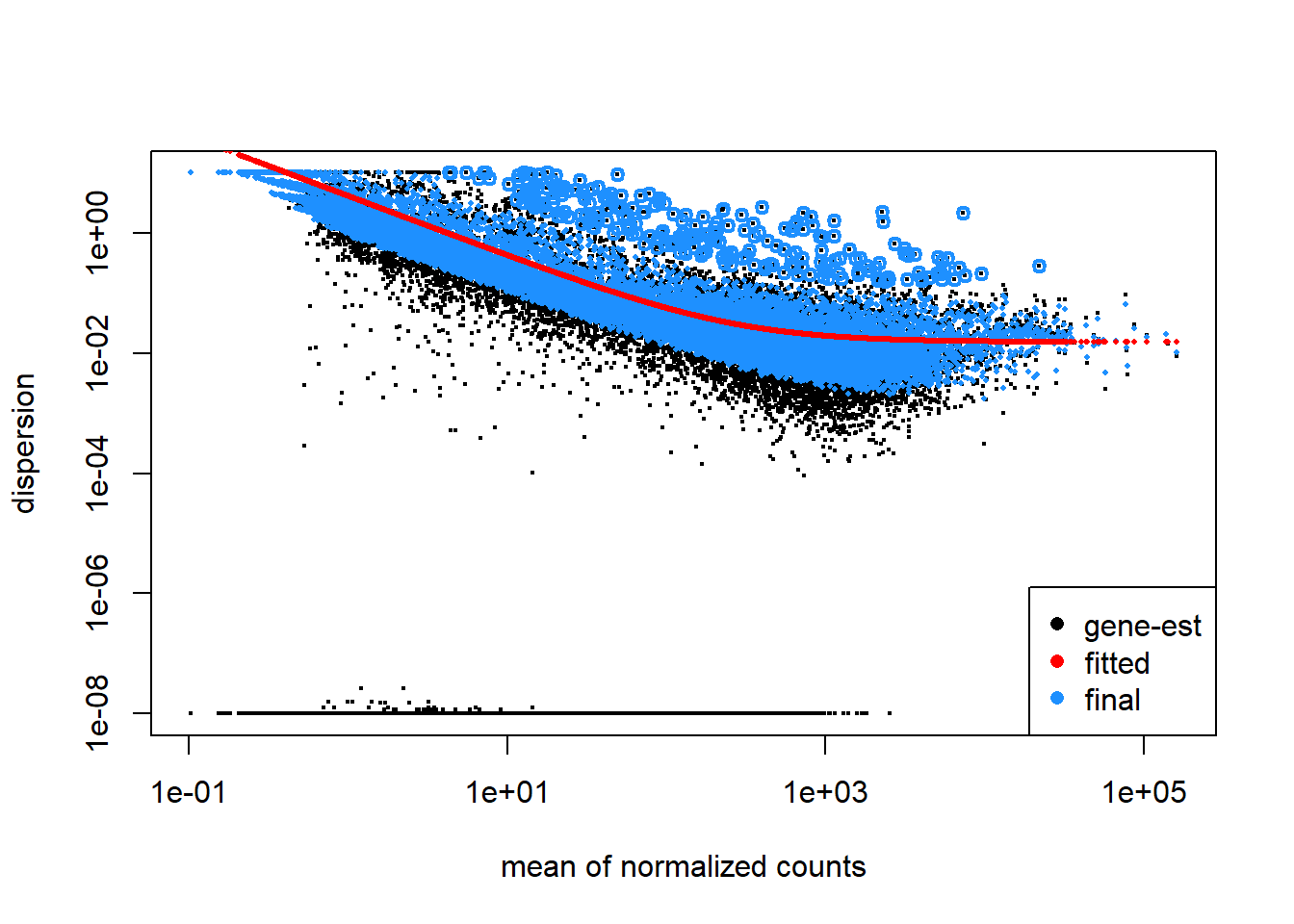
## rowData names(10): baseMean baseVar ... dispOutlier dispMAP

## colnames(9): SUS1 SUS2 ... RES3 RES4

## colData names(2): rbtype sizeFactor

Plotting the dispersion to visualise data

plotDispEsts(dds)



Now to fit a negative binomial GLM and calculate Wald statistic. The Wald test was incorporated to test for differential expression between resistant-breaking (Nr:1) and susceptible (Nr:0) *N. ribisnigri* biotypes.

Null hypothesis = no differential expression between Nr:1 and Nr:0.

dds <-nbinomWaldTest(dds)

dds

## class: DESeqDataSet

## dim: 18872 9

## metadata(1): version

## assays(4): counts mu H cooks

## rownames(18872): augustus-Scaffold\_1-processed-gene-1.2

## augustus-Scaffold\_1-processed-gene-3.0 ...

## maker-Scaffold\_998-augustus-gene-0.0

## augustus-Scaffold\_999-processed-gene-0.3

## rowData names(22): baseMean baseVar ... deviance maxCooks

## colnames(9): SUS1 SUS2 ... RES3 RES4

## colData names(2): rbtype sizeFactor

View results

resultsNames(dds)

## [1] "Intercept" "rbtype\_SUS\_vs\_RES"

To generate more accurate log2 fold change estimates, the apeglm package was utilised to generate LFC estimates (Zhu et al., 2018). The more recent versions of DESeq2 package does not perform shrinkage of the LFC estimates by default. When information of a gene is low, apeglm allows for the shrinkage of the LFC estimates towards zero when the information of a gene is low (i.e. low counts or high dispersion) (Zhu et al., 2018).

Alpha is *P*-value

res <- lfcShrink(dds=dds, coef = 2, type = "apeglm")

alpha <-0.05

View results

summary(res) ; mcols(res)

##

## out of 18872 with nonzero total read count

## adjusted p-value < 0.1

## LFC > 0 (up) : 598, 3.2%

## LFC < 0 (down) : 554, 2.9%

## outliers [1] : 60, 0.32%

## low counts [2] : 6204, 33%

## (mean count < 3)

## [1] see 'cooksCutoff' argument of ?results

## [2] see 'independentFiltering' argument of ?results

## DataFrame with 5 rows and 2 columns

## type description

## <character> <character>

## baseMean intermediate mean of normalized c..

## log2FoldChange results log2 fold change (MA..

## lfcSE results posterior SD: rbtype..

## pvalue results Wald test p-value: r..

## padj results BH adjusted p-values

For ease of viewing, significantly DE genes are ordered by *P*-value

res.ordered <- res[order(res$padj),]

res.ordered ; dim(res.ordered)

## log2 fold change (MAP): rbtype SUS vs RES

## Wald test p-value: rbtype SUS vs RES

## DataFrame with 18872 rows and 5 columns

## baseMean log2FoldChange lfcSE

## <numeric> <numeric> <numeric>

## maker-Scaffold\_2698-augustus-gene-0.8 655.0535 -3.527614 0.119132

## augustus-Scaffold\_463-processed-gene-0.8 481.5691 -2.560592 0.269206

## augustus-Scaffold\_4753-processed-gene-1.4 44.6054 2.876173 0.342276

## augustus-Scaffold\_784-processed-gene-2.7 22.7633 -2.439800 0.367568

## maker-Scaffold\_4214-augustus-gene-0.27 1405.3243 -0.731487 0.121986

## ... ... ... ...

## augustus-Scaffold\_995-processed-gene-0.4 0.102448 0.00218592 0.122421

## maker-Scaffold\_997-augustus-gene-0.1 1.458628 0.00712607 0.122125

## maker-Scaffold\_998-augustus-gene-0.2 0.263067 0.00331658 0.122453

## maker-Scaffold\_998-augustus-gene-0.0 0.632388 -0.00505025 0.122404

## augustus-Scaffold\_999-processed-gene-0.3 0.312881 -0.00387592 0.122487

## pvalue padj

## <numeric> <numeric>

## maker-Scaffold\_2698-augustus-gene-0.8 2.82053e-195 3.55613e-191

## augustus-Scaffold\_463-processed-gene-0.8 1.15147e-22 7.25889e-19

## augustus-Scaffold\_4753-processed-gene-1.4 3.70325e-18 1.55635e-14

## augustus-Scaffold\_784-processed-gene-2.7 1.91550e-12 6.03766e-09

## maker-Scaffold\_4214-augustus-gene-0.27 1.22059e-10 2.88096e-07

## ... ... ...

## augustus-Scaffold\_995-processed-gene-0.4 0.829947 NA

## maker-Scaffold\_997-augustus-gene-0.1 0.561868 NA

## maker-Scaffold\_998-augustus-gene-0.2 0.682352 NA

## maker-Scaffold\_998-augustus-gene-0.0 0.491657 NA

## augustus-Scaffold\_999-processed-gene-0.3 0.560801 NA

## [1] 18872 5

table(res.ordered$padj < alpha)

##

## FALSE TRUE

## 11919 689

Removing genes which DeSeq2 has filtered out

res.filtered <-res.ordered[!is.na(res.ordered$padj),]

res.filtered ; dim(res.filtered)

## log2 fold change (MAP): rbtype SUS vs RES

## Wald test p-value: rbtype SUS vs RES

## DataFrame with 12608 rows and 5 columns

## baseMean log2FoldChange lfcSE

## <numeric> <numeric> <numeric>

## maker-Scaffold\_2698-augustus-gene-0.8 655.0535 -3.527614 0.119132

## augustus-Scaffold\_463-processed-gene-0.8 481.5691 -2.560592 0.269206

## augustus-Scaffold\_4753-processed-gene-1.4 44.6054 2.876173 0.342276

## augustus-Scaffold\_784-processed-gene-2.7 22.7633 -2.439800 0.367568

## maker-Scaffold\_4214-augustus-gene-0.27 1405.3243 -0.731487 0.121986

## ... ... ... ...

## maker-Scaffold\_4257-augustus-gene-4.3 67.5184 9.23453e-05 0.1082653

## augustus-Scaffold\_10-processed-gene-0.3 508.1105 -8.58747e-05 0.0986863

## maker-Scaffold\_475-augustus-gene-7.6 303.0801 1.10025e-04 0.0785460

## maker-Scaffold\_650-augustus-gene-1.4 1675.3757 1.10061e-04 0.0686087

## maker-Scaffold\_772-augustus-gene-1.27 3221.5519 -1.56398e-04 0.1008540

## pvalue padj

## <numeric> <numeric>

## maker-Scaffold\_2698-augustus-gene-0.8 2.82053e-195 3.55613e-191

## augustus-Scaffold\_463-processed-gene-0.8 1.15147e-22 7.25889e-19

## augustus-Scaffold\_4753-processed-gene-1.4 3.70325e-18 1.55635e-14

## augustus-Scaffold\_784-processed-gene-2.7 1.91550e-12 6.03766e-09

## maker-Scaffold\_4214-augustus-gene-0.27 1.22059e-10 2.88096e-07

## ... ... ...

## maker-Scaffold\_4257-augustus-gene-4.3 0.999209 0.999548

## augustus-Scaffold\_10-processed-gene-0.3 0.999517 0.999755

## maker-Scaffold\_475-augustus-gene-7.6 0.999910 0.999989

## maker-Scaffold\_650-augustus-gene-1.4 0.999989 0.999989

## maker-Scaffold\_772-augustus-gene-1.27 0.999889 0.999989

## [1] 12608 5

table(res.filtered$padj < alpha)

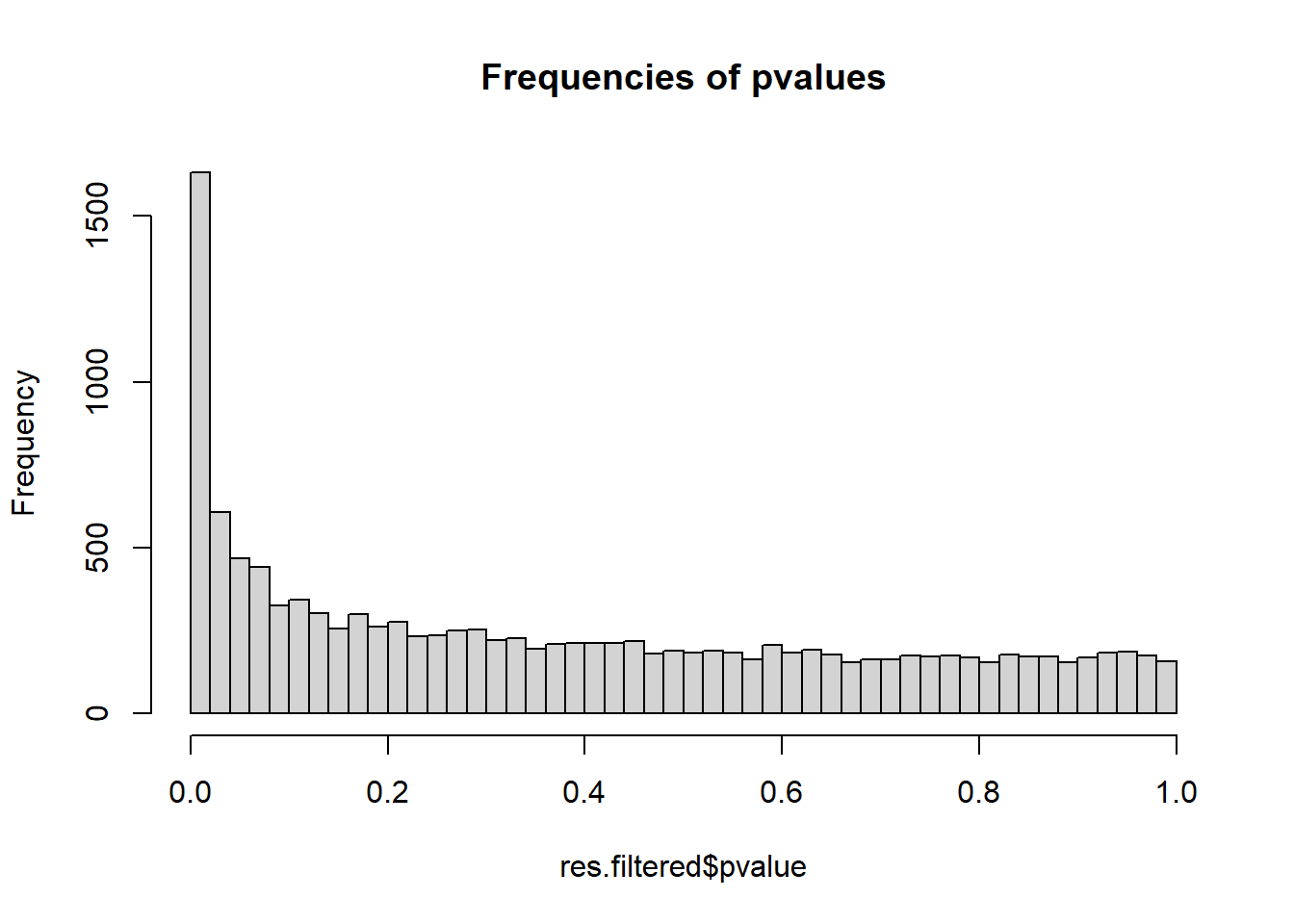
##

## FALSE TRUE

## 11919 689

Plot of adjusted *P* values

hist(res.filtered$pvalue, main="Frequencies of pvalues", breaks=50)



Significance abundance plot (SA) for both filtered and unfiltered data

Used this to create SAplot2

SAplot.2 <- **function**(res, alpha=0.05,**...**){

*#*

*# SA plot for DESEQ2 results*

*#*

data <- data.frame(gene=row.names(res), pvalue=-log10(res$padj), bm=log10(res$baseMean))

data <- na.omit(data)

data <- transform(data, colour=ifelse(pvalue>-log10(alpha), "red", "black"))

*#head(data) ; dim(data)*

*#*

title <- paste("SA plot: p<", alpha, sep='')

plot(data$bm, data$pvalue, main=title, pch=16, cex=0.3, col=data$colour,

xlab ="log base mean count", ylab="-log10(adjusted p-value)", **...**)

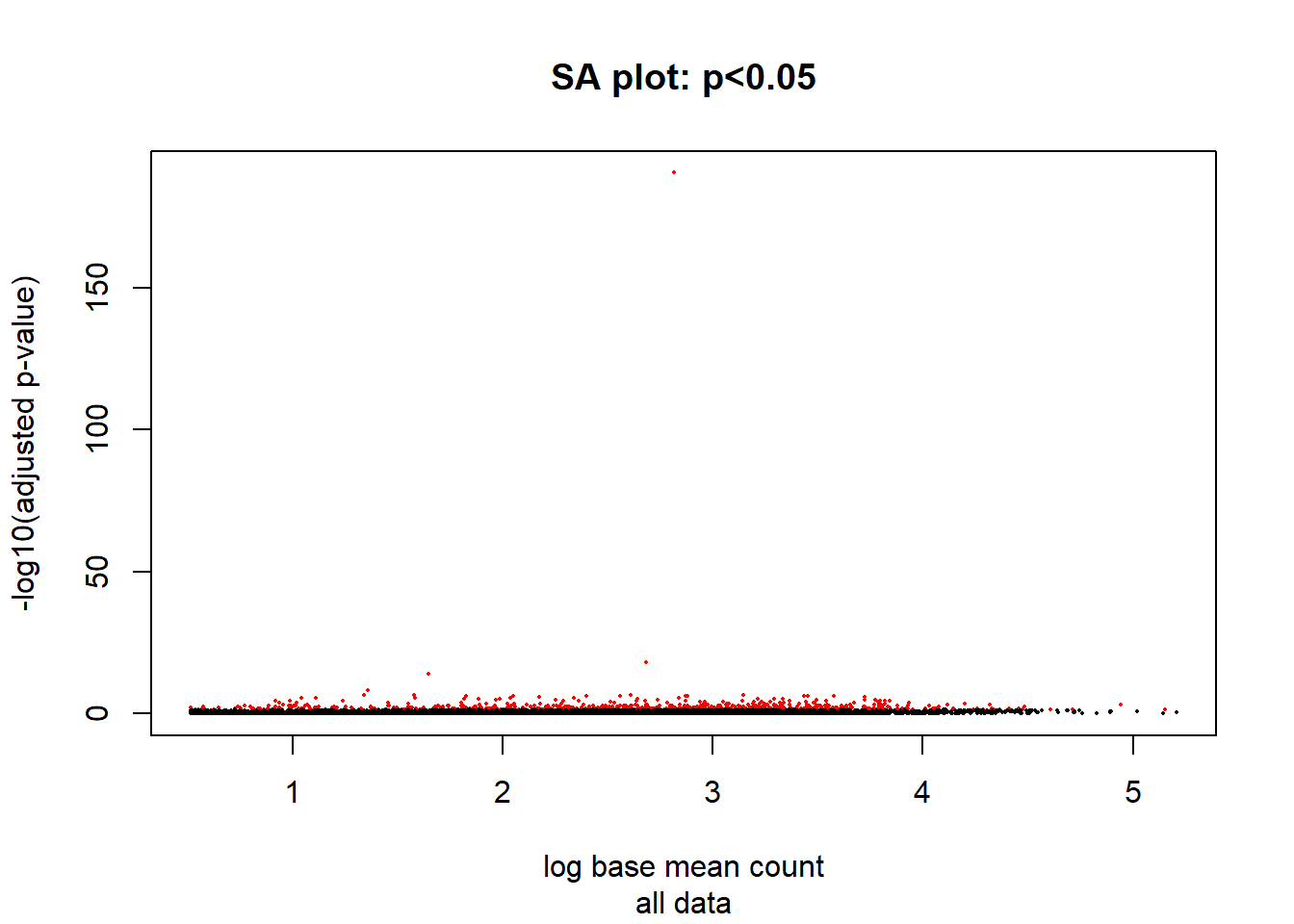
**return**()

}

SAplot

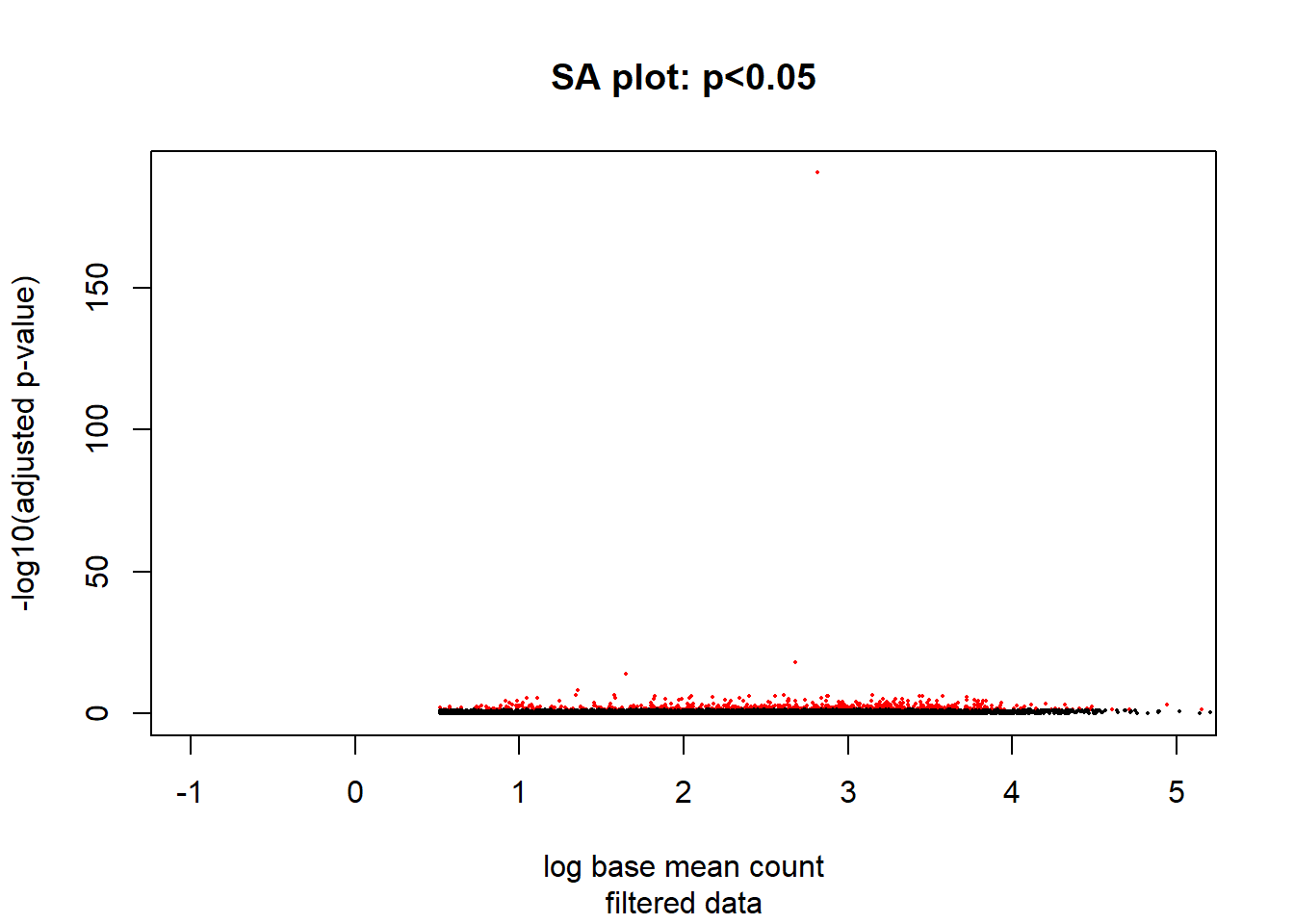
alpha <-0.05

SAplot.2(res.ordered, alpha=alpha, sub="all data")



## NULL

SAplot.2(res.filtered, alpha=alpha, xlim=c(-1,5), sub="filtered data")

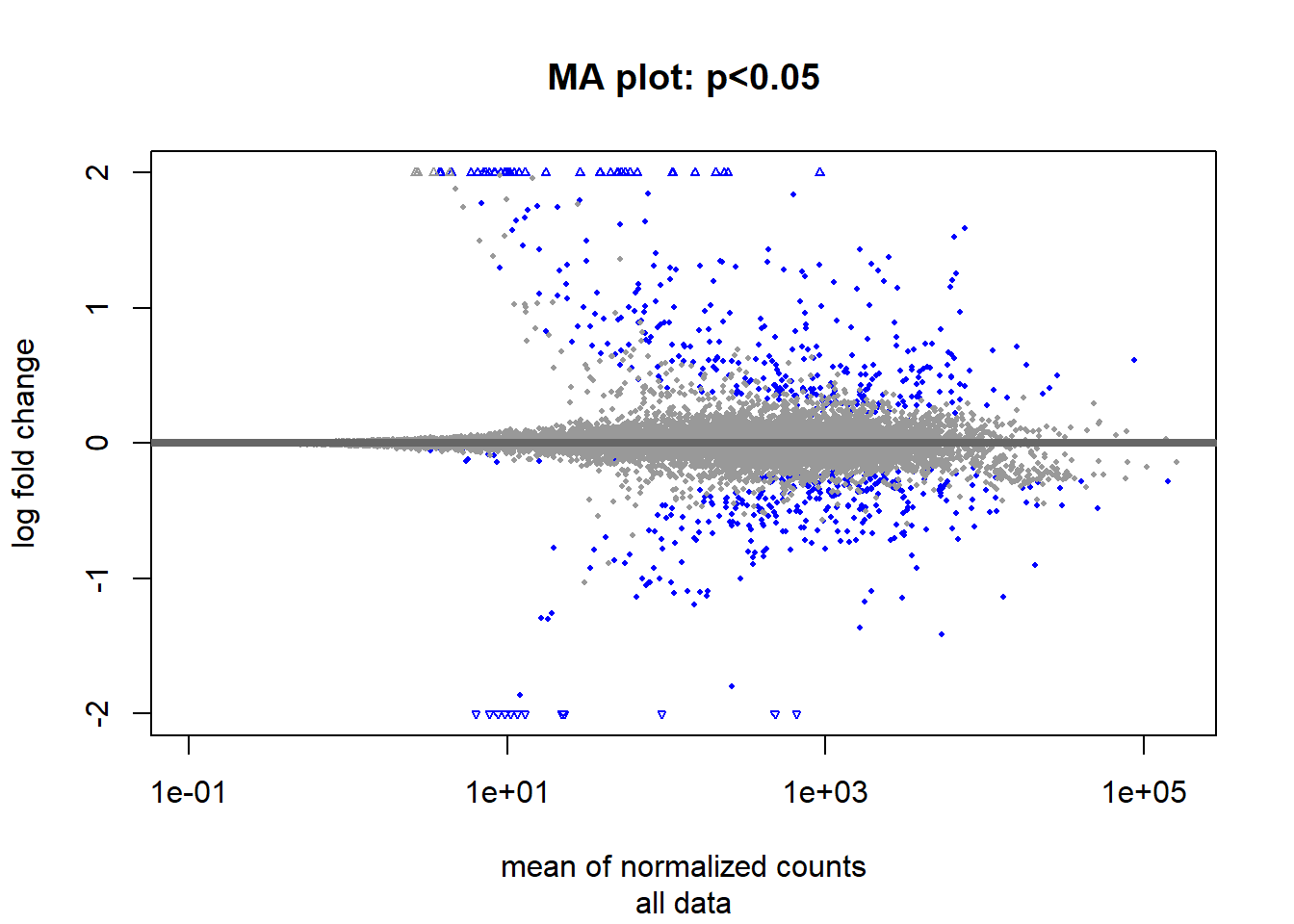


## NULL

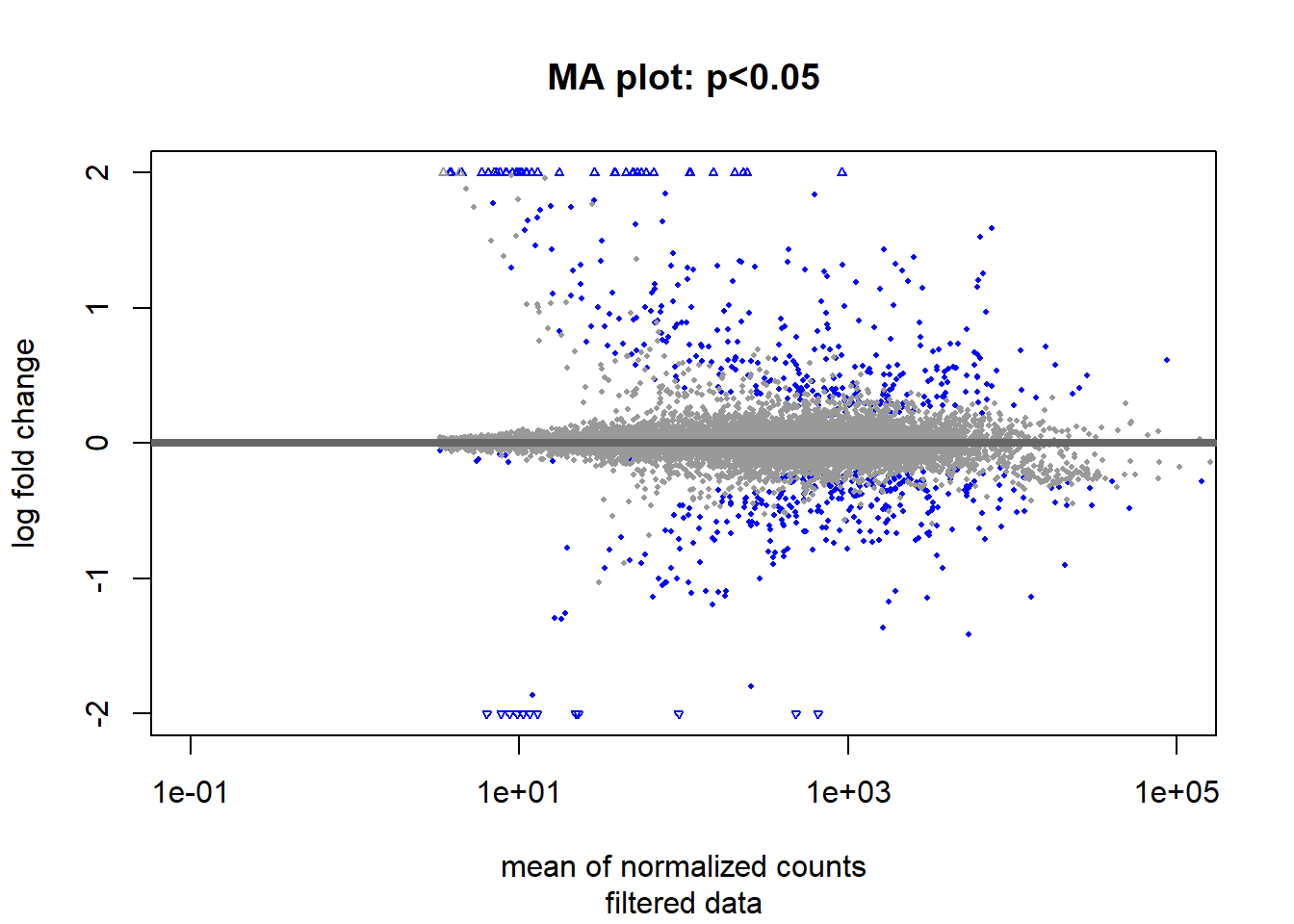
MA plot: log ratio(M) vs. mean average (A)

title <- paste("MA plot: p<", alpha, sep="")

plotMA(res.ordered, ylim=c(-2,2), alpha=alpha, main=title, sub="all data")



plotMA(res.filtered, ylim=c(-2,2), xlim=c(0.1,100000), alpha=alpha, main=title, sub="filtered data")



Visulising DE genes using a volcano plot

**library**(EnhancedVolcano)

## Loading required package: ggrepel

## Registered S3 methods overwritten by 'ggalt':

## method from

## grid.draw.absoluteGrob ggplot2

## grobHeight.absoluteGrob ggplot2

## grobWidth.absoluteGrob ggplot2

## grobX.absoluteGrob ggplot2

## grobY.absoluteGrob ggplot2

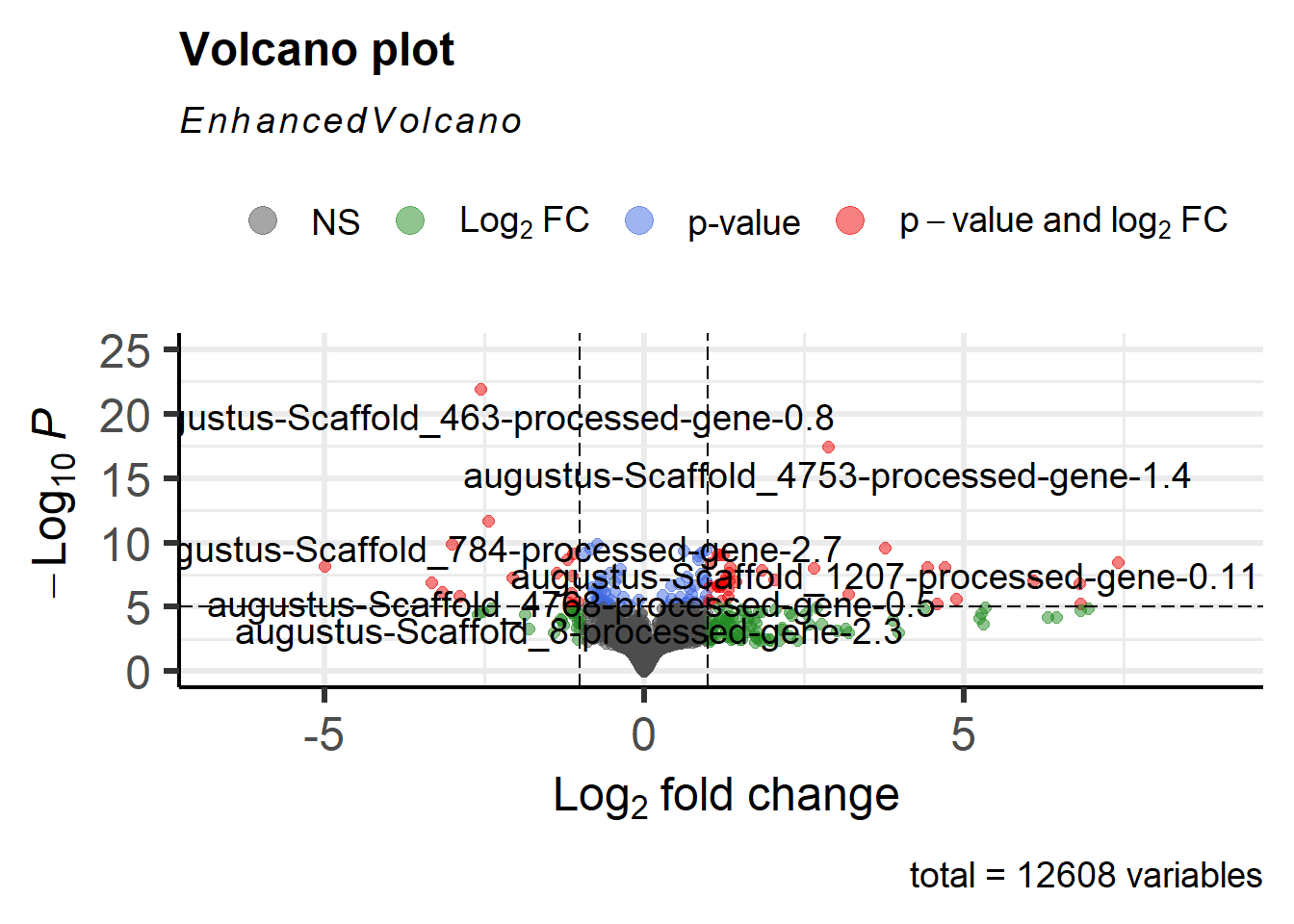
Now for the volcano plot

EnhancedVolcano(res.filtered,

lab = rownames(res.filtered),

x = 'log2FoldChange',

y = 'pvalue', ylim=c(0,25))



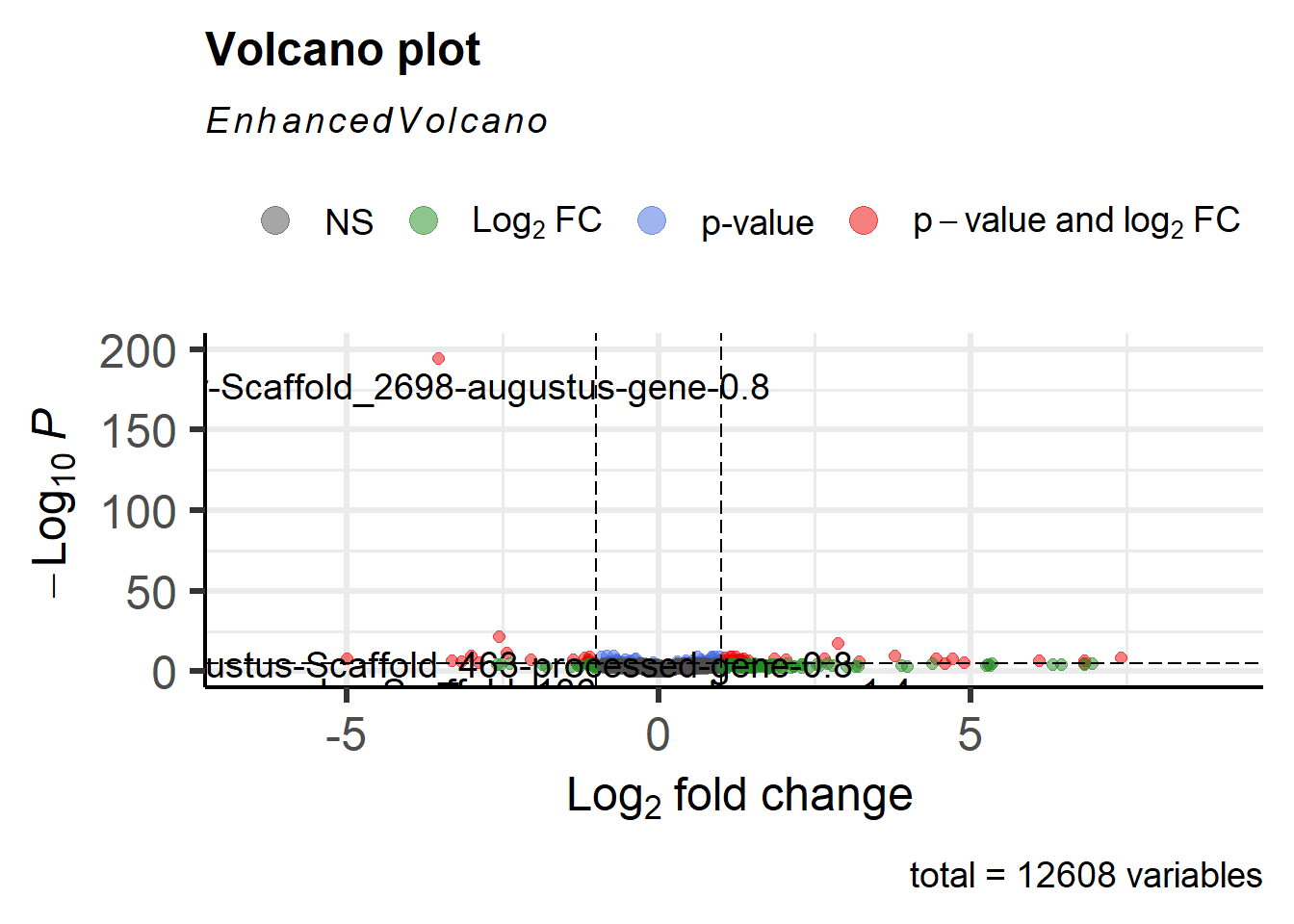
It was found that Y-lim was too strict and cut-off an important DE gene

EnhancedVolcano(res.filtered,

lab = rownames(res.filtered),

x = 'log2FoldChange',

y = 'pvalue', ylim=c(0,200))



Still not a great view, as all other DE genes are compressed. Adapted this code for better visuals and to scale the Y axis to show the DE gene of interest along with the remaining DE genes:

volcanoPlot.2 <- **function**(res, alpha=0.05){

*#*

*# Volcano plot for DESEQ2 results*

*# based on https://twbattaglia.github.io/2016/12/17/volcano-plot/*

*#*

**library**(ggplot2)

**library**(magrittr)

**library**(plyr)

**library**(dplyr)

**library**(ggrepel)

*#*

data <- data.frame(gene = row.names(res), pvalue = -log10(res$padj), lfc = res$log2FoldChange)

data <- na.omit(data)

*#*

*# colour significant genes, depending whether up or down regulated)*

threshold = -log10(alpha)

data <- data %>%

mutate(color = ifelse(data$lfc > 0 & data$pvalue > threshold,

yes = "Up",

no = ifelse(data$lfc < 0 & data$pvalue > threshold,

yes = "Down",

no = "none")))

*#*

return.plot <- ggplot(data, aes(x = lfc, y = pvalue)) +

coord\_cartesian(xlim =rev(c(-5, 5))) +

geom\_point(aes(color = factor(color)), size = 1.75, alpha = 0.8, na.rm = T) + *# add gene points*

theme\_bw(base\_size = 16) + *# clean up theme*

theme(legend.position = "none") + *# remove legend*

ggtitle(label = "Volcano Plot", subtitle = paste("p<" , alpha , sep="")) + *# add title*

xlab(expression(log[2]("Down" / "Up"))) + *# x-axis label*

ylab(expression(-log[10]("adjusted p-value"))) + *# y-axis label*

geom\_vline(xintercept = 0, colour = "black") + *# add line at 0*

geom\_hline(yintercept = threshold, colour = "black") + *# -log10(alpha)*

annotate(geom = "text", label = "Up", x = -2, y = 85, size = 7, colour = "black") + *# add Down text*

annotate(geom = "text", label = "Down", x = 2, y = 85, size = 7, colour = "black") + *# add Up text*

scale\_color\_manual(values = c("Down" = "#E64B35", "Up" = "#3182bd", "none" = "#636363")) *# change colors*

*# Scaled Y-axis with log1p function*

return.plot <- return.plot + scale\_y\_continuous(trans = "log1p")

*# Subset table to show top and bottom gene labels*

labelled <- rbind(top\_n(data, n = -2, wt = lfc) , top\_n(data, n = 0, wt = lfc))

*# Add layer of text annotation*

return.plot <- return.plot +

geom\_text\_repel(data = labelled,

mapping = aes(label = gene),

size = 3,

fontface = 'bold',

color = 'black',

box.padding = unit(0.5, "lines"),

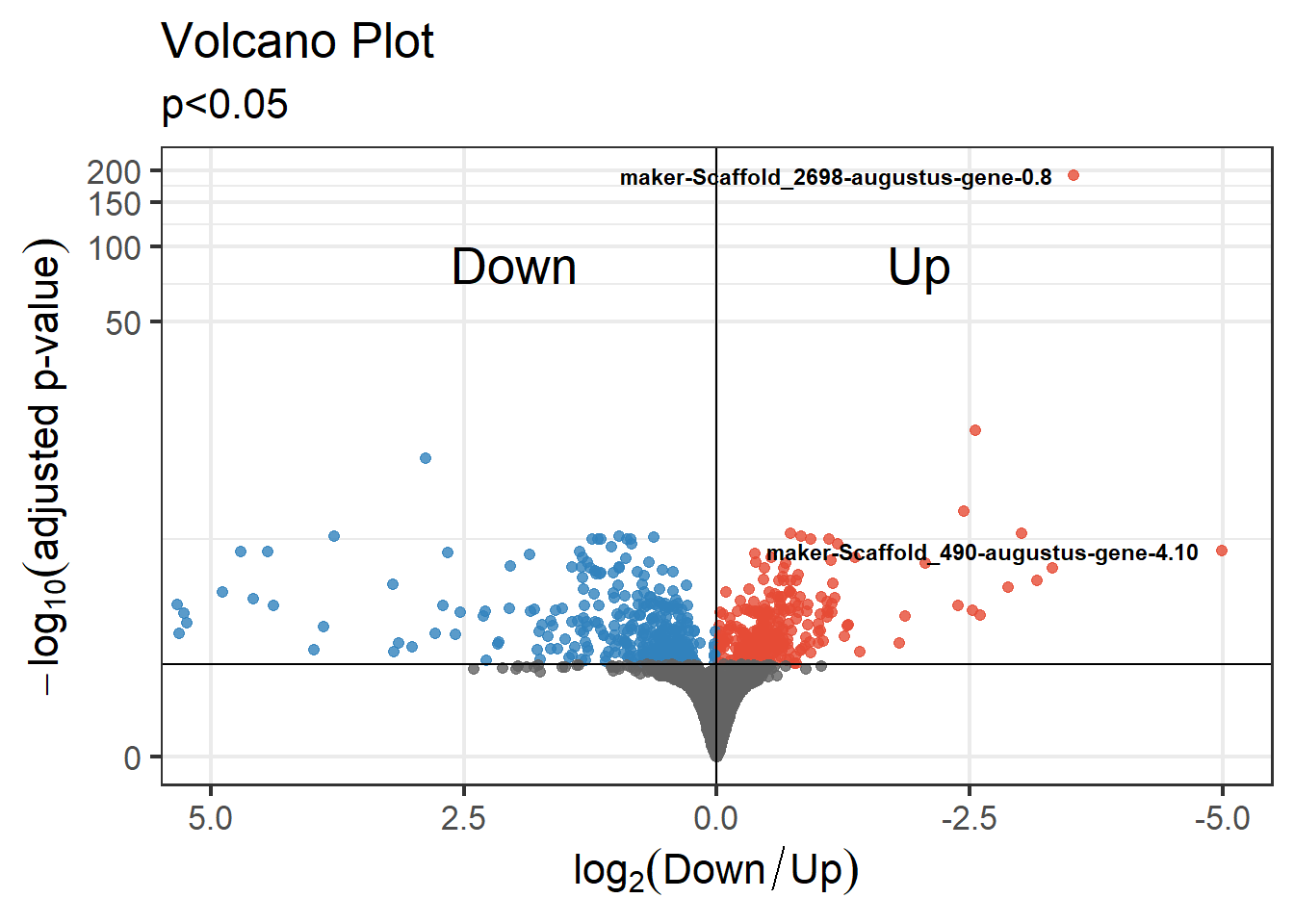
point.padding = unit(0.5, "lines"))

**return**(return.plot)

}

Viewing new volcano plot

volcanoPlot.2(res.filtered, alpha=alpha)



A total of 689 DE genes were identified with a *P*-value of 0.05. A stricter *P*-value was incorporated to provide more informative results.

alpha <-0.00001

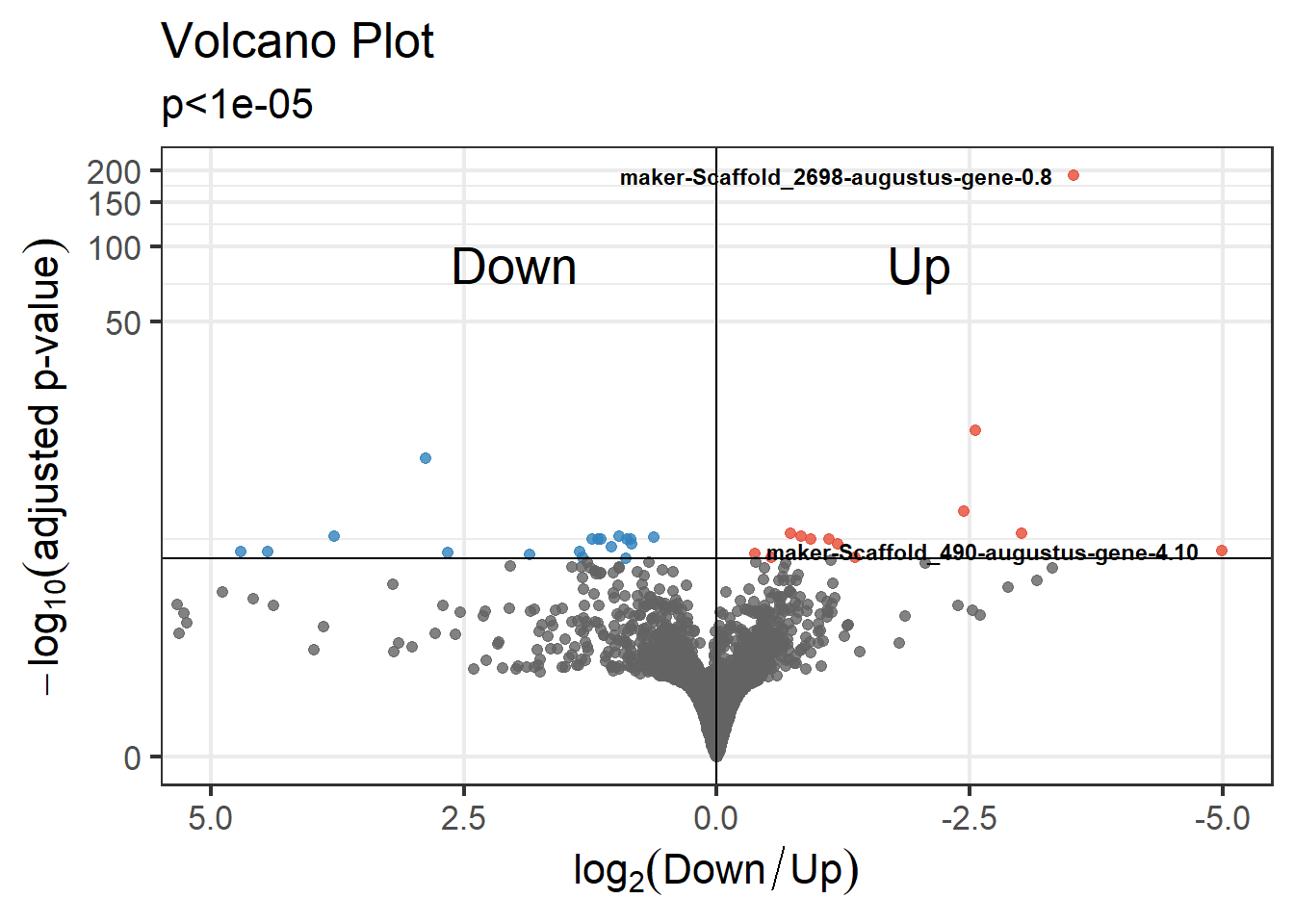
Confirming *P*-value adjustment

DEgenes <- rownames(subset(res.filtered, padj<alpha)) ; length(DEgenes)

## [1] 32

New volcano plot with adjusted *P*-value (0.00001)

volcanoPlot.2(res.filtered, alpha=alpha)



To plot RNA counts for identified 32 DE genes for both resistant-breaking and susceptible *N. ribisnigri* biotypes. This function lists the top 32 DE genes in order of *P*-value

DEgenes <- rownames(subset(res.filtered, padj<alpha)) ; length(DEgenes)

## [1] 32

head(DEgenes, n=32)

## [1] "maker-Scaffold\_2698-augustus-gene-0.8"

## [2] "augustus-Scaffold\_463-processed-gene-0.8"

## [3] "augustus-Scaffold\_4753-processed-gene-1.4"

## [4] "augustus-Scaffold\_784-processed-gene-2.7"

## [5] "maker-Scaffold\_4214-augustus-gene-0.27"

## [6] "maker-Scaffold\_910-augustus-gene-0.6"

## [7] "augustus-Scaffold\_1207-processed-gene-0.11"

## [8] "maker-Scaffold\_821-augustus-gene-1.13"

## [9] "maker-Scaffold\_4762-augustus-gene-30.11"

## [10] "augustus-Scaffold\_1052-processed-gene-1.15"

## [11] "maker-Scaffold\_4732-augustus-gene-5.24"

## [12] "augustus-Scaffold\_634-processed-gene-0.9"

## [13] "augustus-Scaffold\_2666-processed-gene-0.1"

## [14] "maker-Scaffold\_200-augustus-gene-0.17"

## [15] "maker-Scaffold\_2061-augustus-gene-1.1"

## [16] "augustus-Scaffold\_2128-processed-gene-0.25"

## [17] "maker-Scaffold\_647-augustus-gene-1.3"

## [18] "augustus-Scaffold\_134-processed-gene-6.11"

## [19] "augustus-Scaffold\_82-processed-gene-1.8"

## [20] "augustus-Scaffold\_1158-processed-gene-0.1"

## [21] "augustus-Scaffold\_2243-processed-gene-0.15"

## [22] "maker-Scaffold\_490-augustus-gene-4.10"

## [23] "maker-Scaffold\_200-augustus-gene-0.18"

## [24] "maker-Scaffold\_922-augustus-gene-0.7"

## [25] "augustus-Scaffold\_1043-processed-gene-0.0"

## [26] "augustus-Scaffold\_2727-processed-gene-0.1"

## [27] "augustus-Scaffold\_1068-processed-gene-4.10"

## [28] "augustus-Scaffold\_4330-processed-gene-1.6"

## [29] "maker-Scaffold\_119-augustus-gene-0.4"

## [30] "maker-Scaffold\_4299-augustus-gene-0.56"

## [31] "augustus-Scaffold\_4700-processed-gene-2.10"

## [32] "augustus-Scaffold\_49-processed-gene-4.9"

All 32 DE genes were plotted to view RNA count data between both resistant-breaking and susceptible *N. ribisnigri* biotypes

DE genes 1-10

par(mfrow = c(2, 5))

mygene1="maker-Scaffold\_2698-augustus-gene-0.8"

mygene2="augustus-Scaffold\_463-processed-gene-0.8"

mygene3="augustus-Scaffold\_4753-processed-gene-1.4"

mygene4="augustus-Scaffold\_784-processed-gene-2.7"

mygene5="maker-Scaffold\_4214-augustus-gene-0.27"

mygene6="maker-Scaffold\_910-augustus-gene-0.6"

mygene7="augustus-Scaffold\_1207-processed-gene-0.11"

mygene8="maker-Scaffold\_821-augustus-gene-1.13"

mygene9="maker-Scaffold\_4762-augustus-gene-30.11"

mygene10="augustus-Scaffold\_1052-processed-gene-1.15"

plotCounts(dds, gene=mygene1, intgroup="rbtype", pch=19, main=mygene1)

plotCounts(dds, gene=mygene2, intgroup="rbtype", pch=19, main=mygene2)

plotCounts(dds, gene=mygene3, intgroup="rbtype", pch=19, main=mygene3)

plotCounts(dds, gene=mygene4, intgroup="rbtype", pch=19, main=mygene4)

plotCounts(dds, gene=mygene5, intgroup="rbtype", pch=19, main=mygene5)

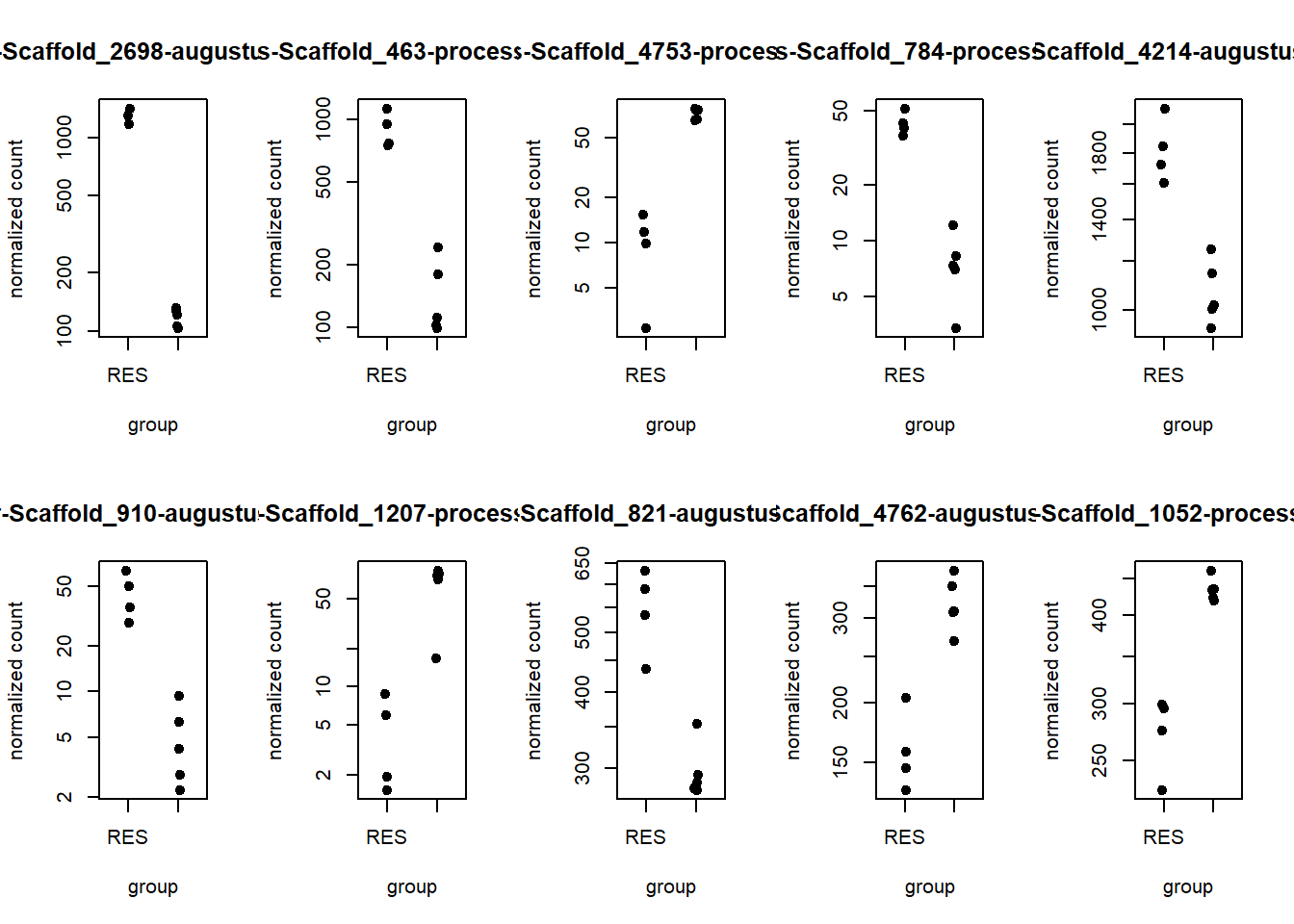
plotCounts(dds, gene=mygene6, intgroup="rbtype", pch=19, main=mygene6)

plotCounts(dds, gene=mygene7, intgroup="rbtype", pch=19, main=mygene7)

plotCounts(dds, gene=mygene8, intgroup="rbtype", pch=19, main=mygene8)

plotCounts(dds, gene=mygene9, intgroup="rbtype", pch=19, main=mygene9)

plotCounts(dds, gene=mygene10, intgroup="rbtype", pch=19, main=mygene10)



DE genes 11-20

par(mfrow = c(2, 5))

mygene11="maker-Scaffold\_4732-augustus-gene-5.24"

mygene12="augustus-Scaffold\_634-processed-gene-0.9"

mygene13="augustus-Scaffold\_2666-processed-gene-0.1"

mygene14="maker-Scaffold\_200-augustus-gene-0.17"

mygene15="maker-Scaffold\_2061-augustus-gene-1.1"

mygene16="augustus-Scaffold\_2128-processed-gene-0.25"

mygene17="maker-Scaffold\_647-augustus-gene-1.3"

mygene18="augustus-Scaffold\_134-processed-gene-6.11"

mygene19="augustus-Scaffold\_82-processed-gene-1.8"

mygene20="augustus-Scaffold\_1158-processed-gene-0.1"

plotCounts(dds, gene=mygene11, intgroup="rbtype", pch=19, main=mygene11)

plotCounts(dds, gene=mygene12, intgroup="rbtype", pch=19, main=mygene12)

plotCounts(dds, gene=mygene13, intgroup="rbtype", pch=19, main=mygene13)

plotCounts(dds, gene=mygene14, intgroup="rbtype", pch=19, main=mygene14)

plotCounts(dds, gene=mygene15, intgroup="rbtype", pch=19, main=mygene15)

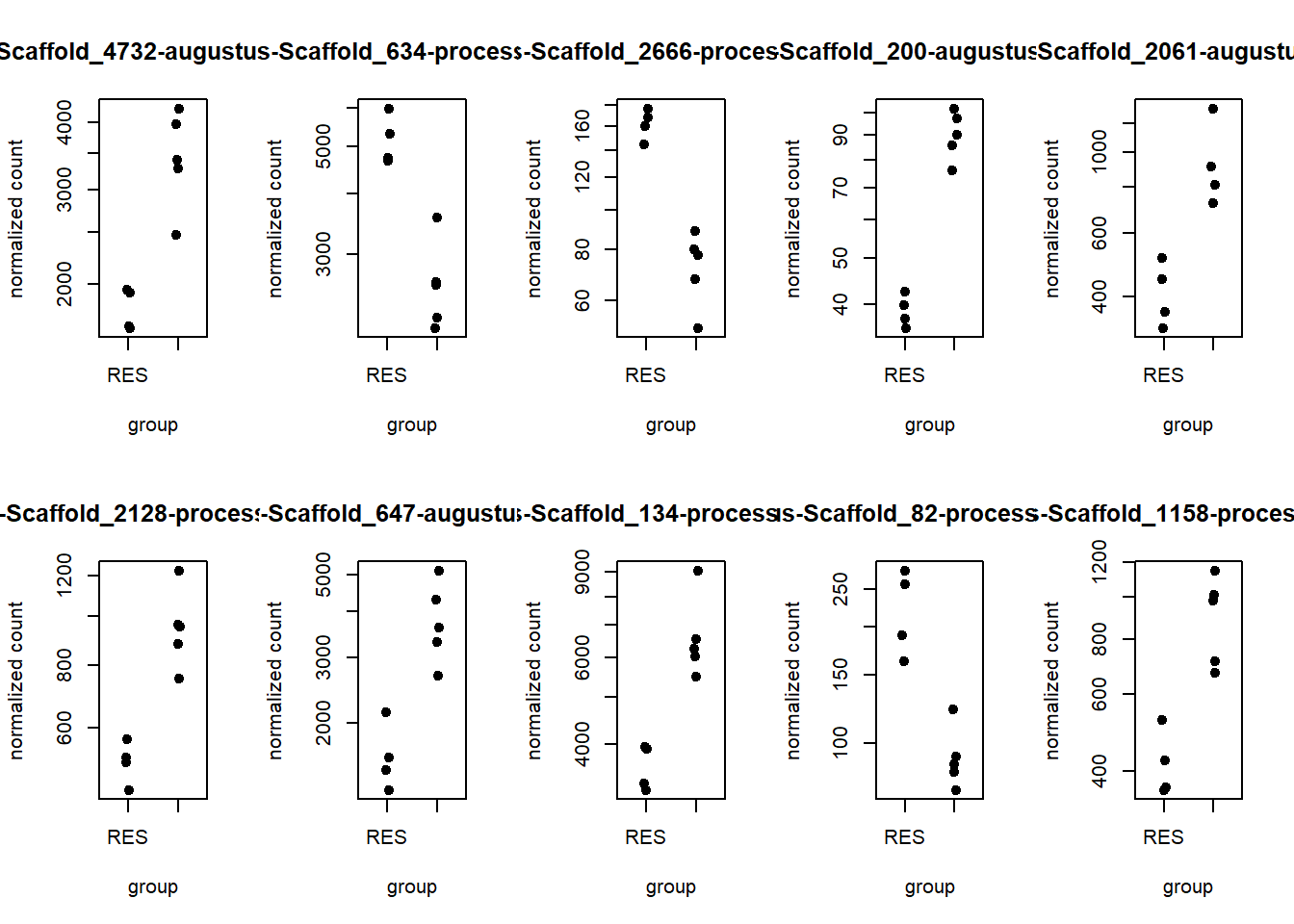
plotCounts(dds, gene=mygene16, intgroup="rbtype", pch=19, main=mygene16)

plotCounts(dds, gene=mygene17, intgroup="rbtype", pch=19, main=mygene17)

plotCounts(dds, gene=mygene18, intgroup="rbtype", pch=19, main=mygene18)

plotCounts(dds, gene=mygene19, intgroup="rbtype", pch=19, main=mygene19)

plotCounts(dds, gene=mygene20, intgroup="rbtype", pch=19, main=mygene20)



DE genes 21-30

par(mfrow = c(2, 5))

mygene21="augustus-Scaffold\_2243-processed-gene-0.15"

mygene22="maker-Scaffold\_490-augustus-gene-4.10"

mygene23="maker-Scaffold\_200-augustus-gene-0.18"

mygene24="maker-Scaffold\_922-augustus-gene-0.7"

mygene25="augustus-Scaffold\_1043-processed-gene-0.0"

mygene26="augustus-Scaffold\_2727-processed-gene-0.1"

mygene27="augustus-Scaffold\_1068-processed-gene-4.10"

mygene28="augustus-Scaffold\_4330-processed-gene-1.6"

mygene29="maker-Scaffold\_119-augustus-gene-0.4"

mygene30="maker-Scaffold\_4299-augustus-gene-0.56"

plotCounts(dds, gene=mygene21, intgroup="rbtype", pch=19, main=mygene21)

plotCounts(dds, gene=mygene22, intgroup="rbtype", pch=19, main=mygene22)

plotCounts(dds, gene=mygene23, intgroup="rbtype", pch=19, main=mygene23)

plotCounts(dds, gene=mygene24, intgroup="rbtype", pch=19, main=mygene24)

plotCounts(dds, gene=mygene25, intgroup="rbtype", pch=19, main=mygene25)

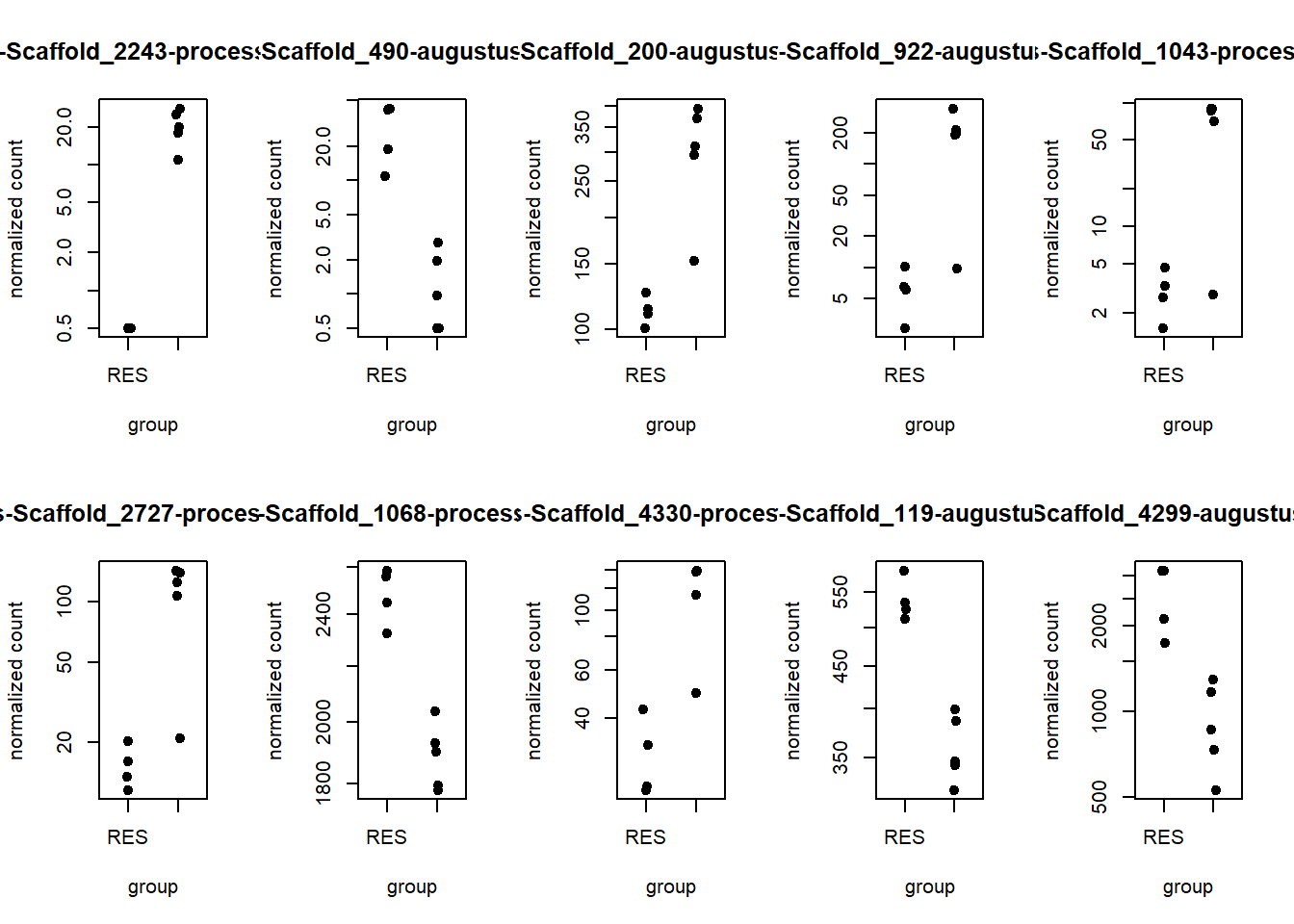
plotCounts(dds, gene=mygene26, intgroup="rbtype", pch=19, main=mygene26)

plotCounts(dds, gene=mygene27, intgroup="rbtype", pch=19, main=mygene27)

plotCounts(dds, gene=mygene28, intgroup="rbtype", pch=19, main=mygene28)

plotCounts(dds, gene=mygene29, intgroup="rbtype", pch=19, main=mygene29)

plotCounts(dds, gene=mygene30, intgroup="rbtype", pch=19, main=mygene30)



DE genes 31-32

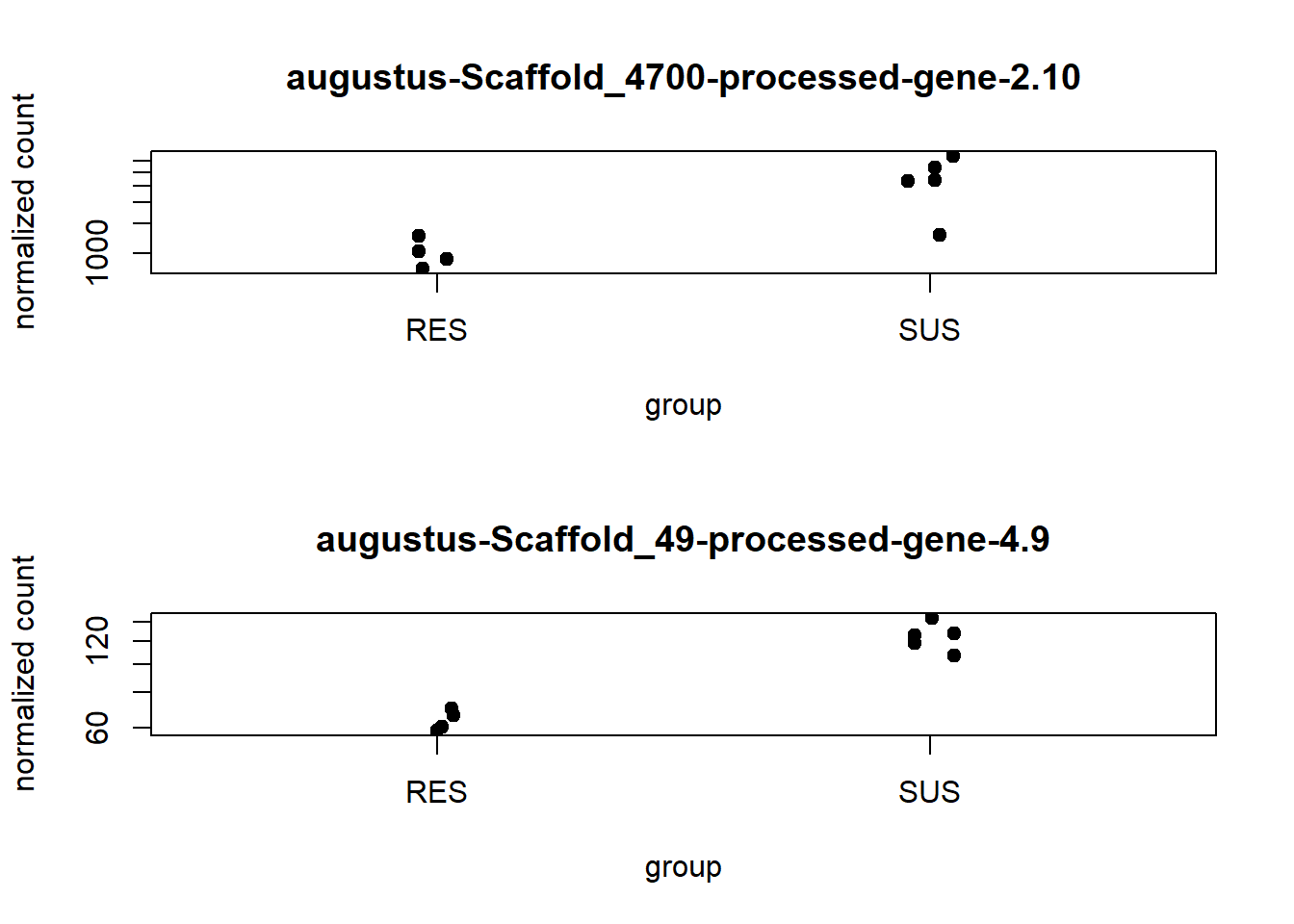
par(mfrow = c(2, 1))

mygene31="augustus-Scaffold\_4700-processed-gene-2.10"

mygene32="augustus-Scaffold\_49-processed-gene-4.9"

plotCounts(dds, gene=mygene31, intgroup="rbtype", pch=19, main=mygene31)

plotCounts(dds, gene=mygene32, intgroup="rbtype", pch=19, main=mygene32)



par(mfrow = c(1, 1))

par(mfrow = c(1, 1))

Producing a Heatmap for the top 32 DE genes between the resistant-breaking and susceptible *N. ribisnigri* biotypes

top <-32

mygenes <-head(DEgenes,top)

hm <- assay(rld)[mygenes,]

hm <- hm - rowMeans(hm)

pheatmap(hm, scale="row", main=paste("Heatmap: p<", alpha, sep="") )

