

# recount (gene and exon analyses)

*Kai Kammers and Shannon Ellis*

*June 09, 2016*

## Contents

Gene level analysis	1
Gene set analysis	12
Exon level analysis	15
Concordance (genes)	28
Concordance (GO groups)	31
Reproducibility	34

Included is an example of how to download and analyze expression data from SRA study SRP032798. The data come from human breast cancer samples, and we compare the transcriptomes of triple negative breast cancer(TNBC) and HER2-positive breast cancer samples. Code here demonstrates how to carry out differential gene, exon, expressed region, and exon-exon junction analyses within a single study using ‘limma’ and ‘voom’. We test for concordance among the results of each analysis and demonstrate how to carry out gene ontology analysis using ‘topGO’ to characterize top hits from differential expression analyses.

We first load the required packages.

```
## load libraries
library('recount')
library('SummarizedExperiment')
library('limma')
library('edgeR')
library('qvalue')
library('topGO')
library('matrixStats')
library('RSkittleBrewer')
library('derfinder')
```

## Gene level analysis

We first download the project of interest (SRP032798), obtaining expression data for the study of interest. We obtain summaries of the number of samples and genes included using colData() and rowData(), respectively.

```
## Find the project of interest (SRP032789), e.g. with parts of the abstract
project_info <- abstract_search('To define the digital transcriptome of three breast cancer')

## Explore information
project_info

##      number_samples species
## 865           20    human
##
```



```

##                                     <numeric>  <logical>
## SRR1027171                         1      TRUE
## SRR1027173                         1      TRUE
## SRR1027174                         1      TRUE
## SRR1027175                         1      TRUE
## SRR1027176                         1      TRUE
## ...
## SRR1027187                         1.0000000  TRUE
## SRR1027188                         1.0000000  TRUE
## SRR1027189                         1.0000000  TRUE
## SRR1027190                         1.0000000  TRUE
## SRR1027172                         0.7273084  TRUE
##           sra_misreported_paired_end mapped_read_count      auc
##                                     <logical>  <integer>  <numeric>
## SRR1027171                         FALSE    86949307 5082692127
## SRR1027173                         FALSE   104337779 6077034329
## SRR1027174                         FALSE   95271238 5504462845
## SRR1027175                         FALSE   88820239 5150234117
## SRR1027176                         FALSE   93464650 5416681912
## ...
## ...
## SRR1027187                         FALSE   64697612 3567078255
## SRR1027188                         FALSE   65278500 4856453823
## SRR1027189                         FALSE   65328289 4858587600
## SRR1027190                         FALSE   73911898 5501089036
## SRR1027172                         FALSE   57523391 3351013968
##           sharq_tissue sharq_cell_type biosample_submission_date
##                                     <character>  <character>  <character>
## SRR1027171             breast        esc  2013-11-07T12:40:22.203
## SRR1027173             breast        esc  2013-11-07T12:40:32.283
## SRR1027174             breast        esc  2013-11-07T12:40:28.283
## SRR1027175             breast        esc  2013-11-07T12:40:34.343
## SRR1027176             breast        esc  2013-11-07T12:40:36.303
## ...
## ...
## SRR1027187             breast        esc  2013-11-07T12:40:56.180
## SRR1027188             breast        esc  2013-11-07T12:40:58.170
## SRR1027189             breast        esc  2013-11-07T12:40:20.227
## SRR1027190             breast        esc  2013-11-07T12:40:18.090
## SRR1027172             breast        esc  2013-11-07T12:40:26.217
##           biosample_publication_date biosample_update_date
##                                     <character>  <character>
## SRR1027171  2013-11-08T01:11:17.160 2014-03-07T16:09:38.542
## SRR1027173  2013-11-08T01:11:14.827 2014-03-07T16:09:38.698
## SRR1027174  2013-11-08T01:11:52.283 2014-03-07T16:09:38.637
## SRR1027175  2013-11-08T01:11:15.963 2014-03-07T16:09:38.731
## SRR1027176  2013-11-08T01:11:46.430 2014-03-07T16:09:38.768
## ...
## ...
## SRR1027187  2013-11-08T01:11:29.587 2014-03-07T16:09:39.093
## SRR1027188  2013-11-08T01:12:06.660 2014-03-07T16:09:39.130
## SRR1027189  2013-11-08T01:11:33.080 2014-03-07T16:09:38.498
## SRR1027190  2013-11-08T01:12:11.320 2014-03-07T16:09:38.469
## SRR1027172  2013-11-08T01:11:45.250 2014-03-07T16:09:38.604
##           avg_read_length  bigwig_file
##                                     <integer>  <character>
## SRR1027171            120 SRR1027171.bw

```

```

## SRR1027173          120 SRR1027173.bw
## SRR1027174          120 SRR1027174.bw
## SRR1027175          120 SRR1027175.bw
## SRR1027176          120 SRR1027176.bw
## ...                 ...
## SRR1027187          120 SRR1027187.bw
## SRR1027188          150 SRR1027188.bw
## SRR1027189          150 SRR1027189.bw
## SRR1027190          150 SRR1027190.bw
## SRR1027172          87  SRR1027172.bw

## At the gene level, the row data includes the names of the genes and
## the sum of the reduced exons widths, which can be used for taking into
## account the gene length.
rowData(rse_gene)

```

```

## DataFrame with 23779 rows and 2 columns
##   gene_id bp_length
##   <character> <integer>
## 1 1           4027
## 2 10          1317
## 3 100         1532
## 4 1000        4473
## 5 100008589  5071
## ...         ...
## 23775       9991  8234
## 23776       9992  803
## 23777       9993  4882
## 23778       9994  6763
## 23779       9997  1393

```

Downloaded count data are first scaled to take into account differing coverage between samples. Phenotype data (pheno) are obtained and ordered to match the sample order of the gene expression data (rse\_gene). Only those samples that are HER2-positive or TNBC are included for analysis. Prior to differential gene expression analysis, count data are obtained in matrix format and then filtered to only include those genes with greater than five average normalized counts across all samples.

```

## Scale counts by taking into account the total coverage per sample
rse <- scale_counts(rse_gene)

## download additional phenotype data from
## http://trace.ncbi.nlm.nih.gov/Traces/study/?acc=SRP032789
pheno <- read.table('SraRunTable_SRP032789.txt', sep = '\t',
                     header = TRUE,
                     stringsAsFactors = FALSE)

## obtain correct order for pheno data
pheno <- pheno[match(rse$run, pheno$Run_s), ]
identical(pheno$Run_s, rse$run)

## [1] TRUE
head(cbind(pheno$Run_s, rse$run))

```

```

##      [,1]      [,2]
## [1,] "SRR1027171" "SRR1027171"
## [2,] "SRR1027173" "SRR1027173"

```

```

## [3,] "SRR1027174" "SRR1027174"
## [4,] "SRR1027175" "SRR1027175"
## [5,] "SRR1027176" "SRR1027176"
## [6,] "SRR1027177" "SRR1027177"

## obtain grouping information
colData(rse)$group <- pheno$tumor_type_s
table(colData(rse)$group)

##
## HER2 Positive Breast Tumor      Non-TNBC Breast Tumor
##          5                      6
## Normal Breast Organoids        TNBC Breast Tumor
##          3                      6

## subset data to HER2 and TNBC types
rse <- rse[, rse$group %in% c('HER2 Positive Breast Tumor',
  'TNBC Breast Tumor')]
rse

## class: RangedSummarizedExperiment
## dim: 23779 11
## metadata(0):
## assays(1): counts
## rownames(23779): 1 10 ... 9994 9997
## rowData names(2): gene_id bp_length
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(19): project sample ... bigwig_file group
## obtain count matrix
counts <- assays(rse)$counts

## filter count matrix
filter <- apply(counts, 1, function(x) mean(x) > 5)
counts <- counts[filter, ]
dim(counts)

## [1] 17874     11
counts_genes <- counts

```

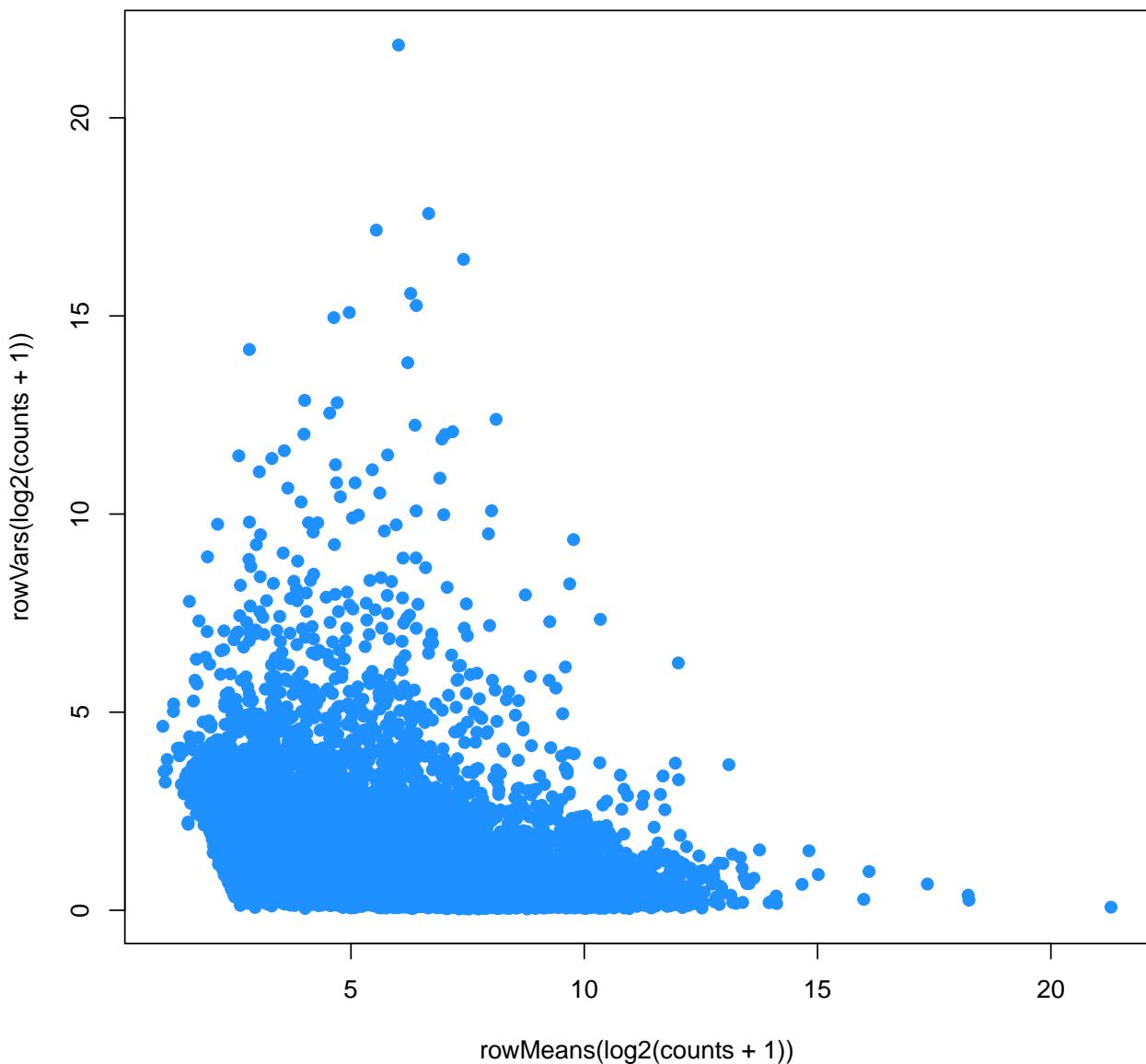
To get a better sense of the data, we plot the mean-variance relationship for each gene. Similarly, we run principal component analysis (PCA) to identify any sample outliers within the data. We assess the variance explained by each of the first 11 PCs as well as visualize the relationship of each sample in the first two PCs.

```

## set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse$group))

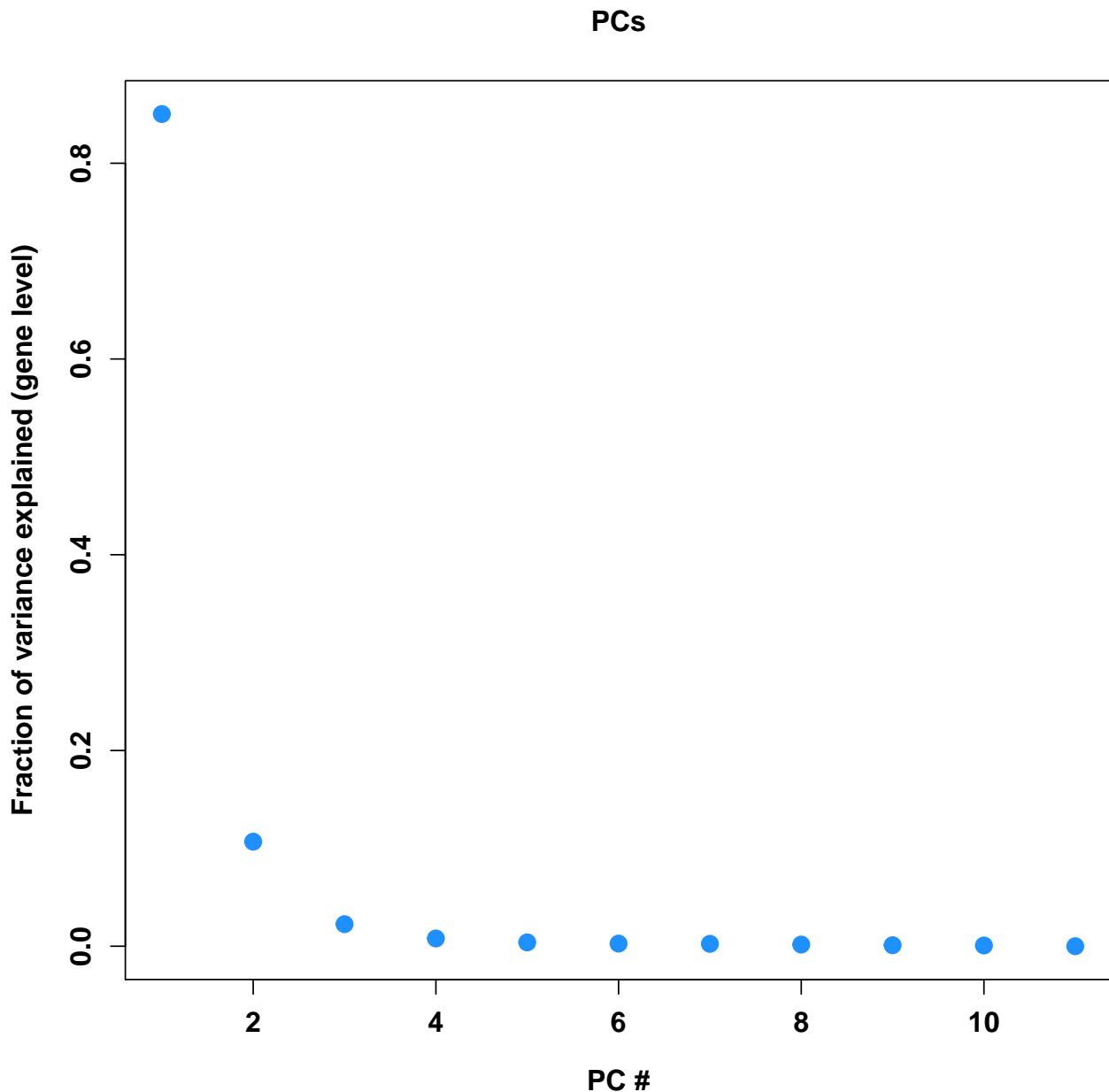
## Look at mean variance relationship
plot(rowMeans(log2(counts + 1)), rowVars(log2(counts + 1)),
  pch = 19, col = trop[2])

```

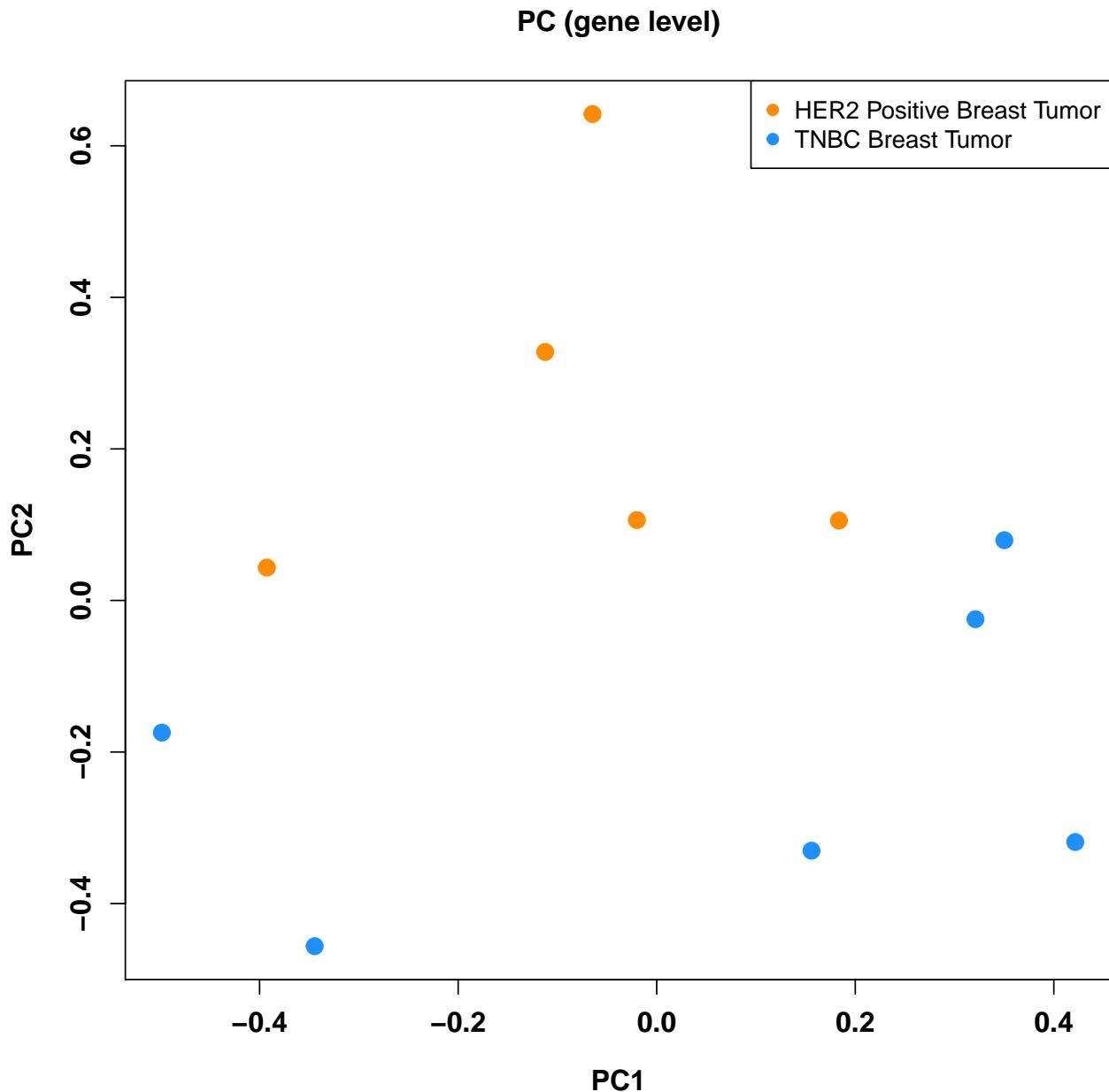


```
## calculate PCs with svd function
expr.pca <- svd(counts - rowMeans(counts))

## plot PCs
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(expr.pca$d^2 / sum(expr.pca$d^2), pch = 19, col = trop[2], cex = 1.5,
     ylab = 'Fraction of variance explained (gene level)', xlab = 'PC #',
     main = 'PCs')
```



```
##plot PC1 vs. PC2
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(expr.pca$v[, 1], expr.pca$v[, 2], pch = 19, col = trop[cols], cex = 1.5,
     xlab = 'PC1', ylab = 'PC2',
     main = 'PC (gene level)')
legend('topright', pch = 19, col = trop[c(1, 2)],
       names(summary(as.factor(rse$group))))
```



Having determined there are no sample outliers in these data, we carry out differential gene expression analysis. Differential gene expression between TNBC and HER2-positive samples are determined using limma and voom. Differentially expressed genes are visualized using a volcano plot to compare the effect size of the differential expression (as measured by the log<sub>2</sub>(fold change) in expression) and its significance (-log<sub>10</sub>p-value).

```
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse$group)
design

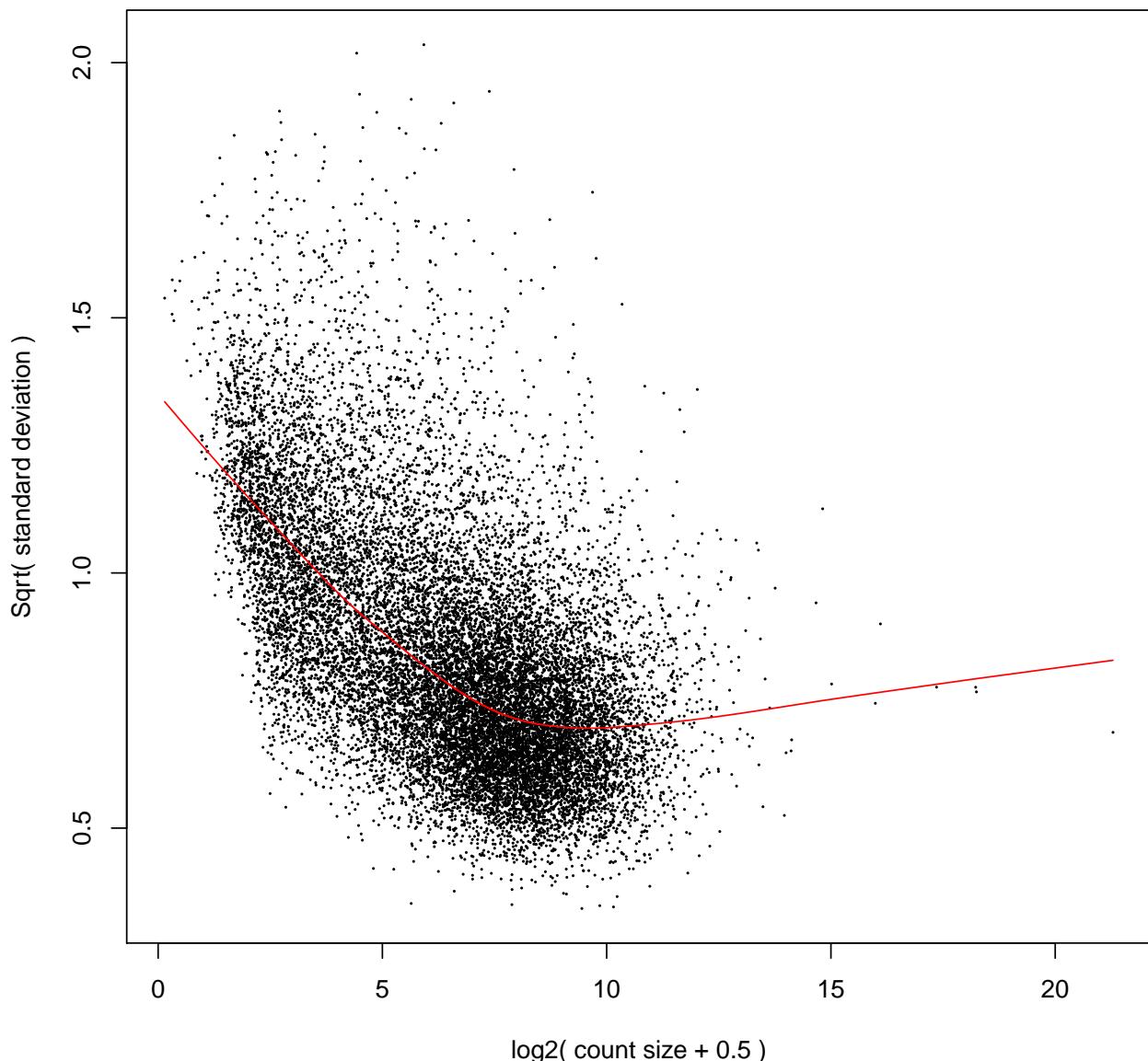
##      (Intercept) rse$groupTNBC Breast Tumor
## 1              1                      1
## 2              1                      1
## 3              1                      1
## 4              1                      1
## 5              1                      1
```

```

## 6      1      0
## 7      1      0
## 8      1      0
## 9      1      0
## 10     1      0
## 11     1      1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`rse$group`
## [1] "contr.treatment"
dge <- DGEList(counts = counts)
dge <- calcNormFactors(dge)
v <- voom(dge, design, plot = TRUE)

```

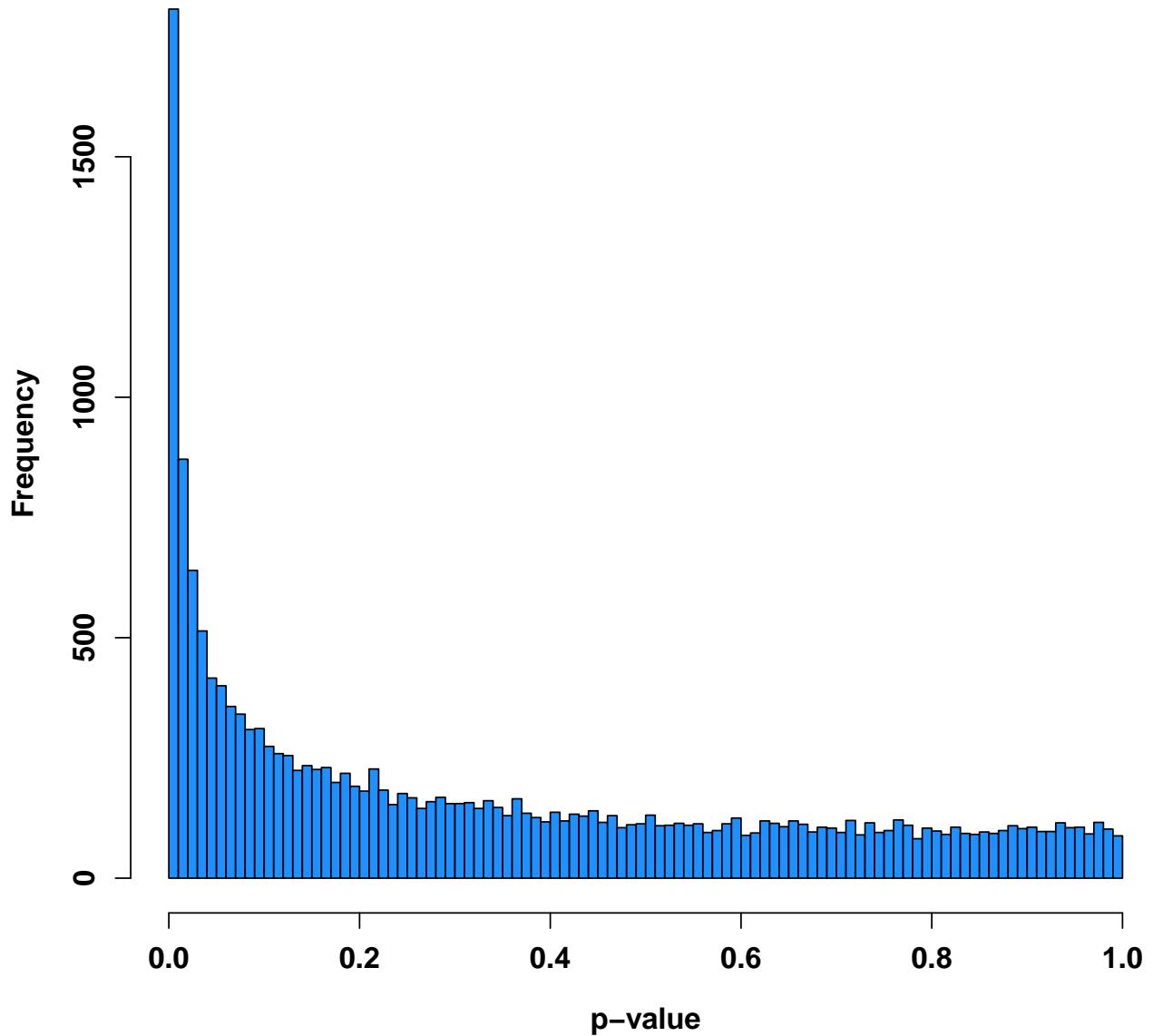
**voom: Mean–variance trend**



```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC <- fit$coefficients[, 2]
p.mod <- fit$p.value[, 2]
q.mod <- qvalue(p.mod)$q
res.genes <- data.frame(log2FC, p.mod, q.mod)
rownames(res.genes) <- rownames(counts)
sum(res.genes$q.mod < 0.05)

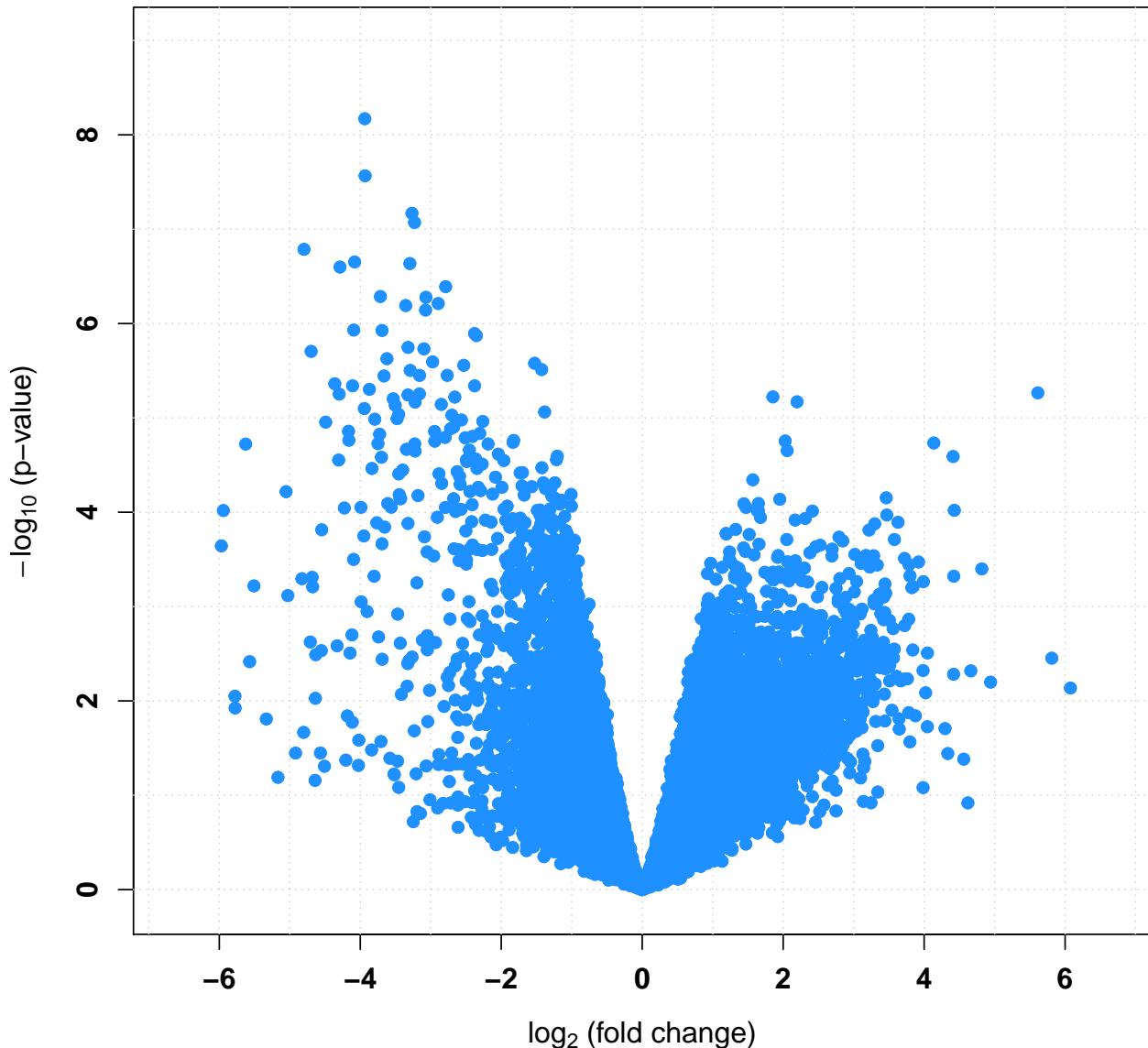
## [1] 1611
## Histogram
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
hist(p.mod, col = trop[2], xlab = 'p-value',
     main = 'Histogramm of p-values', breaks = 100)
```

### Histogramm of p-values



```
## Volcano plot
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
rx2 <- c(-1, 1) * 1.1 * max(abs(log2FC))
ry2 <- c(-0.1, max(-log10(p.mod))) * 1.1
plot(log2FC, -log10(p.mod),
      pch = 19, xlim = rx2, ylim = ry2, col = trop[2],
      xlab = bquote(paste(log[2], ' (fold change)'), ylab = bquote(paste(-log[10], ' (p-value)'))))
abline(v = seq(-10, 10, 1), col = 'lightgray', lty = 'dotted')
abline(h = seq(0, 23, 1), col = 'lightgray', lty = 'dotted')
points(log2FC, -log10(p.mod), pch = 19, col = trop[2])
title('Volcano plot: TNBC vs. HER2+ in SRP032789 (gene level)')
```

### Volcano plot: TNBC vs. HER2+ in SRP032789 (gene level)



### Gene set analysis

To get a better understanding of those genes showing differential gene expression, we utilize `topGO`, a gene set analysis library. Genes included in this analysis are those reaching a q-value cutoff less than 0.05.

```
names(q.mod) <- rownames(counts)
interesting <- function(x) x < 0.05
```

After determining which genes to include for analysis, `topGO` objects are generated and the enrichment tests are run. The Kolmogorov-Smirnov (`ks`) test is used to test for distributional differences. Here, we ask whether each GO group is “enriched” for differentially expressed (`q.mod < 0.05`) genes. Equivalently, we are testing whether the p-value distributions are the same for genes in and outside of each gene ontology. We run tests on the “biological processes” ontology.

```

topgoobjBP <- new('topGOdata',
  description = 'biological process',
  ontology = 'BP', allGenes = q.mod, geneSelectionFun = interesting,
  annotationFun = annFUN.org, mapping = 'org.Hs.eg.db', ID = 'entrez')

##
## Building most specific GOs .....
## ( 10647 GO terms found. )

##
## Build GO DAG topology .....
## ( 14443 GO terms and 34755 relations. )

##
## Annotating nodes .....
## ( 13762 genes annotated to the GO terms. )

bpptest <- runTest(topgoobjBP, algorithm = 'weight01', statistic = 'ks')

##
##           -- Weight01 Algorithm --
##
##           the algorithm is scoring 14443 nontrivial nodes
##           parameters:
##               test statistic: ks
##               score order: increasing

##
##   Level 20: 1 nodes to be scored    (0 eliminated genes)

##
##   Level 19: 7 nodes to be scored    (0 eliminated genes)

##
##   Level 18: 20 nodes to be scored   (1 eliminated genes)

##
##   Level 17: 40 nodes to be scored   (30 eliminated genes)

##
##   Level 16: 130 nodes to be scored  (91 eliminated genes)

##
##   Level 15: 266 nodes to be scored  (179 eliminated genes)

##
##   Level 14: 520 nodes to be scored  (552 eliminated genes)

##
##   Level 13: 925 nodes to be scored  (1159 eliminated genes)

##
##   Level 12: 1336 nodes to be scored (2451 eliminated genes)

##
##   Level 11: 1602 nodes to be scored (4326 eliminated genes)

##
##   Level 10: 1878 nodes to be scored (6122 eliminated genes)

```

```

## 
##   Level 9: 1967 nodes to be scored (8287 eliminated genes)
## 
##   Level 8: 1855 nodes to be scored (9989 eliminated genes)
## 
##   Level 7: 1641 nodes to be scored (11002 eliminated genes)
## 
##   Level 6: 1187 nodes to be scored (11986 eliminated genes)
## 
##   Level 5: 679 nodes to be scored (12493 eliminated genes)
## 
##   Level 4: 293 nodes to be scored (13053 eliminated genes)
## 
##   Level 3: 74 nodes to be scored (13234 eliminated genes)
## 
##   Level 2: 21 nodes to be scored (13397 eliminated genes)
## 
##   Level 1: 1 nodes to be scored (13475 eliminated genes)
bptest

## 
## Description: biological process
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 14443 GO terms scored: 50 terms with p < 0.01
## Annotation data:
##     Annotated genes: 13762
##     Significant genes: 1131
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 14443

bpres_gene <- GenTable(topgoobjBP, pval = bptest,
                       topNodes = length(bptest@score), numChar = 100)
head(bpres_gene, n = 10)

##          GO.ID
## 1  GO:0008589
## 2  GO:0016579
## 3  GO:0030049
## 4  GO:0007050
## 5  GO:0006355
## 6  GO:0070933
## 7  GO:0071557
## 8  GO:0010606
## 9  GO:0001816
## 10 GO:0050673
##                                         Term
## 1 regulation of smoothened signaling pathway
## 2 protein deubiquitination
## 3 muscle filament sliding

```

```

## 4                      cell cycle arrest
## 5          regulation of transcription, DNA-templated
## 6                      histone H4 deacetylation
## 7                      histone H3-K27 demethylation
## 8  positive regulation of cytoplasmic mRNA processing body assembly
## 9                      cytokine production
## 10                     epithelial cell proliferation
##   Annotated Significant Expected      pval
## 1       61        9    5.01 0.00012
## 2      101       14    8.30 0.00032
## 3       30        7    2.47 0.00042
## 4      237       26   19.48 0.00066
## 5     3062      298  251.64 0.00083
## 6       10        0    0.82 0.00084
## 7       4         3    0.33 0.00125
## 8       6         3    0.49 0.00212
## 9     543        25   44.63 0.00214
## 10     318       35   26.13 0.00237

```

## Exon level analysis

As above, we're interested here in differential expression. However, rather than summarizing across genes, this analysis will look for differential expression at the exon level. In this analysis, we include all exons that map to the previous filtered genes and again carry out differential expression analysis using limma and voom.

Here, we download data from the same project as above (SRP032798); however, this time, we are interested in obtaining the exon-level data.

```

## Find a project of interest (SRP032789)
project_info <- abstract_search('To define the digital transcriptome of three breast cancer')
project_info

##   number_samples species
## 865           20   human
##
## 865 Goal: To define the digital transcriptome of three breast cancer subtypes (TNBC, Non-TNBC, and H
##   project
## 865 SRP032789

## Browse the project at SRA
browse_study(project_info$project)

## Download the exon-level RangedSummarizedExperiment data
if(!file.exists(file.path('SRP032789', 'rse_exon.Rdata'))) {
  download_study(project_info$project, type = 'rse-exon')
}

## Load the data
load(file.path(project_info$project, 'rse_exon.Rdata'))
rse_exon

## class: RangedSummarizedExperiment
## dim: 226117 20

```

```

## metadata(0):
## assays(1): counts
## rownames(226117): 1 1 ... 9997 9997
## rowData names(0):
## colnames(20): SRR1027171 SRR1027173 ... SRR1027190 SRR1027172
## colData names(18): project sample ... avg_read_length bigwig_file
## This is the sample phenotype data provided by the recount project
colData(rse_exon)

## DataFrame with 20 rows and 18 columns
##   project      sample experiment       run
##   <character> <character> <character> <character>
## SRR1027171    SRP032789   SRS500214   SRX374850  SRR1027171
## SRR1027173    SRP032789   SRS500216   SRX374852  SRR1027173
## SRR1027174    SRP032789   SRS500217   SRX374853  SRR1027174
## SRR1027175    SRP032789   SRS500218   SRX374854  SRR1027175
## SRR1027176    SRP032789   SRS500219   SRX374855  SRR1027176
## ...
##   ...
##   ...
## SRR1027187    SRP032789   SRS500230   SRX374866  SRR1027187
## SRR1027188    SRP032789   SRS500231   SRX374867  SRR1027188
## SRR1027189    SRP032789   SRS500232   SRX374868  SRR1027189
## SRR1027190    SRP032789   SRS500233   SRX374869  SRR1027190
## SRR1027172    SRP032789   SRS500215   SRX374851  SRR1027172
##   read_count_as_reported_by_sra reads_aligned
##                           <integer> <integer>
## SRR1027171           88869444  88869444
## SRR1027173           107812596 107812596
## SRR1027174           98563260  98563260
## SRR1027175           91327892  91327892
## SRR1027176           96513572  96513572
## ...
##   ...
##   ...
## SRR1027187           75260678  75260678
## SRR1027188           65709192  65709192
## SRR1027189           65801392  65801392
## SRR1027190           74356276  74356276
## SRR1027172           80986440  58902122
##   proportion_of_reads_reported_by_sra_aligned paired_end
##                                         <numeric> <logical>
## SRR1027171           1          TRUE
## SRR1027173           1          TRUE
## SRR1027174           1          TRUE
## SRR1027175           1          TRUE
## SRR1027176           1          TRUE
## ...
##   ...
##   ...
## SRR1027187           1.0000000 TRUE
## SRR1027188           1.0000000 TRUE
## SRR1027189           1.0000000 TRUE
## SRR1027190           1.0000000 TRUE
## SRR1027172           0.7273084 TRUE
##   sra_misreported_paired_end mapped_read_count      auc
##                           <logical> <integer> <numeric>
## SRR1027171           FALSE        86949307 5082692127
## SRR1027173           FALSE        104337779 6077034329
## SRR1027174           FALSE        95271238 5504462845

```

```

## SRR1027175 FALSE 88820239 5150234117
## SRR1027176 FALSE 93464650 5416681912
## ...
## SRR1027187 FALSE 64697612 3567078255
## SRR1027188 FALSE 65278500 4856453823
## SRR1027189 FALSE 65328289 4858587600
## SRR1027190 FALSE 73911898 5501089036
## SRR1027172 FALSE 57523391 3351013968
## shard_tissue shard_cell_type biosample_submission_date
## <character> <character> <character>
## SRR1027171 breast esc 2013-11-07T12:40:22.203
## SRR1027173 breast esc 2013-11-07T12:40:32.283
## SRR1027174 breast esc 2013-11-07T12:40:28.283
## SRR1027175 breast esc 2013-11-07T12:40:34.343
## SRR1027176 breast esc 2013-11-07T12:40:36.303
## ...
## SRR1027187 breast esc 2013-11-07T12:40:56.180
## SRR1027188 breast esc 2013-11-07T12:40:58.170
## SRR1027189 breast esc 2013-11-07T12:40:20.227
## SRR1027190 breast esc 2013-11-07T12:40:18.090
## SRR1027172 breast esc 2013-11-07T12:40:26.217
## biosample_publication_date biosample_update_date
## <character> <character>
## SRR1027171 2013-11-08T01:11:17.160 2014-03-07T16:09:38.542
## SRR1027173 2013-11-08T01:11:14.827 2014-03-07T16:09:38.698
## SRR1027174 2013-11-08T01:11:52.283 2014-03-07T16:09:38.637
## SRR1027175 2013-11-08T01:11:15.963 2014-03-07T16:09:38.731
## SRR1027176 2013-11-08T01:11:46.430 2014-03-07T16:09:38.768
## ...
## SRR1027187 2013-11-08T01:11:29.587 2014-03-07T16:09:39.093
## SRR1027188 2013-11-08T01:12:06.660 2014-03-07T16:09:39.130
## SRR1027189 2013-11-08T01:11:33.080 2014-03-07T16:09:38.498
## SRR1027190 2013-11-08T01:12:11.320 2014-03-07T16:09:38.469
## SRR1027172 2013-11-08T01:11:45.250 2014-03-07T16:09:38.604
## avg_read_length bigwig_file
## <integer> <character>
## SRR1027171 120 SRR1027171.bw
## SRR1027173 120 SRR1027173.bw
## SRR1027174 120 SRR1027174.bw
## SRR1027175 120 SRR1027175.bw
## SRR1027176 120 SRR1027176.bw
## ...
## SRR1027187 120 SRR1027187.bw
## SRR1027188 150 SRR1027188.bw
## SRR1027189 150 SRR1027189.bw
## SRR1027190 150 SRR1027190.bw
## SRR1027172 87 SRR1027172.bw

```

As above, downloaded count data are first scaled to take into account differing coverage between samples. The same phenotype data (pheno) are used and again ordered to match the sample order of the expression data (rse\_exon). Only those samples that are HER2-positive or TNBC are included for analysis. Prior to differential exon expression analysis, count data are obtained in matrix format and then filtered to only include exons within genes that had been analyzed previously.

```

## Scale counts by taking into account the total coverage per sample
rse <- scale_counts(rse_exon)

## download pheno data from
## http://trace.ncbi.nlm.nih.gov/Traces/study/?acc=SRP032789
pheno <- read.table('SraRunTable_SRP032789.txt', sep = '\t',
                     header = TRUE,
                     stringsAsFactors = FALSE)

## obtain correct order for pheno data
pheno <- pheno[match(rse$run, pheno$Run_s), ]
identical(pheno$Run_s, rse$run)

## [1] TRUE
head(cbind(pheno$Run_s, rse$run))

##      [,1]      [,2]
## [1,] "SRR1027171" "SRR1027171"
## [2,] "SRR1027173" "SRR1027173"
## [3,] "SRR1027174" "SRR1027174"
## [4,] "SRR1027175" "SRR1027175"
## [5,] "SRR1027176" "SRR1027176"
## [6,] "SRR1027177" "SRR1027177"

## obtain grouping information
colData(rse)$group <- pheno$tumor_type_s
table(colData(rse)$group)

##
## HER2 Positive Breast Tumor      Non-TNBC Breast Tumor
##           5                      6
## Normal Breast Organoids       TNBC Breast Tumor
##           3                      6

## subset data to HER2 and TNBC types
rse <- rse[, rse$group %in% c('HER2 Positive Breast Tumor',
                             'TNBC Breast Tumor')]
rse

## class: RangedSummarizedExperiment
## dim: 226117 11
## metadata(0):
## assays(1): counts
## rownames(226117): 1 1 ... 9997 9997
## rowData names(0):
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(19): project sample ... bigwig_file group
rse.exon.filt <- rse

## obtain count matrix
counts <- assays(rse)$counts
dim(counts)

## [1] 226117      11

```

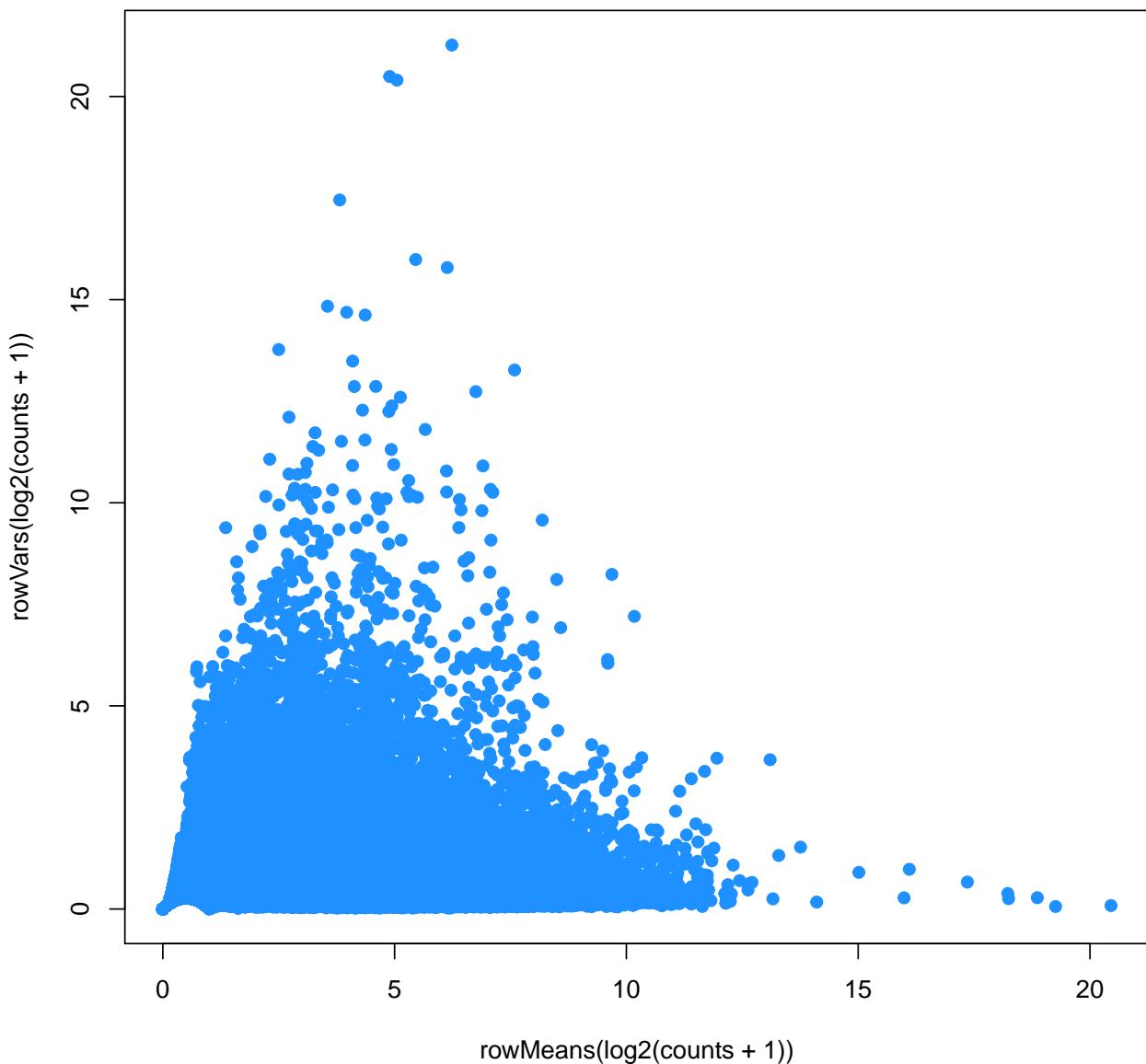
```
## filter count matrix (keep exons that are in filtered gene counts matrix)
filter <- rownames(counts) %in% rownames(counts_genes)
counts <- counts[filter, ]
dim(counts)

## [1] 204559      11
```

As above, to get a better sense of the data, we assess the mean-variance relationship for each exon. Similarly, we run principal component analysis (PCA) to identify any sample outliers within the data. We assess the variance explained by each of the first 11 PCs as well as visualize the relationship of each sample in the first two PCs.

```
## set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse$group))

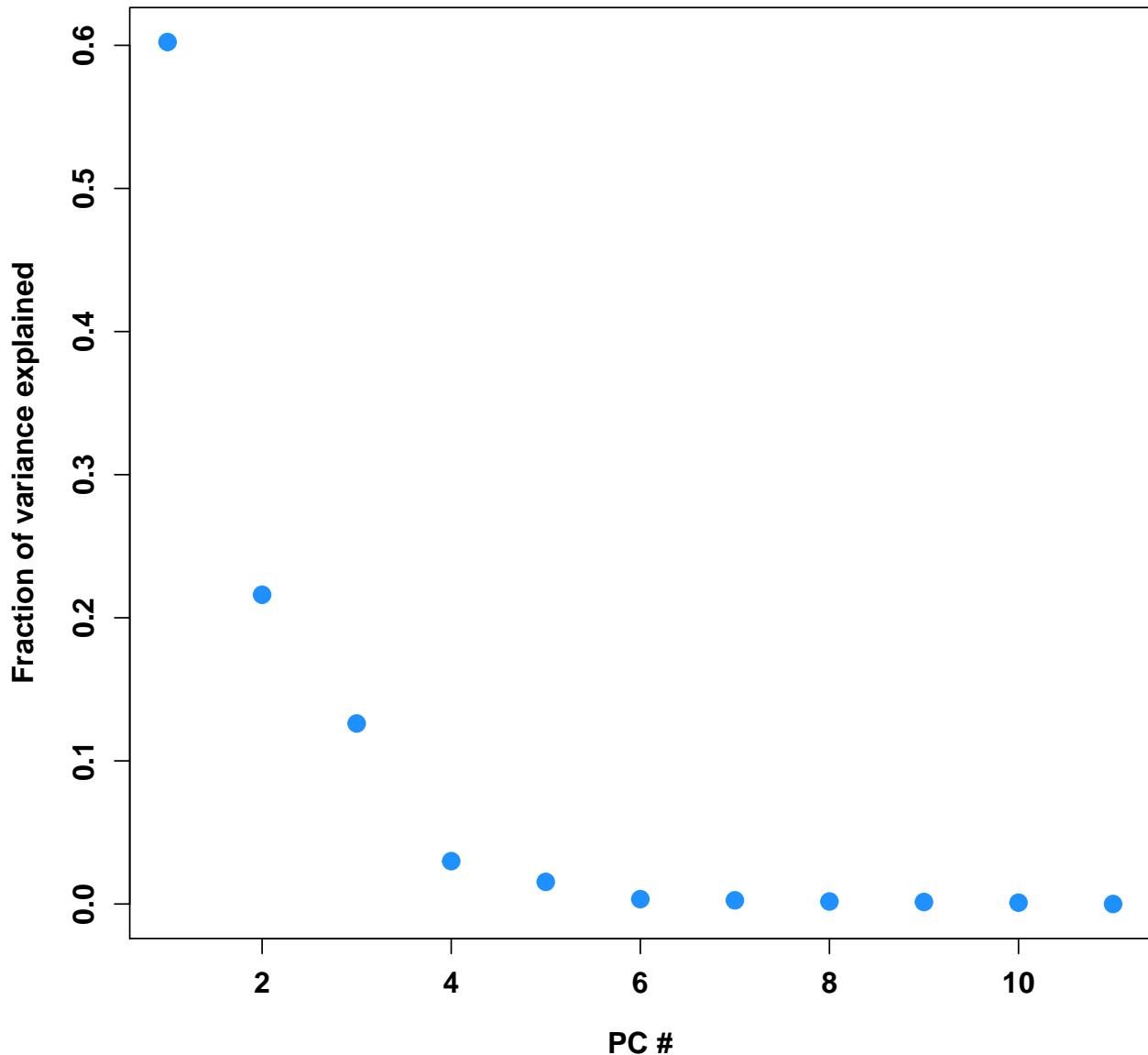
## Look at mean variance relationship
plot(rowMeans(log2(counts + 1)), rowVars(log2(counts + 1)),
      pch = 19, col = trop[2])
```



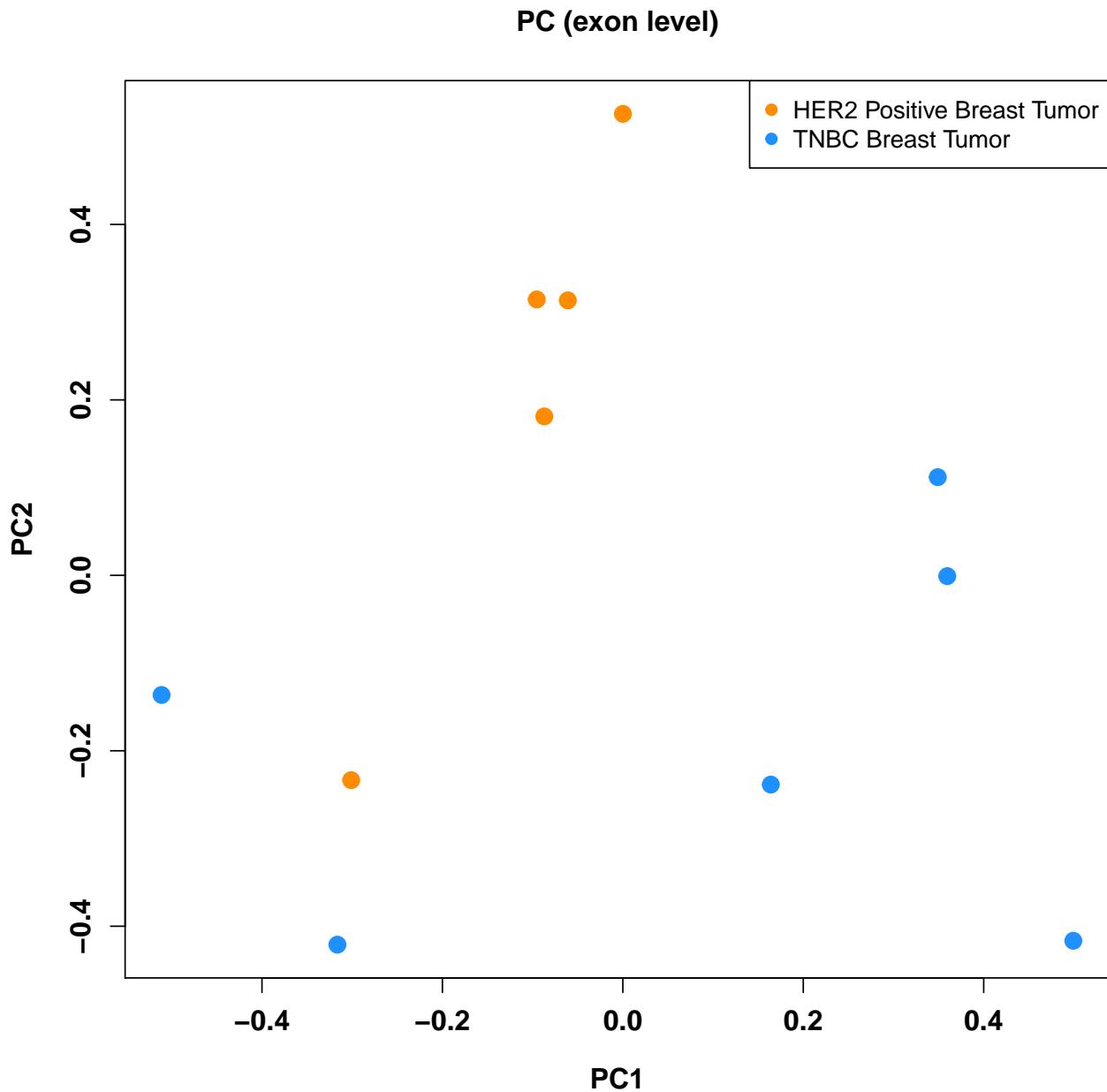
```
## calculate PCs with svd function
expr.pca <- svd(counts - rowMeans(counts))

## plot PCs
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(expr.pca$d^2 / sum(expr.pca$d^2), pch = 19, col = trop[2], cex = 1.5,
     ylab = 'Fraction of variance explained', xlab = 'PC #',
     main = 'PCs (exon level)')
```

### PCs (exon level)



```
##plot PC1 vs. PC2
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(expr.pca$v[, 1], expr.pca$v[, 2], pch = 19, col = trop[cols], cex = 1.5,
     xlab = 'PC1', ylab = 'PC2',
     main = 'PC (exon level)')
legend('topright', pch = 19, col = trop[c(1, 2)],
       names(summary(as.factor(rse$group))))
```



Again, differential expression analysis is carried out using limma and voom; however, this time at the exon, rather than gene, level. Data are again visualized using a volcano plot to assess the strength ( $\log_{10}FC$ ) and statistical significance ( $-\log_{10}p\text{-value}$ ) for each exon.

```
design <- model.matrix(~ rse$group)
design

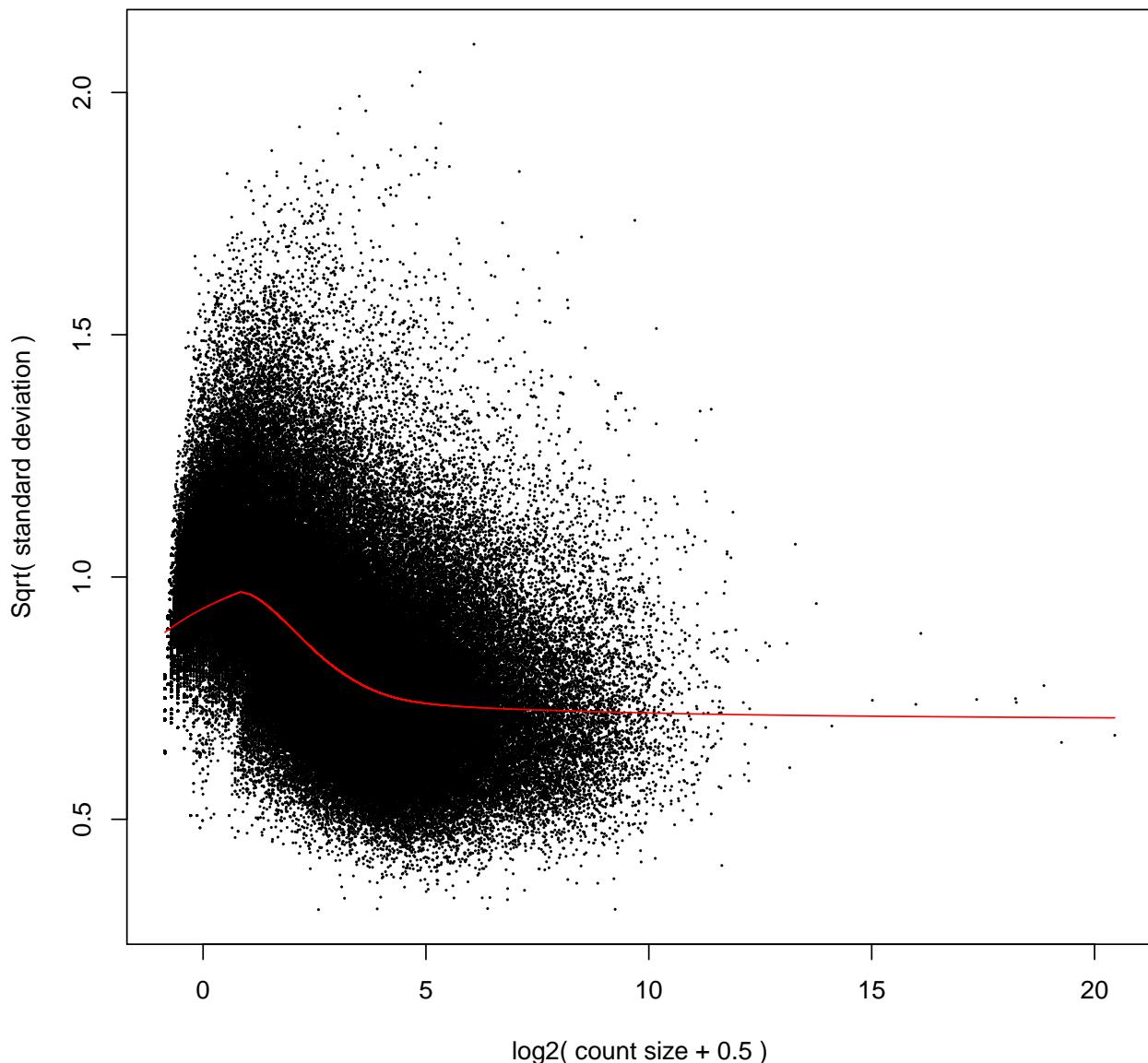
##      (Intercept) rse$groupTNBC Breast Tumor
## 1              1                      1
## 2              1                      1
## 3              1                      1
## 4              1                      1
## 5              1                      1
## 6              1                      0
## 7              1                      0
## 8              1                      0
```

```

## 9          1
## 10         1
## 11         1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`rse$group`
## [1] "contr.treatment"
dge <- DGEList(counts = counts)
dge <- calcNormFactors(dge)
v <- voom(dge, design, plot = TRUE)

```

### voom: Mean–variance trend



```

fit <- lmFit(v, design)
fit <- eBayes(fit)

```

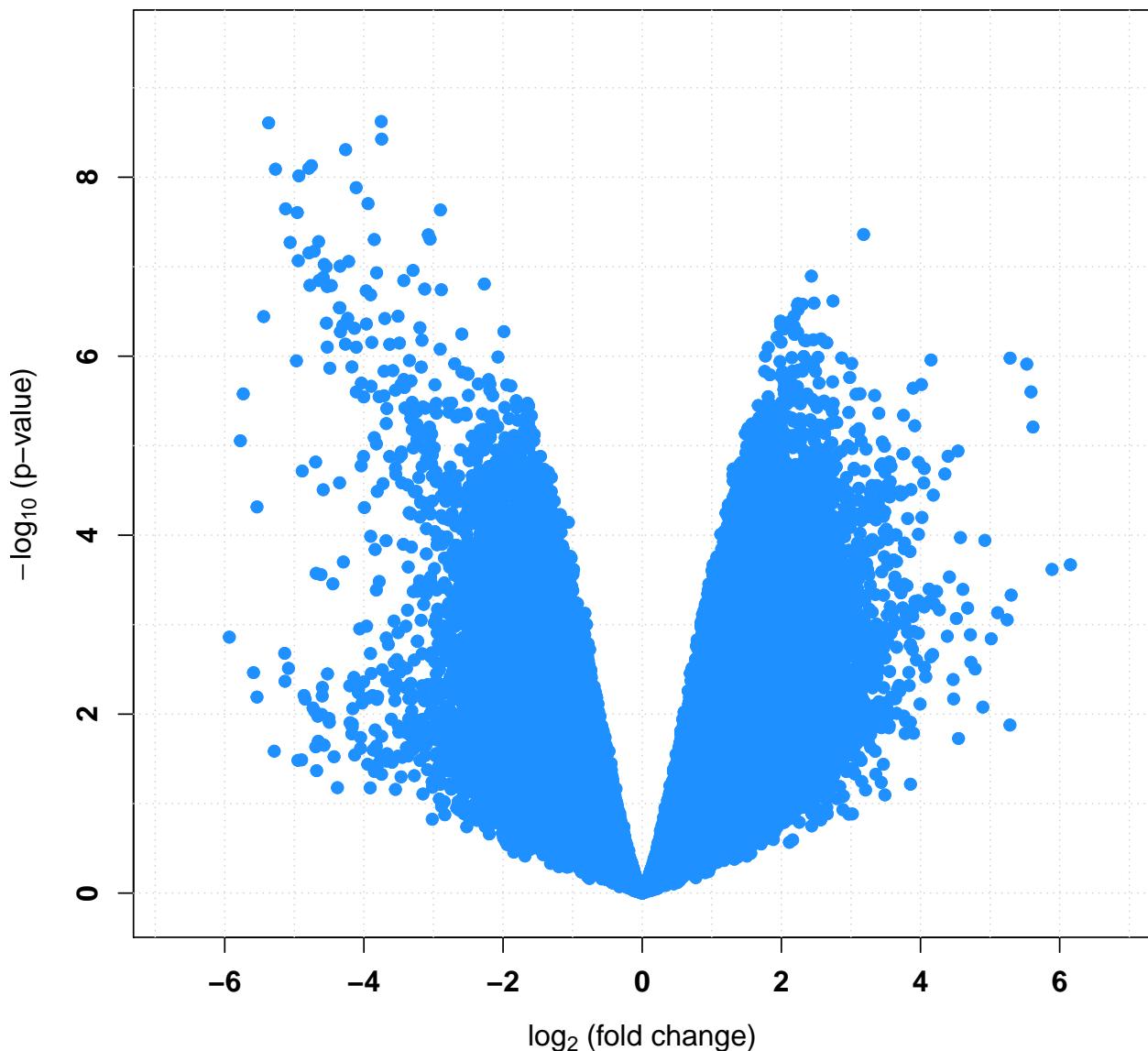
```

log2FC <- fit$coefficients[, 2]
p.mod <- fit$p.value[, 2]
q.mod <- qvalue(p.mod)$q
res.exons <- data.frame(log2FC, p.mod, q.mod)
#determine the number of exons differentially expressed q<0.05
sum(res.exons$q.mod<0.05)

## [1] 23647
## Volcano plot
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
rx2 <- c(-1, 1) * 1.1 * max(abs(log2FC))
ry2 <- c(-0.1, max(-log10(p.mod))) * 1.1
plot(log2FC, -log10(p.mod),
      pch = 19, xlim = rx2, ylim = ry2, col = trop[2],
      xlab = bquote(paste(log[2], ' (fold change)' )),
      ylab = bquote(paste(-log[10], ' (p-value)' )))
abline(v = seq(-10, 10, 1), col = 'lightgray', lty = 'dotted')
abline(h = seq(0, 23, 1), col = 'lightgray', lty = 'dotted')
points(log2FC, -log10(p.mod), pch = 19, col = trop[2])
title('Volcano plot: TNBC vs. HER2+ in SRP032789 (exon level)')

```

### Volcano plot: TNBC vs. HER2+ in SRP032789 (exon level)



To compare findings at the gene and exon level, we obtained a single exon-level p-value for each gene included at the gene-level analysis. To do this, we utilized Simes' rule [PMID: 23330866], such that for each gene included in the gene-level analysis, the p-values for exons within that gene were extracted and sorted. Each exon-level p-value is then multiplied by the number of exons present within the gene. For each exon (1,2...n), this quantity is divided by that exon's rank (where 1=most significant exon and n=least significant). The minimum value from this calculation is assigned as the exon-level p-value at each gene.

```
gene_id <- unique(rownames(counts))

## calculate p-values for gens with Simes' rule
p_gene <- NULL
for(i in seq_len(length(gene_id))){
  gene_id[i]
  p_exon <- res.exons$p.mod[rownames(counts) %in% gene_id[i]]
  p_exon <- sort(p_exon)
```

```

p_exon_simes <- NULL
for(j in 1:length(p_exon)){
  p_exon_simes[j] <- length(p_exon) * p_exon[j] / j
}
p_gene[i] <- min(p_exon_simes)
}

names(p_gene) <- gene_id
q_gene <- qvalue(p_gene)$q
## Determine the number of gene-level exons differentially expressed q<0.05
sum(q_gene<0.05)

## [1] 7935
## As above, 'topGO' can be utilized to assign biological function to
## differentially expressed exons.

## Gene set analysis (p-values of genes derived with Simes' rule from exon
## p-values)
interesting <- function(x) x < 0.05

topgoobjBP <- new('topGOdata',
  description = 'biological process',
  ontology = 'BP', allGenes = q_gene, geneSelectionFun = interesting,
  annotationFun = annFUN.org, mapping = 'org.Hs.eg.db', ID = 'entrez')

##
## Building most specific GOs .....
## ( 10647 GO terms found. )

##
## Build GO DAG topology .....
## ( 14443 GO terms and 34755 relations. )

##
## Annotating nodes .....
## ( 13762 genes annotated to the GO terms. )

bptest <- runTest(topgoobjBP, algorithm = 'weight01', statistic = 'ks')

##
##          -- Weight01 Algorithm --
##
##          the algorithm is scoring 14443 nontrivial nodes
##          parameters:
##              test statistic: ks
##              score order: increasing
##
##          Level 20:  1 nodes to be scored      (0 eliminated genes)
##
##          Level 19:  7 nodes to be scored      (0 eliminated genes)
##
##          Level 18: 20 nodes to be scored     (1 eliminated genes)

```

```

## 
##   Level 17: 40 nodes to be scored (30 eliminated genes)
## 
##   Level 16: 130 nodes to be scored (91 eliminated genes)
## 
##   Level 15: 266 nodes to be scored (179 eliminated genes)
## 
##   Level 14: 520 nodes to be scored (552 eliminated genes)
## 
##   Level 13: 925 nodes to be scored (1159 eliminated genes)
## 
##   Level 12: 1336 nodes to be scored (2451 eliminated genes)
## 
##   Level 11: 1602 nodes to be scored (4326 eliminated genes)
## 
##   Level 10: 1878 nodes to be scored (6122 eliminated genes)
## 
##   Level 9: 1967 nodes to be scored (8287 eliminated genes)
## 
##   Level 8: 1855 nodes to be scored (9989 eliminated genes)
## 
##   Level 7: 1641 nodes to be scored (11002 eliminated genes)
## 
##   Level 6: 1187 nodes to be scored (11986 eliminated genes)
## 
##   Level 5: 679 nodes to be scored (12493 eliminated genes)
## 
##   Level 4: 293 nodes to be scored (13053 eliminated genes)
## 
##   Level 3: 74 nodes to be scored (13234 eliminated genes)
## 
##   Level 2: 21 nodes to be scored (13397 eliminated genes)
## 
##   Level 1: 1 nodes to be scored (13475 eliminated genes)
bpptest

## 
## Description: biological process
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 14443 GO terms scored: 77 terms with p < 0.01
## Annotation data:
##     Annotated genes: 13762
##     Significant genes: 6194
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 14443

```

```

bpres_exon <- GenTable(topgoobjBP, pval = bptest,
  topNodes = length(bptest@score), numChar = 100)
head(bpres_exon, n = 10)

##          GO.ID                               Term
## 1  GO:0031124  mRNA 3'-end processing
## 2  GO:0051493 regulation of cytoskeleton organization
## 3  GO:0000398      mRNA splicing, via spliceosome
## 4  GO:0033120      positive regulation of RNA splicing
## 5  GO:0007049           cell cycle
## 6  GO:0006886      intracellular protein transport
## 7  GO:0006369      termination of RNA polymerase II transcription
## 8  GO:1903507 negative regulation of nucleic acid-templated transcription
## 9  GO:0008286      insulin receptor signaling pathway
## 10 GO:0048025      negative regulation of mRNA splicing, via spliceosome
##   Annotated Significant Expected    pval
## 1        80       61  36.01 5.3e-06
## 2       380      183 171.03 5.4e-06
## 3       264      175 118.82 1.9e-05
## 4        23       20  10.35 2.7e-05
## 5      1650      810 742.63 5.1e-05
## 6       937      482 421.72 0.00020
## 7        47       35  21.15 0.00025
## 8      1072      488 482.49 0.00045
## 9       302      156 135.92 0.00051
## 10       20       15   9.00 0.00060

```

To determine the concordance between the gene- and exon-level analyses, the top hits (as determined by p-value) are compared. Results are plotted such that the points falling along the identity line would indicate complete agreement between the top hits of each analysis.

## Concordance (genes)

```

## obtain and sort p-values for genes
p.mod1 <- res.genes$p.mod
names(p.mod1) <- rownames(res.genes)
p.mod1.sort <- p.mod1[order(p.mod1)]

## obtain dnd sort p-values for genes derived from exons
p.mod2 <- p_gene
p.mod2.sort <- p.mod2[order(p.mod2)]

## overlap for genes between studies
table(names(p.mod1.sort) %in% names(p.mod2.sort))

##
##  TRUE
## 17874
table(names(p.mod2.sort) %in% names(p.mod1.sort))

```

```

##  

##  TRUE  

## 17874  

conc <- NULL  

for(i in seq_len(length(p.mod1.sort))) {  

  conc[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod2.sort)[1:i])  

}  
  

par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5, cex.axis = 1.2)  

plot(seq(1:length(p.mod1.sort)), conc,  

  type = 'l', las = 0,  

  xlim = c(0, 18000),  

  ylim = c(0, 18000),  

  xlab = 'ordered genes',  

  ylab = 'ordered genes (from exon analysis)',  

  main = 'Concordance')  

for(k in 1:3){  

  abline(v = k * 5000, cex = 0.5, col = 'lightgrey')  

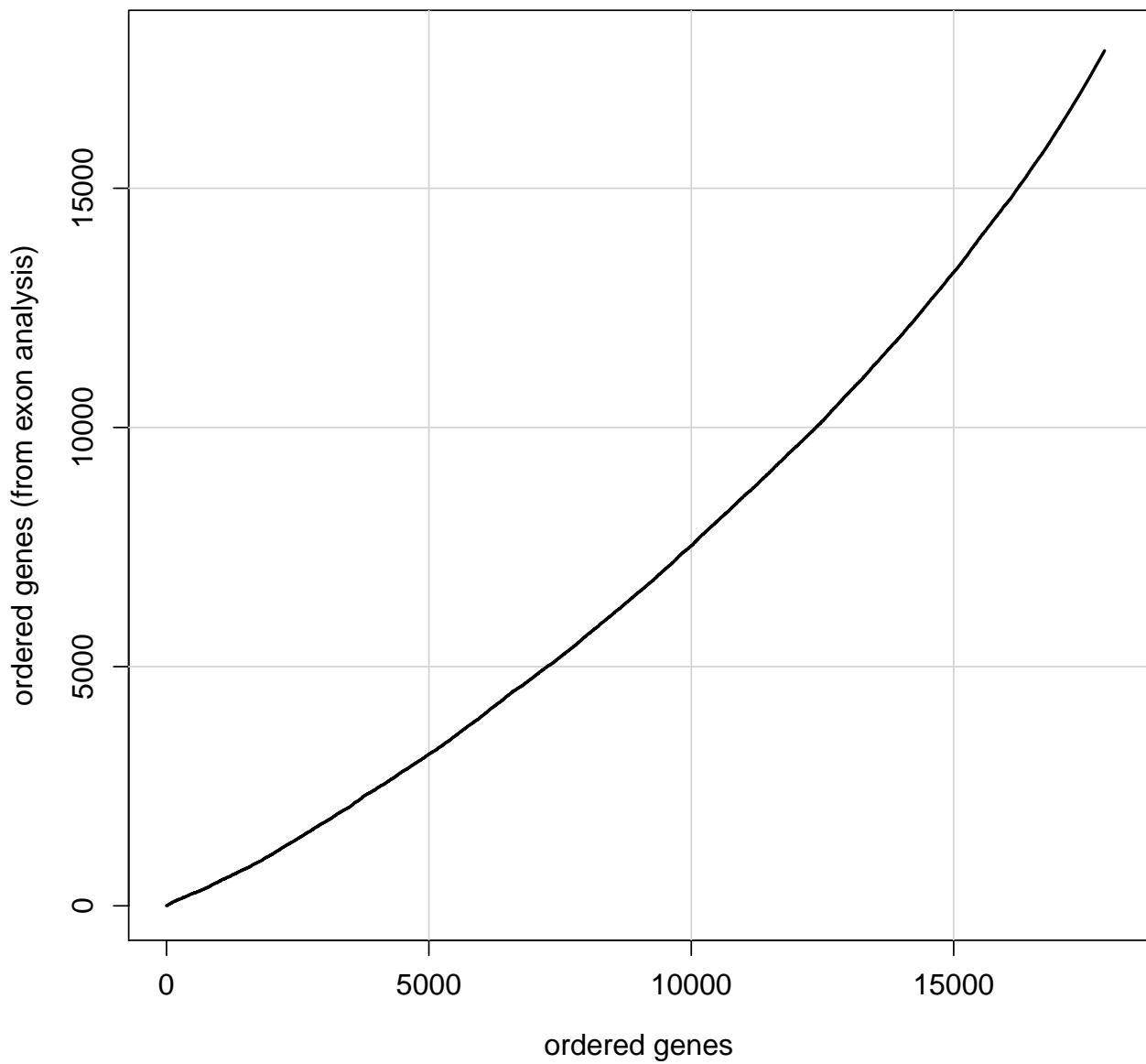
  abline(h = k * 5000, cex = 0.5, col = 'lightgrey')  

}  

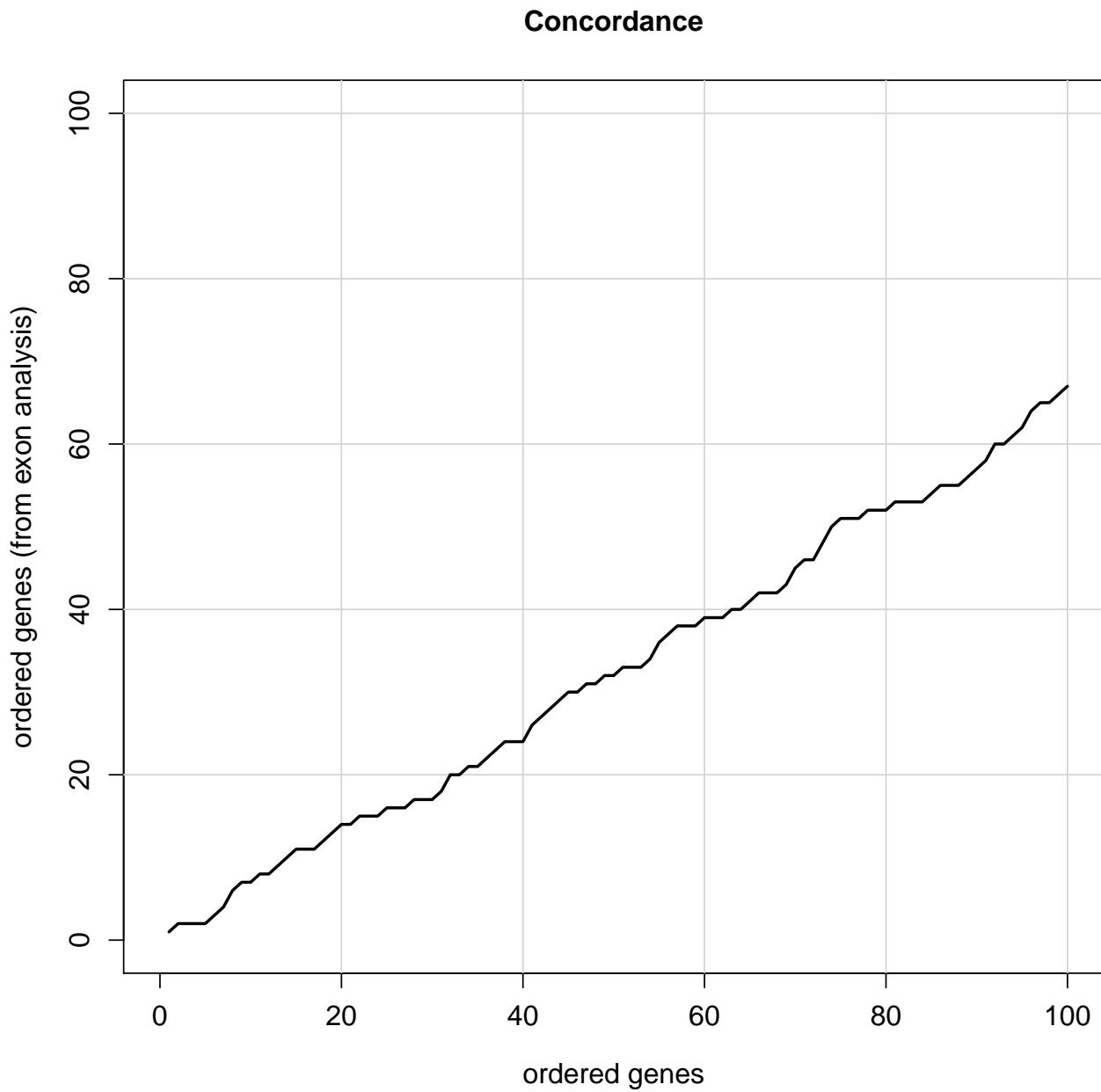
lines(seq(1:length(p.mod1.sort)), conc, col = 'black', lwd = 2)

```

## Concordance



```
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5,
  cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort[1:100])), conc[1:100],
  type = 'l',
  xlim = c(0, 100),
  ylim = c(0, 100),
  xlab = 'ordered genes',
  ylab = 'ordered genes (from exon analysis)',
  main = 'Concordance')
for(k in 1:5){
  abline(v = k * 20, cex = 0.5, col = 'lightgrey')
  abline(h = k * 20, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod1.sort[1:100])), conc[1:100], col = 'black', lwd = 2)
```



Concordance can also be calculated looking at the gene ontology (GO) groups identified from the gene and exon-level analyses. Again, we plot the agreement between the two analyses such that complete agreement between the two analyses would fall along the identity line.

## Concordance (GO groups)

```
## obtain and sort p-values for genes
p.mod1 <- bpres_gene$GO.ID
names(p.mod1) <- bpres_gene$GO.ID
p.mod1.sort <- p.mod1

## obtain and sort p-values for genes derived from exons
```

```

p.mod2 <- bpres_exon$GO.ID
names(p.mod2) <- bpres_exon$GO.ID
p.mod2.sort <- p.mod2

## overlap for genes between studies
table(names(p.mod1.sort) %in% names(p.mod2.sort))

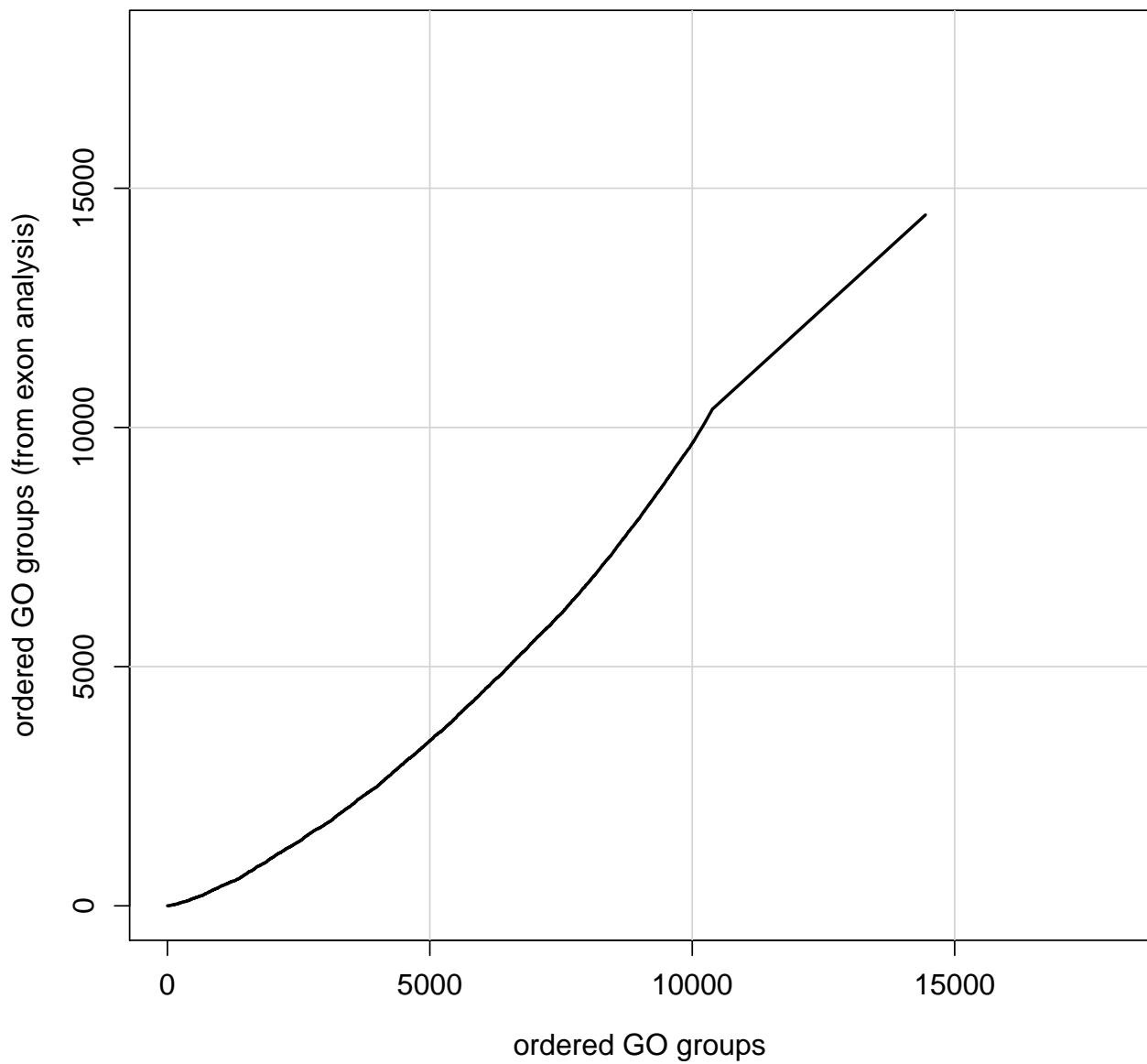
## TRUE
## 14443
table(names(p.mod2.sort) %in% names(p.mod1.sort))

## TRUE
## 14443
conc <- NULL
for(i in 1:length(p.mod1.sort)){
  conc[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod2.sort)[1:i])
}

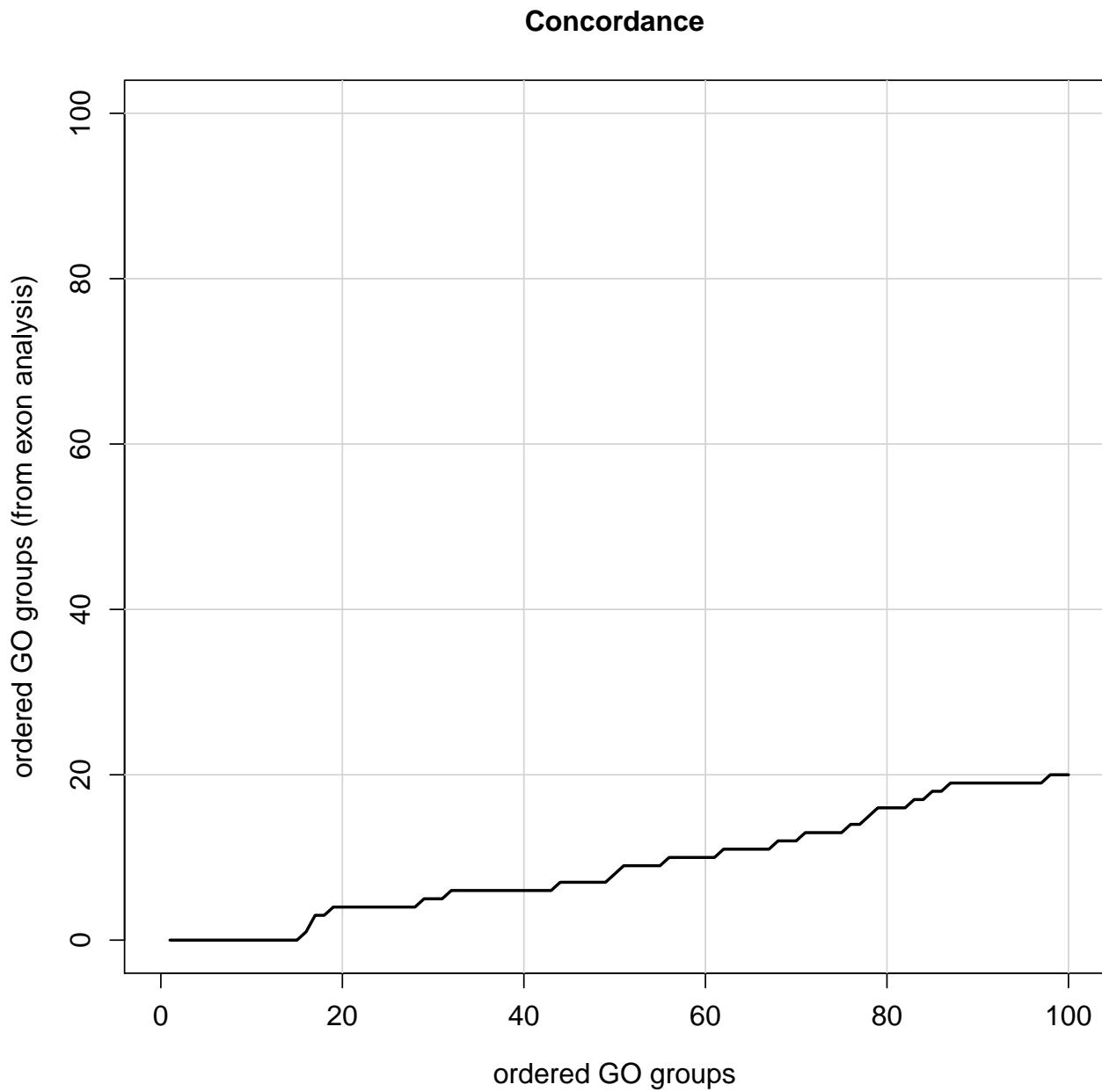
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5, cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort)), conc,
  type = 'l', las = 0,
  xlim = c(0, 18000),
  ylim = c(0, 18000),
  xlab = 'ordered GO groups',
  ylab = 'ordered GO groups (from exon analysis)',
  main = 'Concordance')
for(k in 1:3){
  abline(v = k * 5000, cex = 0.5, col = 'lightgrey')
  abline(h = k * 5000, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod1.sort)), conc, col = 'black', lwd = 2)

```

## Concordance



```
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5,
  cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort[1:100])), conc[1:100],
  type = 'l',
  xlim = c(0, 100),
  ylim = c(0, 100),
  xlab = 'ordered GO groups',
  ylab = 'ordered GO groups (from exon analysis)',
  main = 'Concordance')
for(k in 1:5){
  abline(v = k * 20, cex = 0.5, col = 'lightgrey')
  abline(h = k * 20, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod1.sort[1:100])), conc[1:100], col = 'black', lwd = 2)
```



## Reproducibility

This analysis report was made possible thanks to:

- R (R Core Team, 2016)
- *BiocStyle* (Oleś, Morgan, and Huber, 2016)
- *derfinder* (Collado-Torres, Nellore, Frazee, Wilks, et al., 2016)
- *devtools* (Wickham and Chang, 2016)
- *edgeR* (Robinson, McCarthy, and Smyth, 2010)
- *knitcitations* (Boettiger, 2015)
- *matrixStats* (Bengtsson, 2016)
- *qvalue* (with contributions from Andrew J. Bass, Dabney, and Robinson, 2015)
- *recount* (Collado-Torres and Leek, 2016)

- *rmarkdown* (Allaire, Cheng, Xie, McPherson, et al., 2016)
- *RSkittleBrewer* (Frazee, 2016)
- *SummarizedExperiment* (Morgan, Obenchain, Hester, and Pagès, 2016)
- *topGO* (Alexa and Rahnenfuhrer, 2016)
- *limma* (Law, Chen, Shi, and Smyth, 2014)

## Bibliography file

- [1] A. Alexa and J. Rahnenfuhrer. topGO: Enrichment Analysis for Gene Ontology. R package version 2.25.0. 2016.
- [2] J. Allaire, J. Cheng, Y. Xie, J. McPherson, et al. rmarkdown: Dynamic Documents for R. R package version 0.9.6. 2016. URL: <https://CRAN.R-project.org/package=rmarkdown>.
- [3] J. D. S. with contributions from Andrew J. Bass, A. Dabney and D. Robinson. qvalue: Q-value estimation for false discovery rate control. R package version 2.5.2. 2015. URL: <http://github.com/jdstorey/qvalue>.
- [4] H. Bengtsson. matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors). R package version 0.50.2. 2016. URL: <https://CRAN.R-project.org/package=matrixStats>.
- [5] C. Boettiger. knitcitations: Citations for ‘Knitr’ Markdown Files. R package version 1.0.7. 2015. URL: <https://CRAN.R-project.org/package=knitcitations>.
- [6] L. Collado-Torres and J. T. Leek. recount: Explore and download data from the recount project. R package version 0.99.10. 2016. URL: <https://github.com/leekgroup/recount>.
- [7] L. Collado-Torres, A. Nellore, A. C. Frazee, C. Wilks, et al. “Flexible expressed region analysis for RNA-seq with derfinder”. In: bioRxiv (2016). DOI: 10.1101/015370. URL: <http://biorkxiv.org/content/early/2016/05/07/015370>.
- [8] A. Frazee. RSkittleBrewer: Fun with R Colors. R package version 1.1. 2016. URL: <https://github.com/alyssafrazee/RSkittleBrewer>.
- [9] C. Law, Y. Chen, W. Shi and G. Smyth. “Voom: precision weights unlock linear model analysis tools for RNA-seq read counts”. In: Genome Biology 15 (2014), p. R29.
- [10] M. Morgan, V. Obenchain, J. Hester and H. Pagès. SummarizedExperiment: SummarizedExperiment container. R package version 1.3.4. 2016.
- [11] A. Oleś, M. Morgan and W. Huber. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.1.6. 2016. URL: <https://github.com/Bioconductor/BiocStyle>.
- [12] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: <https://www.R-project.org/>.
- [13] M. D. Robinson, D. J. McCarthy and G. K. Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”. In: Bioinformatics 26 (2010), pp. -1.
- [14] H. Wickham and W. Chang. devtools: Tools to Make Developing R Packages Easier. R package version 1.11.1. 2016. URL: <https://CRAN.R-project.org/package=devtools>.
- ```
## Time spent creating this report:
diff(c(timestart, Sys.time()))

## Time difference of 15.434 mins
## Date this report was generated
message(Sys.time())

## 2016-06-14 11:42:03
## Reproducibility info
options(width = 120)
devtools::session_info()

## Session info -----
```

```

## setting value
## version R version 3.3.0 RC (2016-05-01 r70572)
## system x86_64, darwin13.4.0
## ui X11
## language (EN)
## collate en_US.UTF-8
## tz America/New_York
## date 2016-06-14

## Packages -----
## package          * version  date      source
## acepack           1.3-3.3  2014-11-24 CRAN (R 3.3.0)
## AnnotationDbi    * 1.35.3   2016-05-27 Bioconductor
## bibtex            0.4.0    2014-12-31 CRAN (R 3.3.0)
## Biobase           * 2.33.0   2016-05-05 Bioconductor
## BiocGenerics     * 0.19.1   2016-06-11 Bioconductor
## BiocParallel      1.7.2    2016-05-20 Bioconductor
## BiocStyle          * 2.1.6    2016-06-11 Bioconductor
## biomaRt            2.29.2   2016-05-30 Bioconductor
## Biostrings         2.41.2    2016-06-08 Bioconductor
## bitops              1.0-6    2013-08-17 CRAN (R 3.3.0)
## BSgenome            1.41.0   2016-05-05 Bioconductor
## bumphunter         1.13.0   2016-05-05 Bioconductor
## chron                2.3-47   2015-06-24 CRAN (R 3.3.0)
## cluster              2.0.4    2016-04-18 CRAN (R 3.3.0)
## codetools             0.2-14   2015-07-15 CRAN (R 3.3.0)
## colorout             * 1.1-2    2016-05-05 Github (jalvesaq/colorout@6538970)
## colorspace            1.2-6    2015-03-11 CRAN (R 3.3.0)
## data.table            1.9.6    2015-09-19 CRAN (R 3.3.0)
## DBI                  0.4-1    2016-05-08 CRAN (R 3.3.0)
## derfinder             * 1.7.8    2016-06-08 Bioconductor
## derfinderHelper       1.7.3    2016-05-20 Bioconductor
## devtools              1.11.1   2016-04-21 CRAN (R 3.3.0)
## digest                 0.6.9    2016-01-08 CRAN (R 3.3.0)
## doRNG                  1.6     2014-03-07 CRAN (R 3.3.0)
## edgeR                  * 3.15.0   2016-05-27 Bioconductor
## evaluate                 0.9     2016-04-29 CRAN (R 3.3.0)
## foreach                 1.4.3    2015-10-13 CRAN (R 3.3.0)
## foreign                 0.8-66   2015-08-19 CRAN (R 3.3.0)
## formatR                  1.4     2016-05-09 CRAN (R 3.3.0)
## Formula                 1.2-1    2015-04-07 CRAN (R 3.3.0)
## GenomeInfoDb             * 1.9.1    2016-05-13 Bioconductor
## GenomicAlignments        1.9.2    2016-06-13 Bioconductor
## GenomicFeatures          1.25.12   2016-05-21 Bioconductor
## GenomicFiles              1.9.11   2016-06-03 Bioconductor
## GenomicRanges             * 1.25.4   2016-06-10 Bioconductor
## ggplot2                  2.1.0    2016-03-01 CRAN (R 3.3.0)
## GO.db                   * 3.3.0    2016-05-05 Bioconductor
## graph                   * 1.51.0   2016-05-05 Bioconductor
## gridExtra                 2.2.1    2016-02-29 CRAN (R 3.3.0)
## gtable                   0.2.0    2016-02-26 CRAN (R 3.3.0)
## Hmisc                     3.17-4   2016-05-02 CRAN (R 3.3.0)
## htmltools                  0.3.5    2016-03-21 CRAN (R 3.3.0)
## httr                      1.1.0    2016-01-28 CRAN (R 3.3.0)

```

```

## IRanges * 2.7.6 2016-06-10 Bioconductor
## iterators 1.0.8 2015-10-13 CRAN (R 3.3.0)
## knitr * 1.0.7 2015-10-28 CRAN (R 3.3.0)
## lattice 1.13 2016-05-09 CRAN (R 3.3.0)
## latticeExtra 0.20-33 2015-07-14 CRAN (R 3.3.0)
## limma * 3.29.7 2016-06-13 Bioconductor
## locfit 1.5-9.1 2013-04-20 CRAN (R 3.3.0)
## lubridate 1.5.6 2016-04-06 CRAN (R 3.3.0)
## magrittr 1.5 2014-11-22 CRAN (R 3.3.0)
## Matrix 1.2-6 2016-05-02 CRAN (R 3.3.0)
## matrixStats * 0.50.2 2016-04-24 CRAN (R 3.3.0)
## memoise 1.0.0 2016-01-29 CRAN (R 3.3.0)
## munsell 0.4.3 2016-02-13 CRAN (R 3.3.0)
## nnet 7.3-12 2016-02-02 CRAN (R 3.3.0)
## org.Hs.eg.db * 3.3.0 2016-05-05 Bioconductor
## pkgmaker 0.22 2014-05-14 CRAN (R 3.3.0)
## plyr 1.8.3 2015-06-12 CRAN (R 3.3.0)
## qvalue * 2.5.2 2016-05-20 Bioconductor
## R6 2.1.2 2016-01-26 CRAN (R 3.3.0)
## RColorBrewer 1.1-2 2014-12-07 CRAN (R 3.3.0)
## Rcpp 0.12.5 2016-05-14 CRAN (R 3.3.0)
## RCurl 1.95-4.8 2016-03-01 CRAN (R 3.3.0)
## recount * 0.99.10 2016-06-12 Github (leekgroup/recount@7a7ea73)
## RefManageR 0.10.13 2016-04-04 CRAN (R 3.3.0)
## registry 0.3 2015-07-08 CRAN (R 3.3.0)
## reshape2 1.4.1 2014-12-06 CRAN (R 3.3.0)
## RJSONIO 1.3-0 2014-07-28 CRAN (R 3.3.0)
## rmarkdown * 0.9.6 2016-05-01 CRAN (R 3.3.0)
## rngtools 1.2.4 2014-03-06 CRAN (R 3.3.0)
## rpart 4.1-10 2015-06-29 CRAN (R 3.3.0)
## Rsamtools 1.25.0 2016-05-05 Bioconductor
## RSkittleBrewer * 1.1 2016-06-13 Github (alyssafrazee/RSkittleBrewer@230d1d0)
## RSQLite 1.0.0 2014-10-25 CRAN (R 3.3.0)
## rstudioapi 0.5 2016-01-24 CRAN (R 3.3.0)
## rtracklayer 1.33.5 2016-06-13 Bioconductor
## S4Vectors * 0.11.4 2016-06-11 Bioconductor
## scales 0.4.0 2016-02-26 CRAN (R 3.3.0)
## SparseM * 1.7 2015-08-15 CRAN (R 3.3.0)
## stringi 1.0-1 2015-10-22 CRAN (R 3.3.0)
## stringr 1.0.0 2015-04-30 CRAN (R 3.3.0)
## SummarizedExperiment * 1.3.4 2016-06-10 Bioconductor
## survival 2.39-4 2016-05-11 CRAN (R 3.3.0)
## topGO * 2.25.0 2016-05-05 Bioconductor
## VariantAnnotation 1.19.2 2016-06-07 Bioconductor
## withr 1.0.1 2016-02-04 CRAN (R 3.3.0)
## XML 3.98-1.4 2016-03-01 CRAN (R 3.3.0)
## xtable 1.8-2 2016-02-05 CRAN (R 3.3.0)
## XVector 0.13.0 2016-05-05 Bioconductor
## yaml 2.1.13 2014-06-12 CRAN (R 3.3.0)
## zlibbioc 1.19.0 2016-05-05 Bioconductor

```