recount (gene and exon analyses)

Kai Kammers and Shannon Ellis

July 22, 2016

Contents

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 TNBC samples (triple negative breast cancer) and HER2-positive breast cancer samples (breast cancer type that tests positive for a protein called human epidermal growth factor receptor 2). Code here demonstrates how to carry out differential expression analyses on gene, exon, junction, and differential expressed region (DER) levels 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')
```

1 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
##
## 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 gene level RangedSummarizedExperiment data
if(!file.exists(file.path('SRP032789', 'rse_gene.Rdata'))) {
    download_study(project_info$project)
}

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

```
## class: RangedSummarizedExperiment
## dim: 58037 20
## metadata(0):
## assays(1): counts
## rownames(58037): ENSG00000000003.14 ENSG00000000005.5 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(3): gene_id bp_length symbol
## colnames(20): SRR1027171 SRR1027173 ... SRR1027190 SRR1027172
## colData names(21): project sample ... title characteristics
```

```
## This is the phenotype data provided by the recount project
colData(rse_gene)
```

```
## DataFrame with 20 rows and 21 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_downloaded
##                                <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_downloaded 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
##            sharq_beta_tissue sharq_beta_cell_type
##                  <character>          <character>
## SRR1027171            breast                  esc
## SRR1027173            breast                  esc
## SRR1027174            breast                  esc
## SRR1027175            breast                  esc
## SRR1027176            breast                  esc
## ...                      ...                  ...
## SRR1027187            breast                  esc
## SRR1027188            breast                  esc
## SRR1027189            breast                  esc
## SRR1027190            breast                  esc
## SRR1027172            breast                  esc
##            biosample_submission_date biosample_publication_date
##                          <character>                <character>
## SRR1027171   2013-11-07T12:40:22.203    2013-11-08T01:11:17.160
## SRR1027173   2013-11-07T12:40:32.283    2013-11-08T01:11:14.827
## SRR1027174   2013-11-07T12:40:28.283    2013-11-08T01:11:52.283
## SRR1027175   2013-11-07T12:40:34.343    2013-11-08T01:11:15.963
## SRR1027176   2013-11-07T12:40:36.303    2013-11-08T01:11:46.430
## ...                             ...                        ...
## SRR1027187   2013-11-07T12:40:56.180    2013-11-08T01:11:29.587
## SRR1027188   2013-11-07T12:40:58.170    2013-11-08T01:12:06.660
## SRR1027189   2013-11-07T12:40:20.227    2013-11-08T01:11:33.080
## SRR1027190   2013-11-07T12:40:18.090    2013-11-08T01:12:11.320
```

```
## SRR1027172   2013-11-07T12:40:26.217    2013-11-08T01:11:45.250
##              biosample_update_date avg_read_length geo_accession
##                        <character>       <integer>   <character>
## SRR1027171 2014-03-07T16:09:38.542             120    GSM1261016
## SRR1027173 2014-03-07T16:09:38.698             120    GSM1261018
## SRR1027174 2014-03-07T16:09:38.637             120    GSM1261019
## SRR1027175 2014-03-07T16:09:38.731             120    GSM1261020
## SRR1027176 2014-03-07T16:09:38.768             120    GSM1261021
## ...                            ...             ...           ...
## SRR1027187 2014-03-07T16:09:39.093             120    GSM1261032
## SRR1027188 2014-03-07T16:09:39.130             150    GSM1261033
## SRR1027189 2014-03-07T16:09:38.498             150    GSM1261034
## SRR1027190 2014-03-07T16:09:38.469             150    GSM1261035
## SRR1027172 2014-03-07T16:09:38.604              87    GSM1261017
##              bigwig_file       title
##              <character> <character>
## SRR1027171 SRR1027171.bw       TNBC1
## SRR1027173 SRR1027173.bw       TNBC3
## SRR1027174 SRR1027174.bw       TNBC4
## SRR1027175 SRR1027175.bw       TNBC5
## SRR1027176 SRR1027176.bw       TNBC6
## ...                   ...         ...
## SRR1027187 SRR1027187.bw      HER2-5
## SRR1027188 SRR1027188.bw        NBS1
## SRR1027189 SRR1027189.bw        NBS2
## SRR1027190 SRR1027190.bw        NBS3
## SRR1027172 SRR1027172.bw       TNBC2
##                                   characteristics
##                                   <CharacterList>
## SRR1027171          tumor type: TNBC Breast Tumor
## SRR1027173          tumor type: TNBC Breast Tumor
## SRR1027174          tumor type: TNBC Breast Tumor
## SRR1027175          tumor type: TNBC Breast Tumor
## SRR1027176          tumor type: TNBC Breast Tumor
## ...                                           ...
## SRR1027187 tumor type: HER2 Positive Breast Tumor
## SRR1027188    tumor type: Normal Breast Organoids
## SRR1027189    tumor type: Normal Breast Organoids
## SRR1027190    tumor type: Normal Breast Organoids
## SRR1027172          tumor type: TNBC Breast Tumor
```

```
## 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 58037 rows and 3 columns
##                  gene_id bp_length          symbol
##              <character> <integer> <CharacterList>
## 1     ENSG00000000003.14      4535          TSPAN6
## 2      ENSG00000000005.5      1610            TNMD
## 3     ENSG00000000419.12      1207            DPM1
## 4     ENSG00000000457.13      6883           SCYL3
## 5     ENSG00000000460.16      5967        C1orf112
## ...                  ...       ...             ...
```

```
## 58033  ENSG00000283695.1        61              NA
## 58034  ENSG00000283696.1       997              NA
## 58035  ENSG00000283697.1      1184    LOC101928917
## 58036  ENSG00000283698.1       940              NA
## 58037  ENSG00000283699.1        60         MIR4481
```

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')]

## Save filtered rse object
rse_gene_filt <- rse

## Obtain count matrix
counts <- assays(rse_gene_filt)$counts
```

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

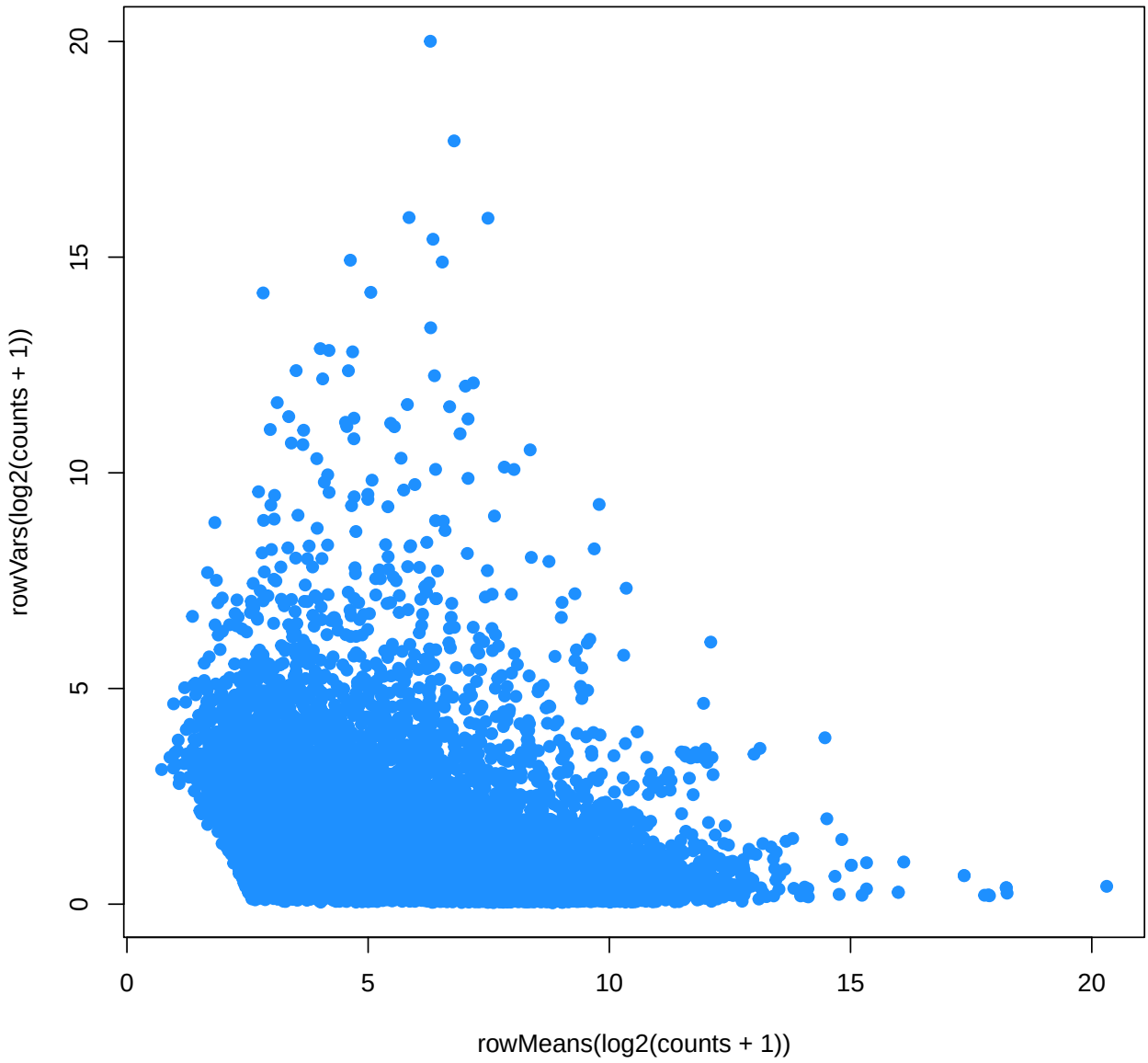
```
## [1] 26742    11
```

```
## Save for gene, exon and junction comparisons
counts_gene <- counts
counts_gene[1:5, 1:5]
```

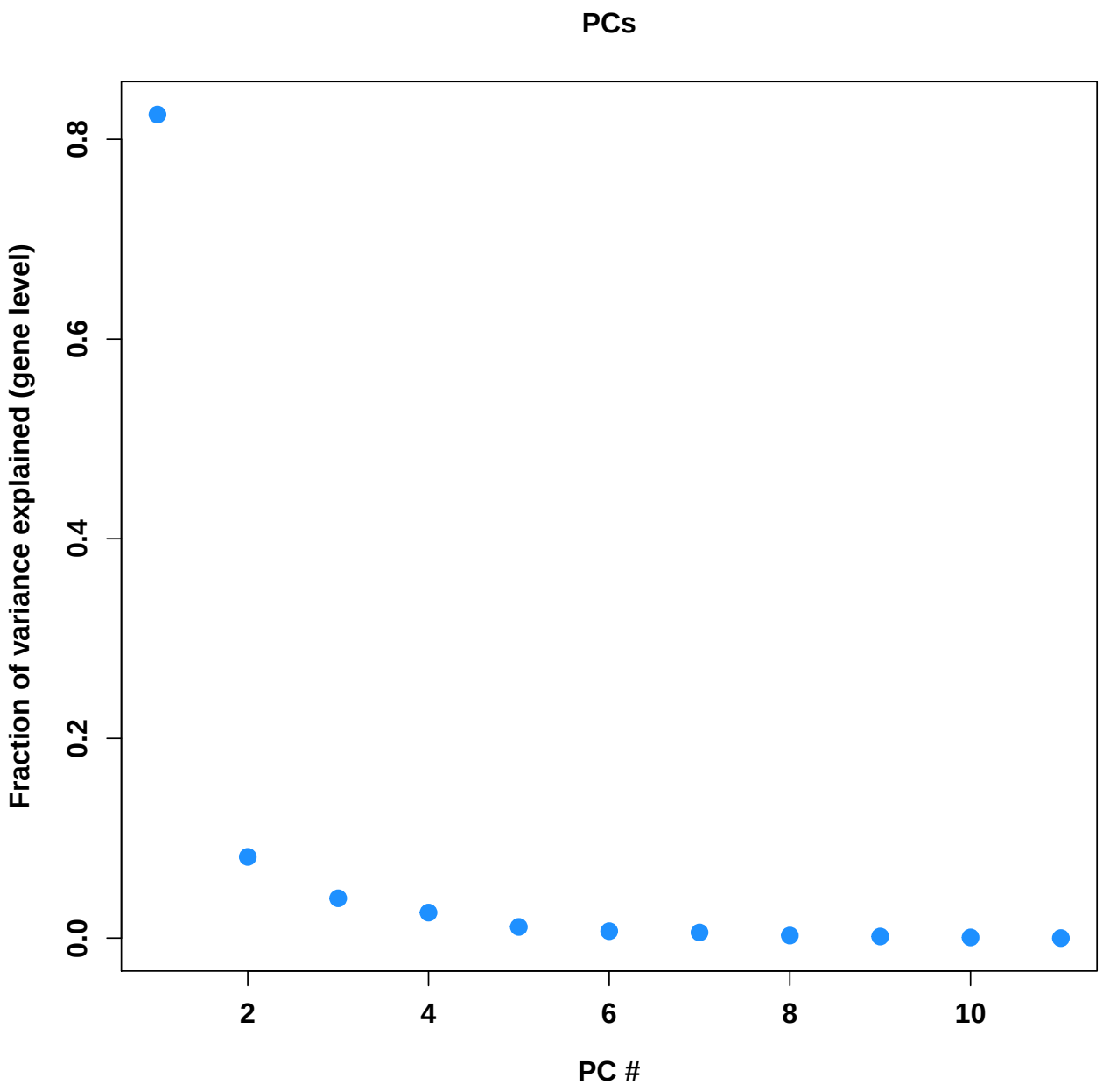
```
##                     SRR1027171 SRR1027173 SRR1027174 SRR1027175 SRR1027176
## ENSG00000000003.14         293        242        321        513        181
## ENSG00000000005.5            6          6          3          2        119
## ENSG00000000419.12         583        303        336        546        391
## ENSG00000000457.13         418        334        197        343        290
## ENSG00000000460.16         381        205        151        209        264
```

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.

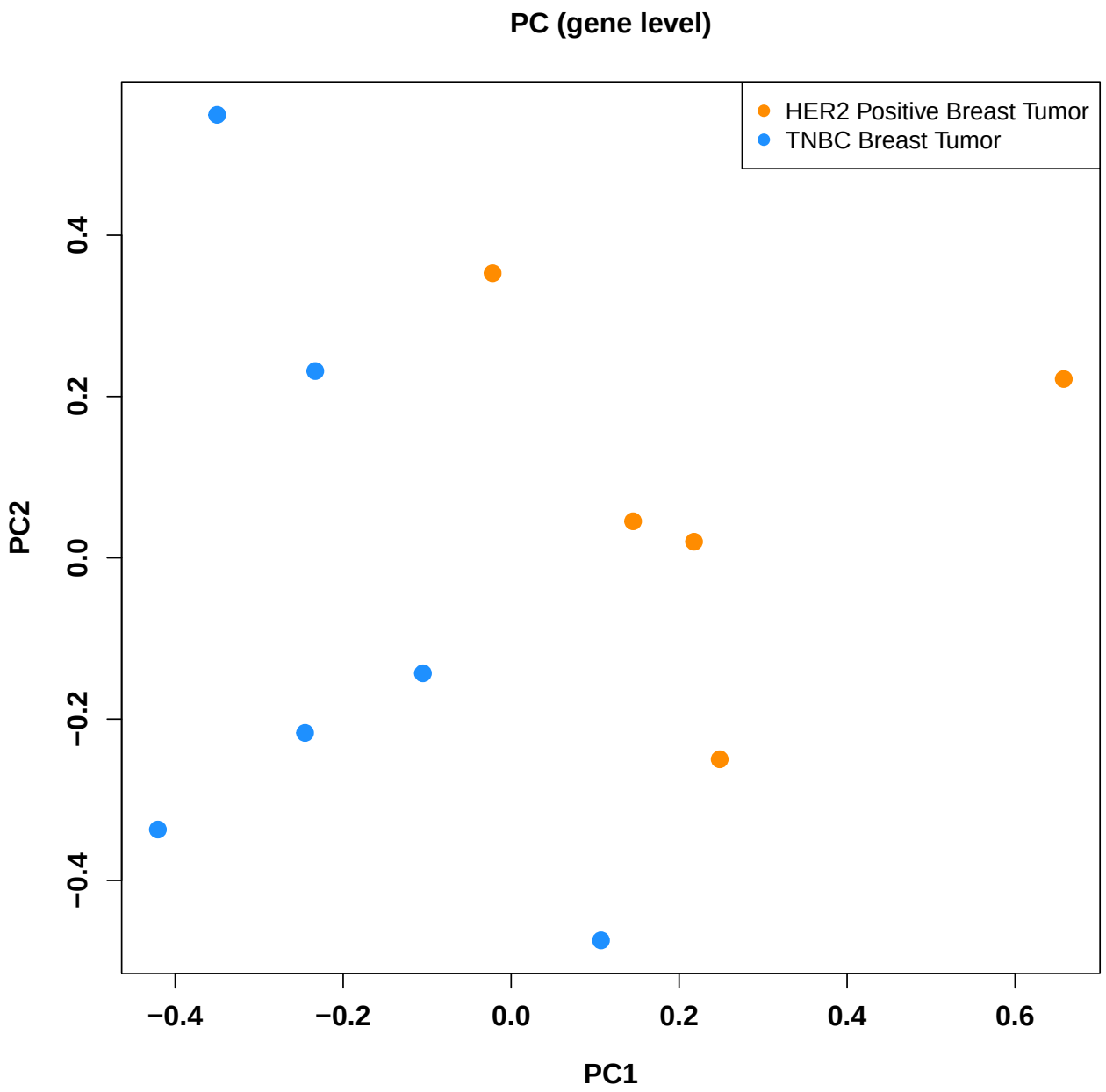
```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse$group))

## 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))), bg="white")
```



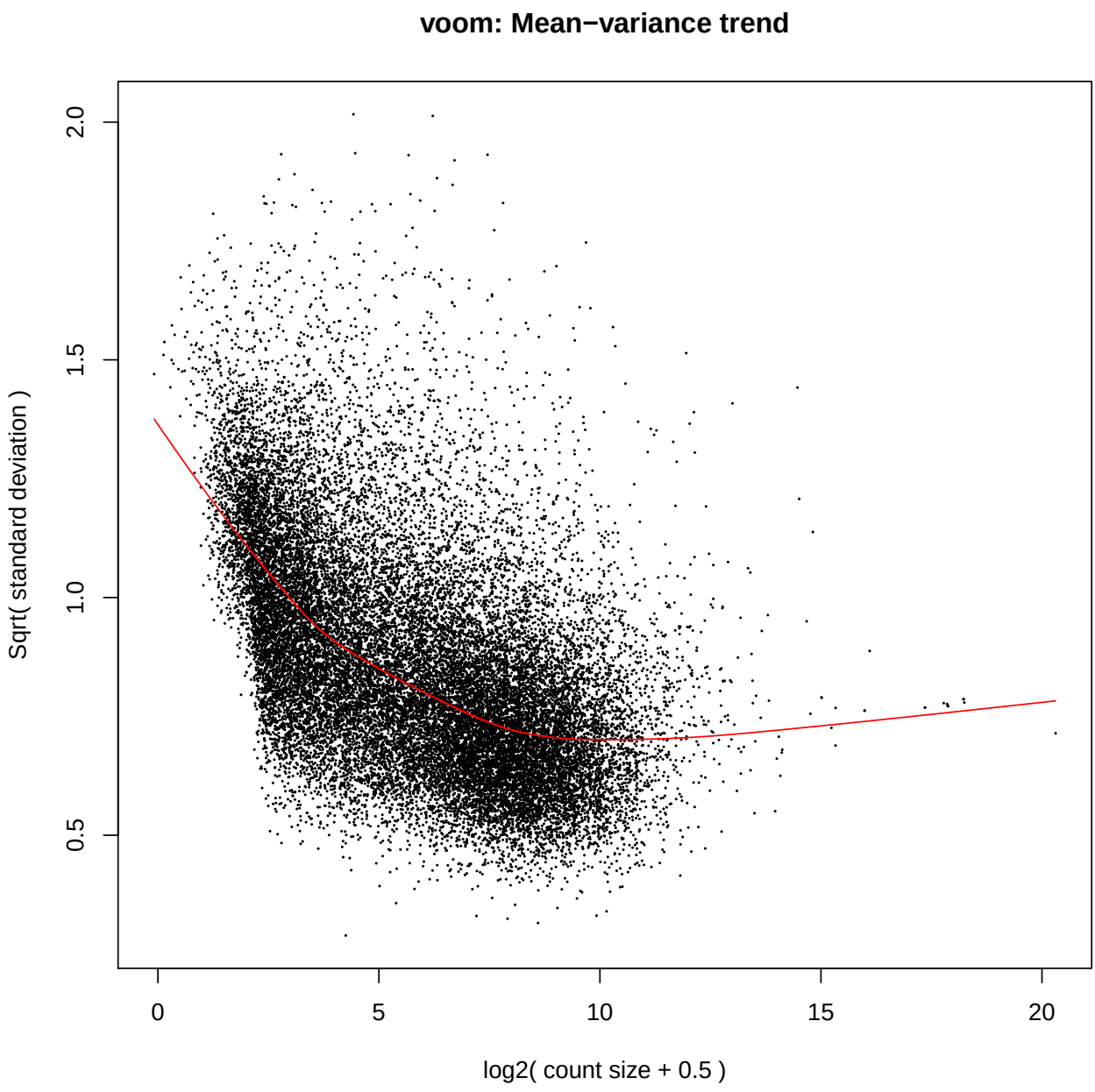
Having determined there are no sample outliers in these data, we carry out differential gene expression analysis. Differential gene expression between TNBC and HER2-positive samples are determined using `limma` and `voom`. Differentially expressed genes are visualized using a volcano plot to compare the effect size of the differential expression [as measured by the $log_2(fold - change)$ in expression] and its significance [$-log_10(p - value)$].

```
## 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)
```



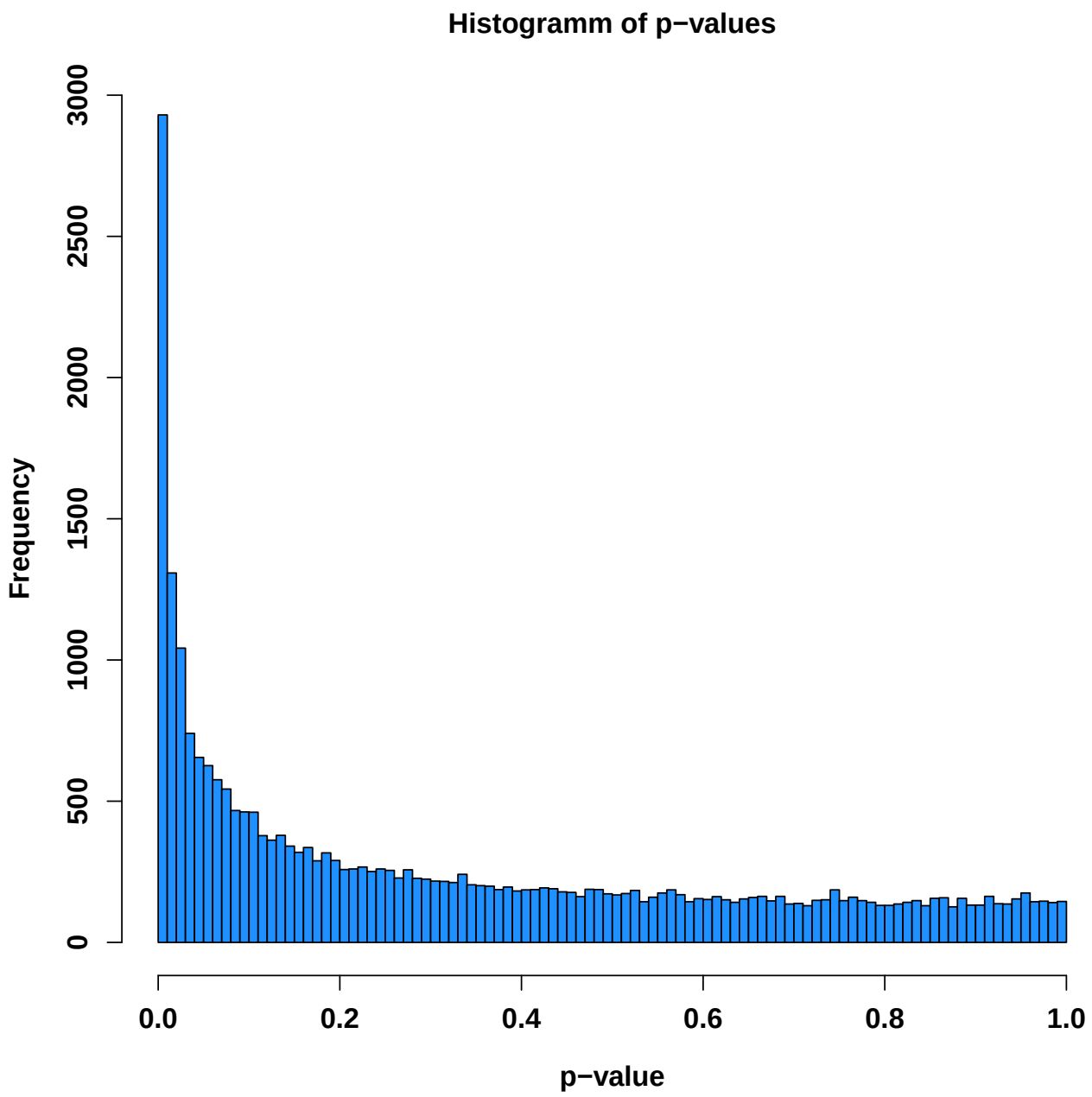
```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC <- fit$coefficients[, 2]
p.mod <- fit$p.value[, 2]
q.mod <- qvalue(p.mod)$q
res_gene <- data.frame(log2FC, p.mod, q.mod)
rownames(res_gene) <- rownames(counts)

## Determine the number of genes differentially expressed at q<0.05
sum(res_gene$q.mod < 0.05)
```

```
## [1] 2974
```

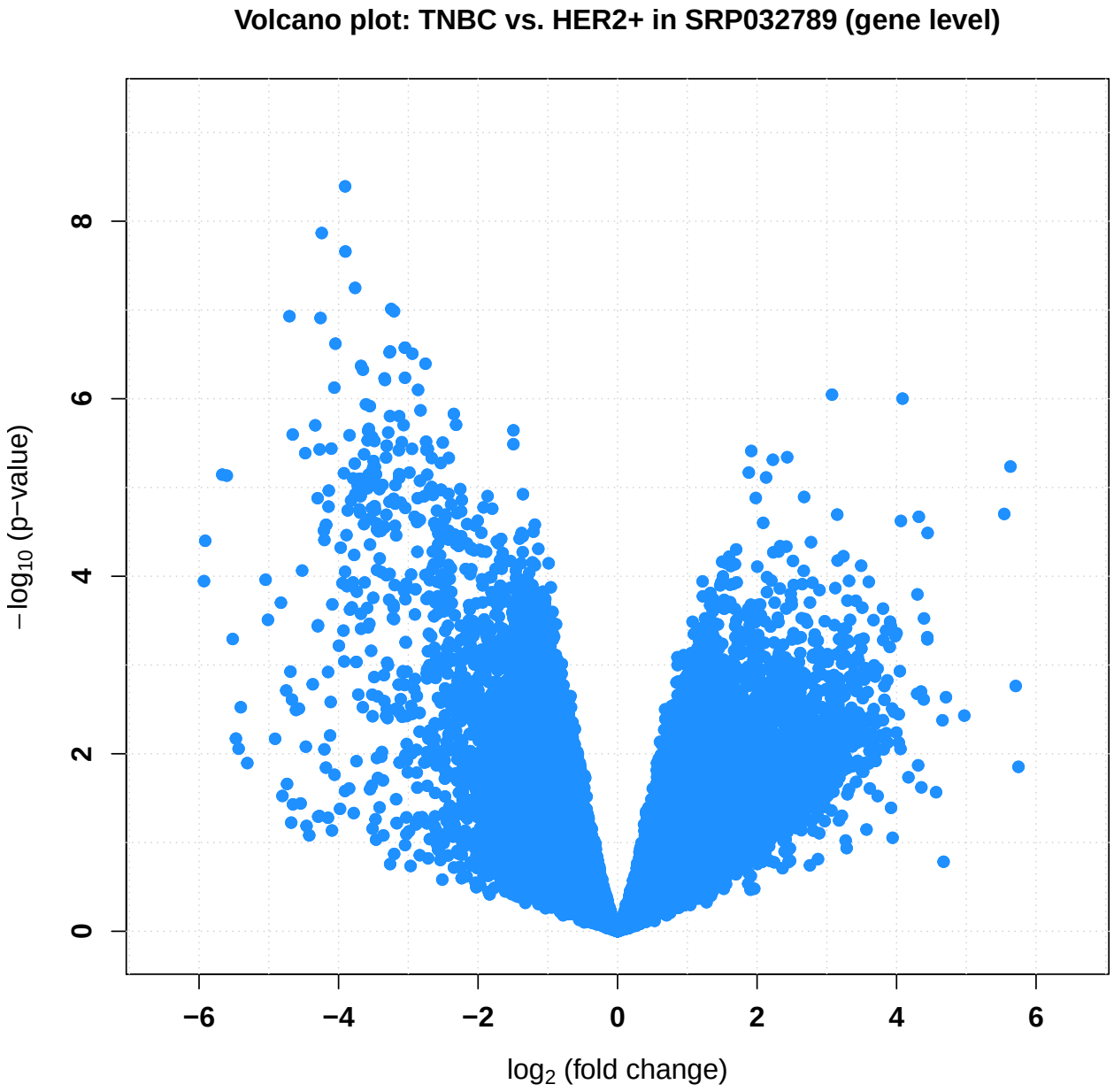
```
table(res_gene$log2FC[res_gene$q.mod < 0.05] > 0 )
```

```
##
## FALSE  TRUE
##  1612  1362
```

```
## Histogram of p-values
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)
```




```
## 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)')
```



2 Gene set enrichment 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 Kolomogorov-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.

```
toens <- function(x) {
    res <- x
    names(res) <- gsub('\\..*', '', names(x))
    return(res)
}

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

```
##
## Building most specific GOs .....
## Loading required package: org.Hs.eg.db
##
##  ( 10869 GO terms found. )
##
## Build GO DAG topology ..........
##  ( 14712 GO terms and 34861 relations. )
##
## Annotating nodes ...............
##  ( 14653 genes annotated to the GO terms. )
```

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

```
##
##           -- Weight01 Algorithm --
##
##       the algorithm is scoring 14712 nontrivial nodes
##       parameters:
##           test statistic: ks
##           score order: increasing
##
##   Level 20:  1 nodes to be scored    (0 eliminated genes)
##
##   Level 19:  8 nodes to be scored    (0 eliminated genes)
##
##   Level 18:  18 nodes to be scored   (1 eliminated genes)
##
##   Level 17:  44 nodes to be scored   (30 eliminated genes)
##
##   Level 16:  119 nodes to be scored  (84 eliminated genes)
##
##   Level 15:  242 nodes to be scored  (178 eliminated genes)
##
##   Level 14:  481 nodes to be scored  (528 eliminated genes)
```

```
##
##   Level 13:  834 nodes to be scored  (1215 eliminated genes)
##
##   Level 12:  1212 nodes to be scored (2372 eliminated genes)
##
##   Level 11:  1559 nodes to be scored (4473 eliminated genes)
##
##   Level 10:  1947 nodes to be scored (6232 eliminated genes)
##
##   Level 9:   2057 nodes to be scored (8549 eliminated genes)
##
##   Level 8:   1949 nodes to be scored (10328 eliminated genes)
##
##   Level 7:   1794 nodes to be scored (11631 eliminated genes)
##
##   Level 6:   1315 nodes to be scored (12606 eliminated genes)
##
##   Level 5:   729 nodes to be scored  (13331 eliminated genes)
##
##   Level 4:   304 nodes to be scored  (13908 eliminated genes)
##
##   Level 3:   77 nodes to be scored   (14160 eliminated genes)
##
##   Level 2:   21 nodes to be scored   (14343 eliminated genes)
##
##   Level 1:   1 nodes to be scored    (14427 eliminated genes)
```

```
bptest
```

```
##
## Description: biological process
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 14712 GO terms scored: 53 terms with p < 0.01
## Annotation data:
##     Annotated genes: 14653
##     Significant genes: 1457
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 14712
```

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

```
##         GO.ID                                      Term Annotated
## 1  GO:0016579                protein deubiquitination       115
## 2  GO:0016569                  chromatin modification       509
## 3  GO:0030049                 muscle filament sliding        32
## 4  GO:0045494           photoreceptor cell maintenance        32
```

```
## 5  GO:0033962 cytoplasmic mRNA processing body assembly        20
## 6  GO:0050776             regulation of immune response       762
## 7  GO:0000042                protein targeting to Golgi        18
## 8  GO:0030521       androgen receptor signaling pathway        60
## 9  GO:0007050                           cell cycle arrest       229
## 10 GO:0071557              histone H3-K27 demethylation         4
##    Significant Expected    pval
## 1           18    11.43   1e-05
## 2           70    50.61 0.00019
## 3            6     3.18 0.00023
## 4            4     3.18 0.00036
## 5            4     1.99 0.00061
## 6           49    75.77 0.00097
## 7            6     1.79 0.00100
## 8           14     5.97 0.00110
## 9           32    22.77 0.00118
## 10           4     0.40 0.00144
```

3 Exon level analysis

As above, we are 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: 329092 20
## metadata(0):
## assays(1): counts
```

```
## rownames(329092): ENSG00000000003.14 ENSG00000000003.14 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(0):
## colnames(20): SRR1027171 SRR1027173 ... SRR1027190 SRR1027172
## colData names(21): project sample ... title characteristics
```

```
## This is the sample phenotype data provided by the recount project
colData(rse_exon)
```

```
## DataFrame with 20 rows and 21 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_downloaded
##                                <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_downloaded 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
##            sharq_beta_tissue sharq_beta_cell_type
##                  <character>          <character>
## SRR1027171            breast                  esc
## SRR1027173            breast                  esc
## SRR1027174            breast                  esc
## SRR1027175            breast                  esc
## SRR1027176            breast                  esc
## ...                      ...                  ...
## SRR1027187            breast                  esc
## SRR1027188            breast                  esc
## SRR1027189            breast                  esc
## SRR1027190            breast                  esc
## SRR1027172            breast                  esc
##            biosample_submission_date biosample_publication_date
##                          <character>                <character>
## SRR1027171   2013-11-07T12:40:22.203    2013-11-08T01:11:17.160
## SRR1027173   2013-11-07T12:40:32.283    2013-11-08T01:11:14.827
## SRR1027174   2013-11-07T12:40:28.283    2013-11-08T01:11:52.283
## SRR1027175   2013-11-07T12:40:34.343    2013-11-08T01:11:15.963
## SRR1027176   2013-11-07T12:40:36.303    2013-11-08T01:11:46.430
## ...                             ...                        ...
## SRR1027187   2013-11-07T12:40:56.180    2013-11-08T01:11:29.587
## SRR1027188   2013-11-07T12:40:58.170    2013-11-08T01:12:06.660
## SRR1027189   2013-11-07T12:40:20.227    2013-11-08T01:11:33.080
## SRR1027190   2013-11-07T12:40:18.090    2013-11-08T01:12:11.320
## SRR1027172   2013-11-07T12:40:26.217    2013-11-08T01:11:45.250
##              biosample_update_date avg_read_length geo_accession
##                        <character>       <integer>   <character>
## SRR1027171 2014-03-07T16:09:38.542             120    GSM1261016
## SRR1027173 2014-03-07T16:09:38.698             120    GSM1261018
## SRR1027174 2014-03-07T16:09:38.637             120    GSM1261019
## SRR1027175 2014-03-07T16:09:38.731             120    GSM1261020
## SRR1027176 2014-03-07T16:09:38.768             120    GSM1261021
## ...                            ...             ...           ...
## SRR1027187 2014-03-07T16:09:39.093             120    GSM1261032
## SRR1027188 2014-03-07T16:09:39.130             150    GSM1261033
## SRR1027189 2014-03-07T16:09:38.498             150    GSM1261034
## SRR1027190 2014-03-07T16:09:38.469             150    GSM1261035
## SRR1027172 2014-03-07T16:09:38.604              87    GSM1261017
##              bigwig_file       title
##              <character> <character>
## SRR1027171 SRR1027171.bw       TNBC1
## SRR1027173 SRR1027173.bw       TNBC3
## SRR1027174 SRR1027174.bw       TNBC4
## SRR1027175 SRR1027175.bw       TNBC5
## SRR1027176 SRR1027176.bw       TNBC6
## ...                  ...         ...
```

```
## SRR1027187 SRR1027187.bw      HER2-5
## SRR1027188 SRR1027188.bw        NBS1
## SRR1027189 SRR1027189.bw        NBS2
## SRR1027190 SRR1027190.bw        NBS3
## SRR1027172 SRR1027172.bw       TNBC2
##                                   characteristics
##                                   <CharacterList>
## SRR1027171          tumor type: TNBC Breast Tumor
## SRR1027173          tumor type: TNBC Breast Tumor
## SRR1027174          tumor type: TNBC Breast Tumor
## SRR1027175          tumor type: TNBC Breast Tumor
## SRR1027176          tumor type: TNBC Breast Tumor
## ...                                           ...
## SRR1027187 tumor type: HER2 Positive Breast Tumor
## SRR1027188    tumor type: Normal Breast Organoids
## SRR1027189    tumor type: Normal Breast Organoids
## SRR1027190    tumor type: Normal Breast Organoids
## SRR1027172          tumor type: TNBC Breast Tumor
```

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')]

## Save filtered rse object
rse_exon_filt <- rse
rse_exon_filt
```

```
## class: RangedSummarizedExperiment
## dim: 329092 11
## metadata(0):
## assays(1): counts
## rownames(329092): ENSG00000000003.14 ENSG00000000003.14 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(0):
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(22): project sample ... characteristics group
```

```
## Obtain count matrix
counts <- assays(rse_exon_filt)$counts
dim(counts)
```

```
## [1] 329092     11
```

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

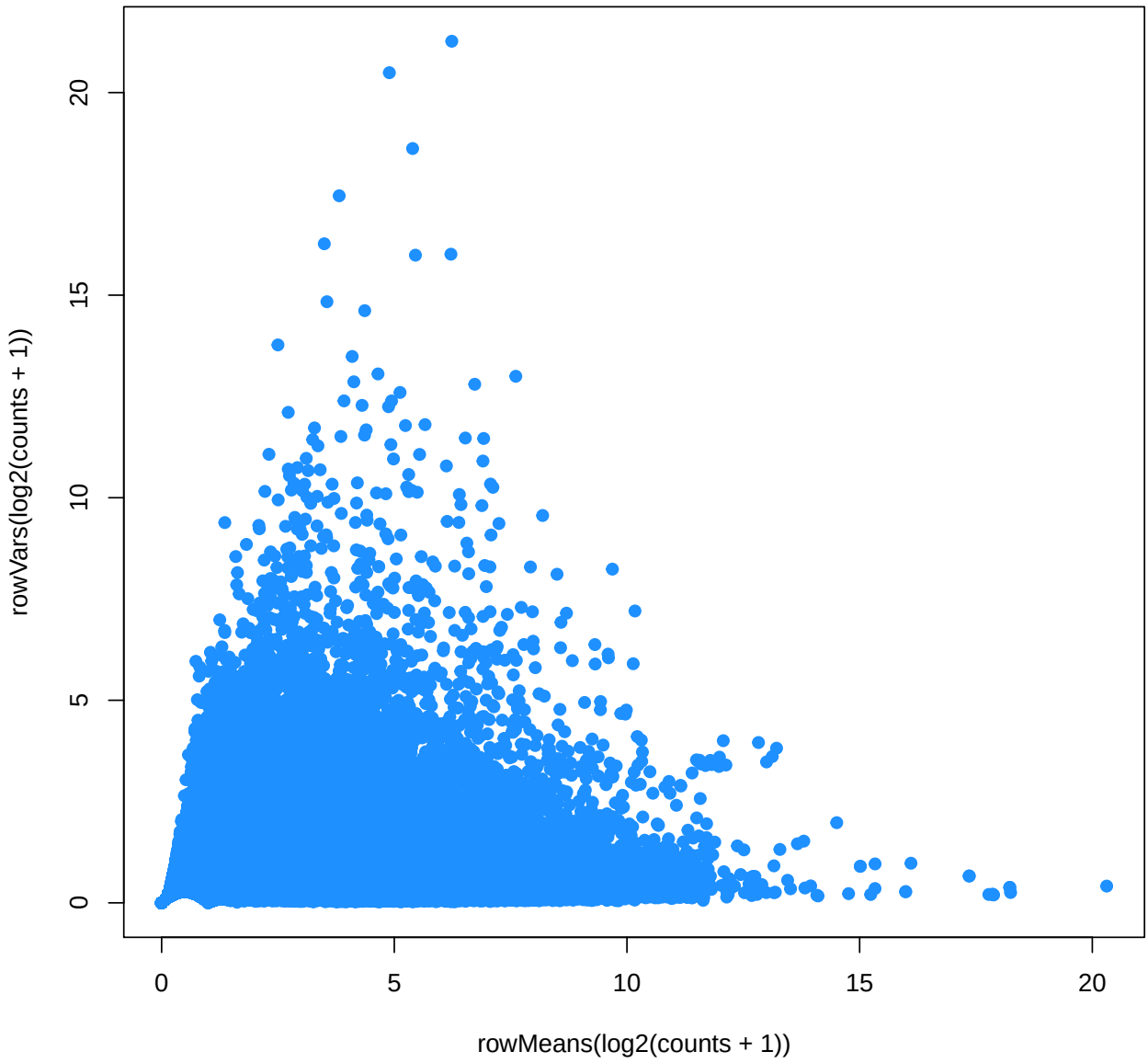
```
## [1] 259626     11
```

```
## Save for gene, exon and junction comparisons
counts_exon <- counts
counts_exon[1:5, 1:5]
```

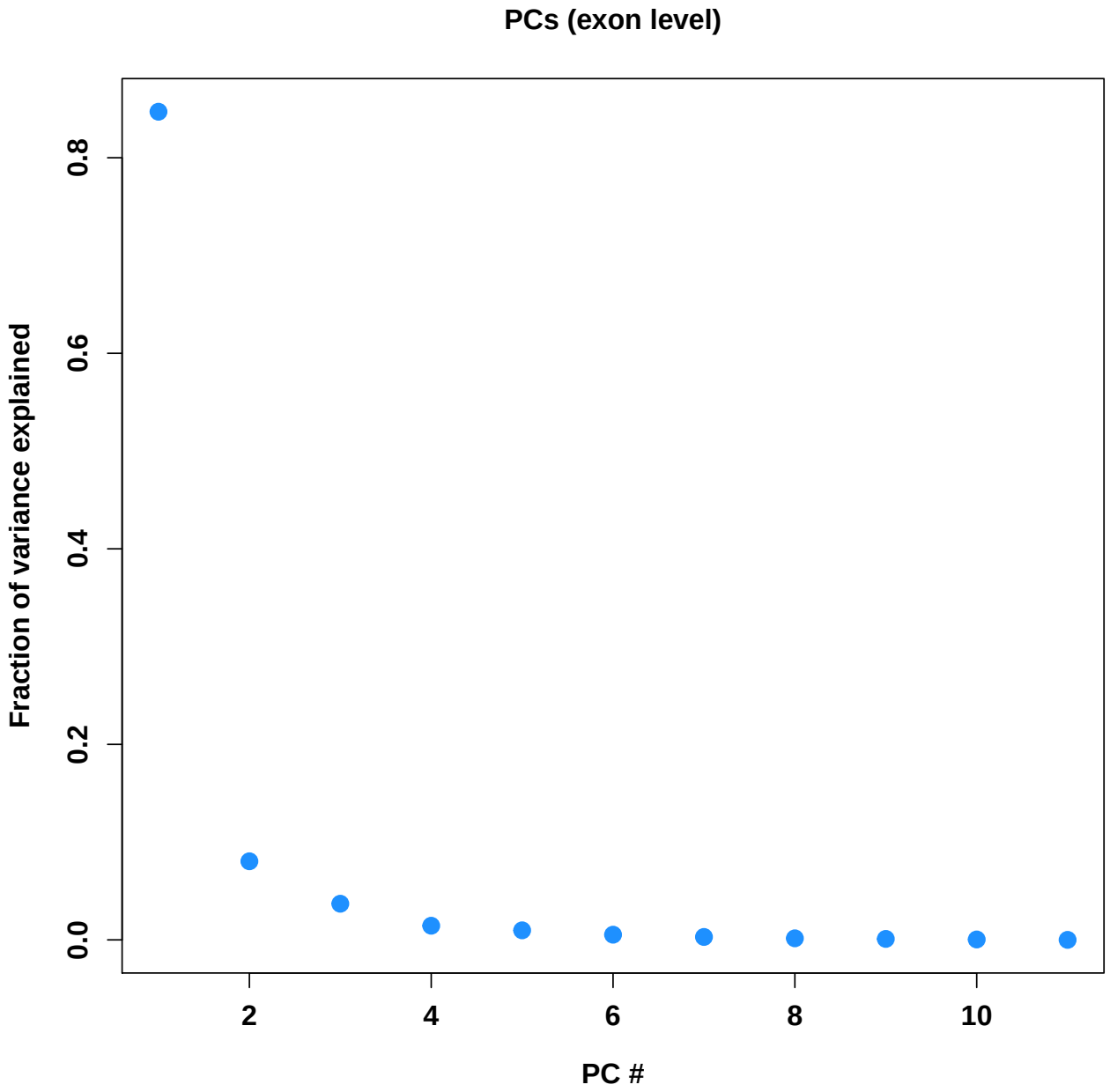
```
##                    SRR1027171 SRR1027173 SRR1027174 SRR1027175 SRR1027176
## ENSG00000000003.14        151        135        169        252         96
## ENSG00000000003.14         25         20         18         30         10
## ENSG00000000003.14          0          0          0          0          0
## ENSG00000000003.14         15         14         14         26          8
## ENSG00000000003.14         22         15         27         30         13
```

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.

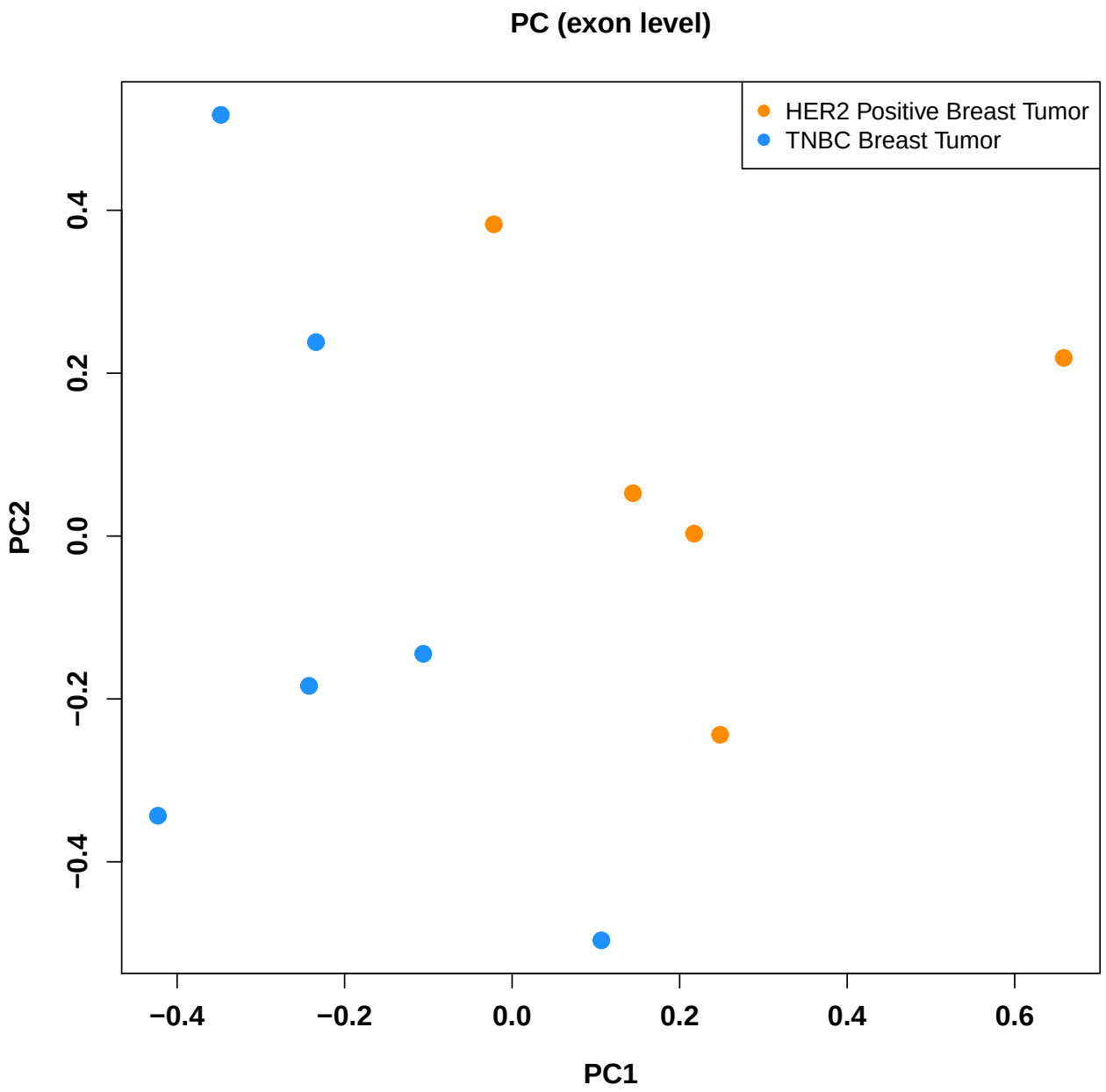
```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse$group))

## 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)')
```




```
## 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))), bg="white")
```



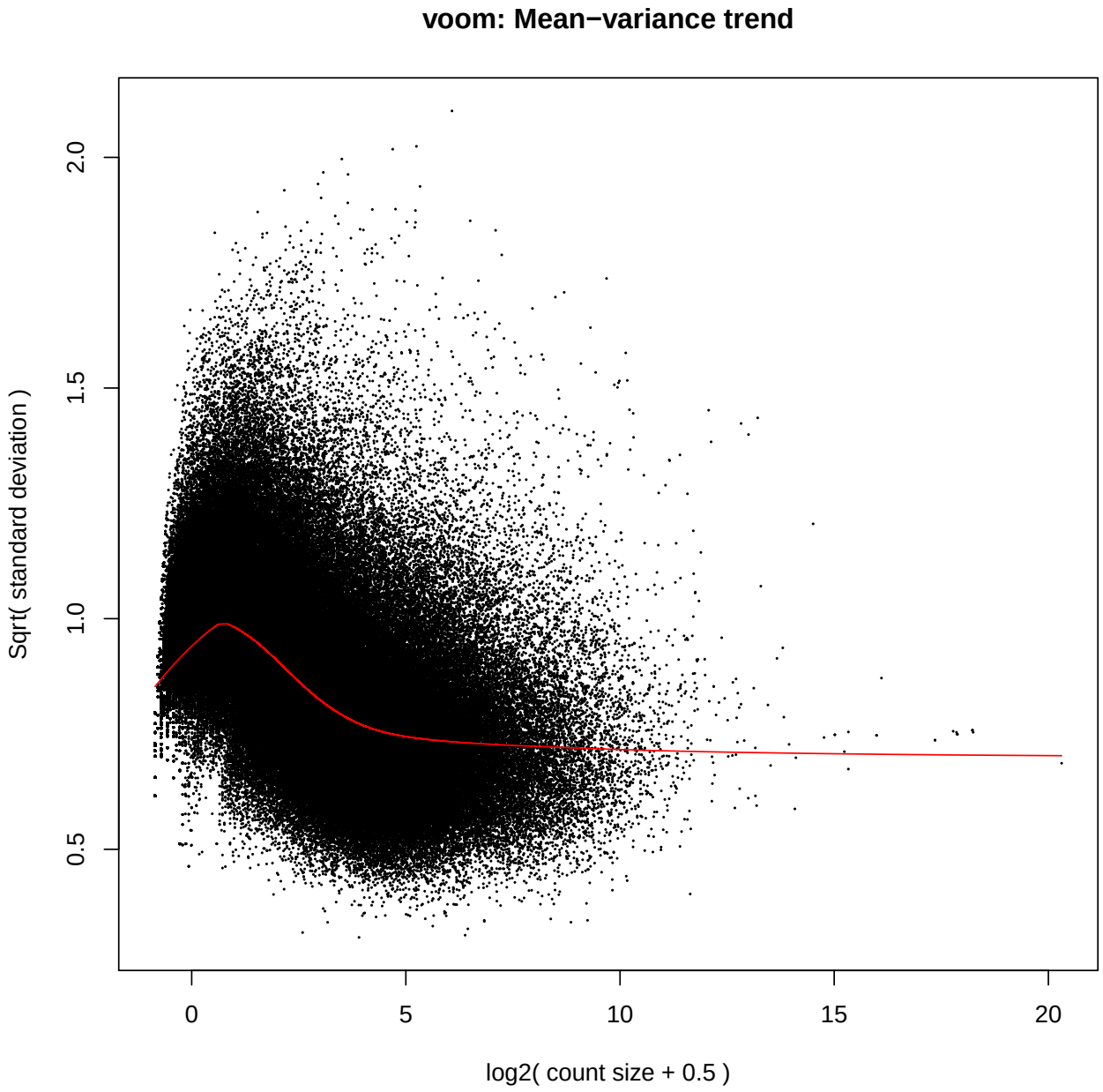
Again, differential expression analysis is carried out using `limma` and `voom`; however, this time at the exon, rather than gene, level. Data are again visualized using a volcano plot to assess the strength [$log_2(fold - change)$ in expression] and its significance [$-log_10(p - value)$].for each exon.

```
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                          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)
```




```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC <- fit$coefficients[, 2]
```

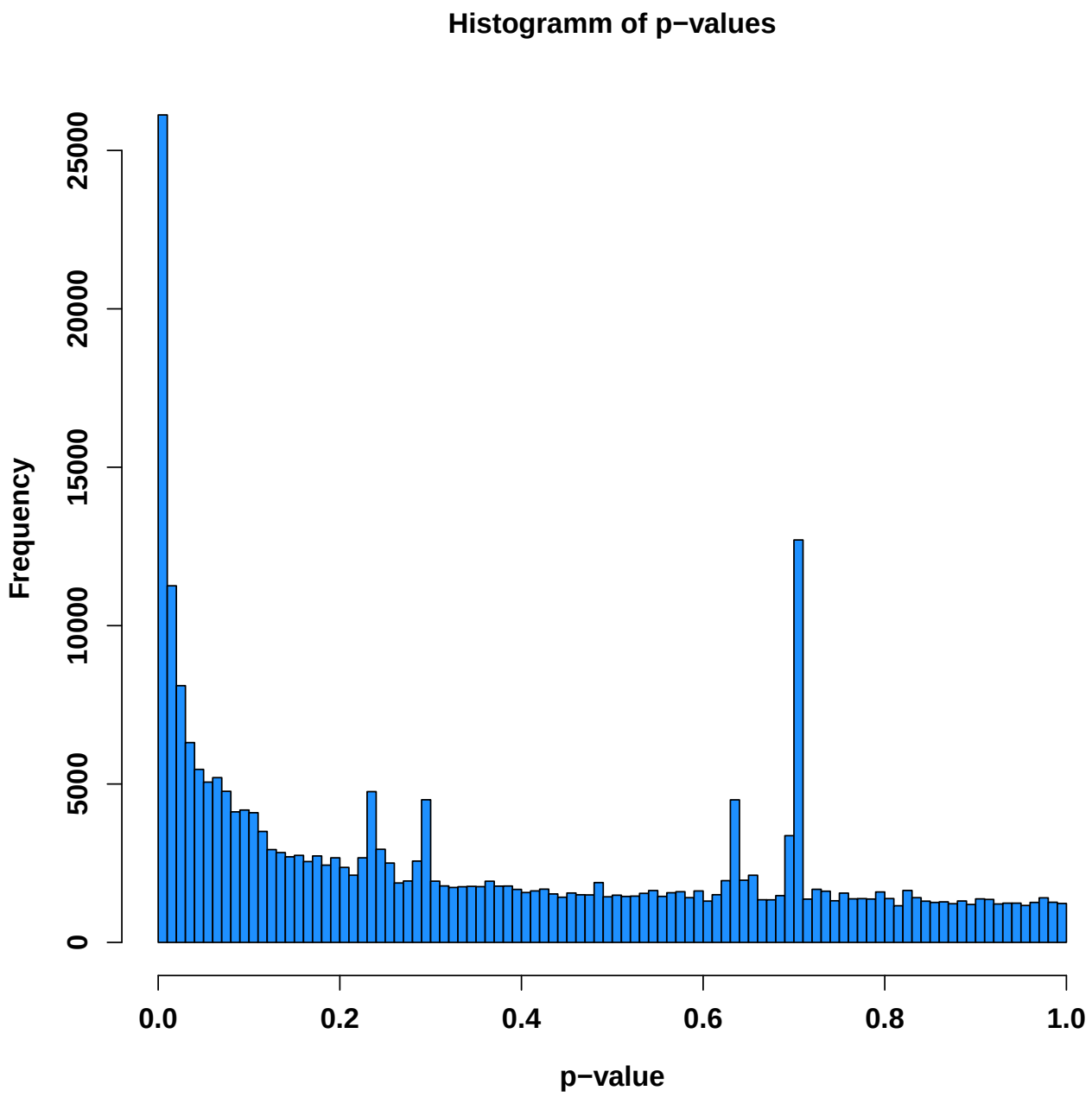
```
p.mod <- fit$p.value[, 2]
q.mod <- qvalue(p.mod)$q
res_exon <- data.frame(log2FC, p.mod, q.mod)


## Determine the number of exons differentially expressed at q<0.05
sum(res_exon$q.mod < 0.05)
```

```
## [1] 27705
```

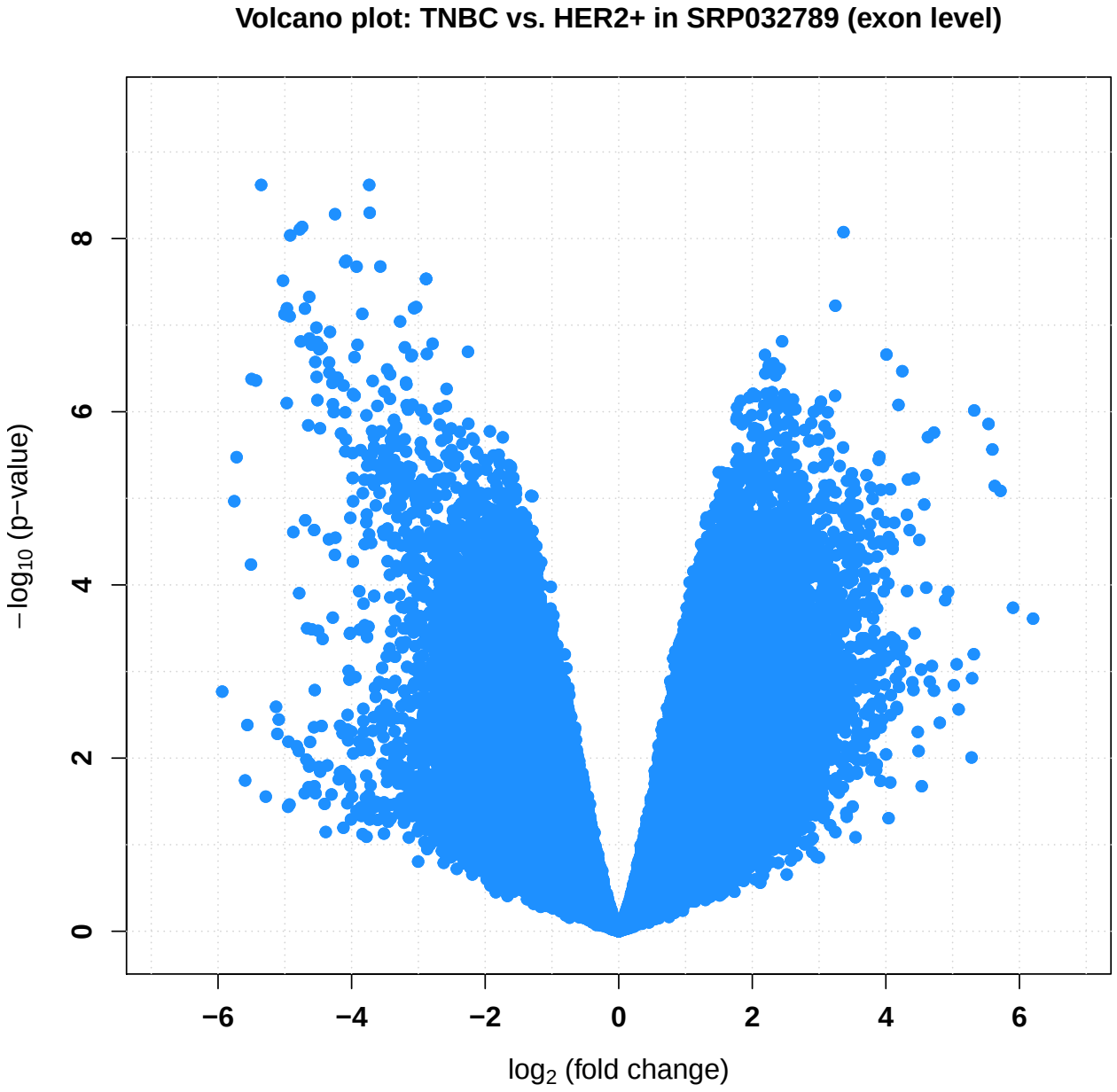
```
table(res_exon$log2FC[res_exon$q.mod < 0.05] > 0 )
```

```
##
## FALSE  TRUE
## 13159 14546
```

```
## Histogram of p-values
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)
```




```
## 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)')
```



4 Junction level analysis

As above, we are interested here in differential expression. However, rather than summarizing across genes, this analysis will look for differential expression at the junction level. In this analysis, we include all junctions 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 junction 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_jx.Rdata'))) {
    download_study(project_info$project, type = 'rse-jx')
}

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

```
## class: RangedSummarizedExperiment
## dim: 672203 20
## metadata(0):
## assays(1): counts
## rownames: NULL
## rowData names(8): junction_id found_junction_gencode_v24 ...
##   symbol class
## colnames(20): SRR1027171 SRR1027173 ... SRR1027190 SRR1027172
## colData names(21): project sample ... title characteristics
```

```
## This is the sample phenotype data provided by the recount project
colData(rse_jx)
```

```
## DataFrame with 20 rows and 21 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_downloaded
##                                <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_downloaded 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
##            sharq_beta_tissue sharq_beta_cell_type
##                  <character>          <character>
## SRR1027171            breast                  esc
## SRR1027173            breast                  esc
## SRR1027174            breast                  esc
## SRR1027175            breast                  esc
## SRR1027176            breast                  esc
## ...                      ...                  ...
## SRR1027187            breast                  esc
## SRR1027188            breast                  esc
## SRR1027189            breast                  esc
## SRR1027190            breast                  esc
## SRR1027172            breast                  esc
##            biosample_submission_date biosample_publication_date
##                          <character>                <character>
## SRR1027171   2013-11-07T12:40:22.203    2013-11-08T01:11:17.160
## SRR1027173   2013-11-07T12:40:32.283    2013-11-08T01:11:14.827
## SRR1027174   2013-11-07T12:40:28.283    2013-11-08T01:11:52.283
## SRR1027175   2013-11-07T12:40:34.343    2013-11-08T01:11:15.963
## SRR1027176   2013-11-07T12:40:36.303    2013-11-08T01:11:46.430
## ...                             ...                        ...
## SRR1027187   2013-11-07T12:40:56.180    2013-11-08T01:11:29.587
## SRR1027188   2013-11-07T12:40:58.170    2013-11-08T01:12:06.660
## SRR1027189   2013-11-07T12:40:20.227    2013-11-08T01:11:33.080
## SRR1027190   2013-11-07T12:40:18.090    2013-11-08T01:12:11.320
```

```
## SRR1027172   2013-11-07T12:40:26.217    2013-11-08T01:11:45.250
##              biosample_update_date avg_read_length geo_accession
##                        <character>       <integer>   <character>
## SRR1027171 2014-03-07T16:09:38.542             120    GSM1261016
## SRR1027173 2014-03-07T16:09:38.698             120    GSM1261018
## SRR1027174 2014-03-07T16:09:38.637             120    GSM1261019
## SRR1027175 2014-03-07T16:09:38.731             120    GSM1261020
## SRR1027176 2014-03-07T16:09:38.768             120    GSM1261021
## ...                            ...             ...           ...
## SRR1027187 2014-03-07T16:09:39.093             120    GSM1261032
## SRR1027188 2014-03-07T16:09:39.130             150    GSM1261033
## SRR1027189 2014-03-07T16:09:38.498             150    GSM1261034
## SRR1027190 2014-03-07T16:09:38.469             150    GSM1261035
## SRR1027172 2014-03-07T16:09:38.604              87    GSM1261017
##              bigwig_file       title
##              <character> <character>
## SRR1027171 SRR1027171.bw       TNBC1
## SRR1027173 SRR1027173.bw       TNBC3
## SRR1027174 SRR1027174.bw       TNBC4
## SRR1027175 SRR1027175.bw       TNBC5
## SRR1027176 SRR1027176.bw       TNBC6
## ...                   ...         ...
## SRR1027187 SRR1027187.bw      HER2-5
## SRR1027188 SRR1027188.bw        NBS1
## SRR1027189 SRR1027189.bw        NBS2
## SRR1027190 SRR1027190.bw        NBS3
## SRR1027172 SRR1027172.bw       TNBC2
##                                   characteristics
##                                   <CharacterList>
## SRR1027171          tumor type: TNBC Breast Tumor
## SRR1027173          tumor type: TNBC Breast Tumor
## SRR1027174          tumor type: TNBC Breast Tumor
## SRR1027175          tumor type: TNBC Breast Tumor
## SRR1027176          tumor type: TNBC Breast Tumor
## ...                                           ...
## SRR1027187 tumor type: HER2 Positive Breast Tumor
## SRR1027188    tumor type: Normal Breast Organoids
## SRR1027189    tumor type: Normal Breast Organoids
## SRR1027190    tumor type: Normal Breast Organoids
## SRR1027172          tumor type: TNBC Breast Tumor
```

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_jx`). 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 junction within genes that had been analyzed previously.

```
## Scale counts by taking into account the total coverage per sample
rse <- scale_counts(rse_jx, by = 'mapped_reads', round = FALSE)

## 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')]

## Save filtered rse object
rse_jx_filt <- rse
rse_jx_filt
```

```
## class: RangedSummarizedExperiment
## dim: 672203 11
## metadata(0):
## assays(1): counts
## rownames: NULL
## rowData names(8): junction_id found_junction_gencode_v24 ...
##   symbol class
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(22): project sample ... characteristics group
```

```
## Obtain count matrix
counts <- assays(rse_jx_filt)$counts
dim(counts)
```

```
## [1] 672203     11
```

```
##### Start: Obtain geneIDs for juctions
## Obtain geneIDs
gene_id <- rownames(counts_gene)

## Save number of genes that a junctions maps to
## We will exclude non-unique junctions later
num_genes <- lapply(rowData(rse_jx_filt)$gene_id_proposed, function(x) length(x))
```

```
num_genes <- unlist(num_genes)

## Save only the first gene_id
jx_gene_id <- lapply(rowData(rse_jx_filt)$gene_id, function(x) x[1])
jx_gene_id <- unlist(jx_gene_id)

## There are NAs: not every junctions is annotated
jx_gene_id[1:100]
```

```
##   [1] NA                  NA                  "ENSG00000227232.5"
##   [4] NA                  NA                  "ENSG00000227232.5"
##   [7] NA                  NA                  NA
##  [10] NA                  NA                  NA
##  [13] NA                  NA                  NA
##  [16] NA                  NA                  NA
##  [19] NA                  NA                  NA
##  [22] NA                  NA                  NA
##  [25] NA                  NA                  "ENSG00000227232.5"
##  [28] NA                  NA                  NA
##  [31] NA                  NA                  NA
##  [34] NA                  NA                  NA
##  [37] NA                  NA                  NA
##  [40] NA                  NA                  NA
##  [43] NA                  NA                  NA
##  [46] NA                  NA                  NA
##  [49] NA                  NA                  NA
##  [52] NA                  NA                  NA
##  [55] NA                  "ENSG00000238009.6" NA
##  [58] NA                  NA                  NA
##  [61] "ENSG00000238009.6" NA                  NA
##  [64] NA                  NA                  NA
##  [67] NA                  NA                  NA
##  [70] "ENSG00000279928.1" "ENSG00000279457.3" NA
##  [73] "ENSG00000279457.3" "ENSG00000279457.3" NA
##  [76] "ENSG00000279457.3" "ENSG00000279457.3" "ENSG00000279457.3"
##  [79] NA                  NA                  NA
##  [82] "ENSG00000279457.3" "ENSG00000279457.3" NA
##  [85] NA                  NA                  "ENSG00000279457.3"
##  [88] NA                  NA                  NA
##  [91] NA                  NA                  NA
##  [94] NA                  NA                  NA
##  [97] NA                  NA                  NA
## [100] NA
```

```
## Compare lengths
length(jx_gene_id) == dim(counts)[1]
```

```
## [1] TRUE
```

```
## Find non-unique mapping junctions
double_jx <- which(num_genes >1)

## Check non-unique mapping junctions
rowData(rse_jx_filt)[double_jx, 'gene_id_proposed']
```

```
## CharacterList of length 7100
## [[1]] ENSG00000237094.11 ENSG00000239906.1
## [[2]] ENSG00000228327.3 ENSG00000237094.11
## [[3]] ENSG00000188157.13 ENSG00000217801.9
## [[4]] ENSG00000188157.13 ENSG00000217801.9
## [[5]] ENSG00000186827.10 ENSG00000186891.13
## [[6]] ENSG00000127054.19 ENSG00000240731.1
## [[7]] ENSG00000160072.19 ENSG00000215915.9
## [[8]] ENSG00000160072.19 ENSG00000215915.9
## [[9]] ENSG00000160072.19 ENSG00000215915.9
## [[10]] ENSG00000160072.19 ENSG00000215915.9
## ...
## <7090 more elements>
```

```
## Set non-unique mapping junctions to "NA" in
jx_gene_id[double_jx] <- NA

rownames(counts) <- jx_gene_id
##### End: Obtain geneIDs for juctions



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

```
## [1] 227127     11
```

```
## Since we only look at a subset of samples, there are many junctions with zero counts
## We remove them
counts <- counts[apply(counts, 1, sum) > 0, ]
dim(counts)
```

```
## [1] 197703     11
```

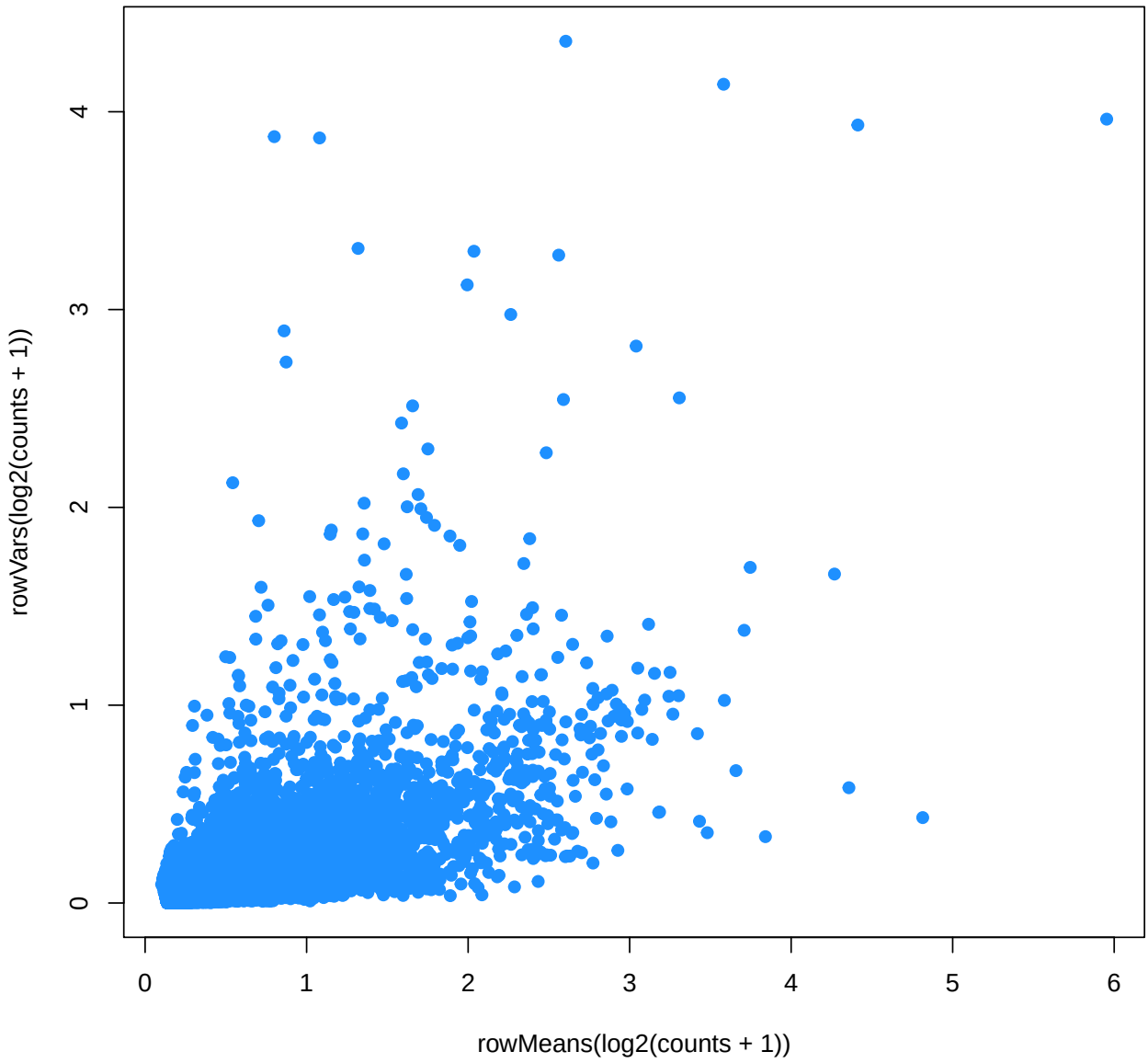
```
## Remove junctions with low counts across samples
counts <- counts[rowMeans(counts) > 0.1, ]

## Save for gene, exon and junction comparisons
counts_jx <- counts
counts_jx[1:10, ]
```

```
##                    SRR1027171 SRR1027173 SRR1027174 SRR1027175 SRR1027176
## ENSG00000188976.10 0.20446141 0.10649173  0.1399513  0.1125870 0.43985733
## ENSG00000188976.10 0.10223070 0.04259669  0.1516139  0.1125870 0.13076839
## ENSG00000188976.10 0.14056722 0.07454421  0.1166261  0.1501159 0.16643250
## ENSG00000188976.10 0.25557676 0.10649173  0.2099269  0.2501932 0.24964875
## ENSG00000188976.10 0.10223070 0.02129835  0.2682400  0.2501932 0.26153679
## ENSG00000188290.10 0.11500954 0.03194752  0.1516139  0.1876449 0.10699232
## ENSG00000187608.8  0.60060539 0.03194752  0.1632765  0.3627802 0.07132822
## ENSG00000188157.13 0.03833651 0.11714091  0.1516139  0.2251739 0.14265643
## ENSG00000188157.13 0.24279792 0.21298347  0.1516139  0.3002319 0.11888036
## ENSG00000188157.13 0.19168257 0.12779008  0.1166261  0.1876449 0.26153679
##                    SRR1027183 SRR1027184 SRR1027185  SRR1027186 SRR1027187
## ENSG00000188976.10 0.00000000 0.02473931 0.02160652 0.009604733 0.08586956
## ENSG00000188976.10 0.05579103 0.08246437 0.07562282 0.028814200 0.17173912
## ENSG00000188976.10 0.08368654 0.06597150 0.11883586 0.048023666 0.12021739
```

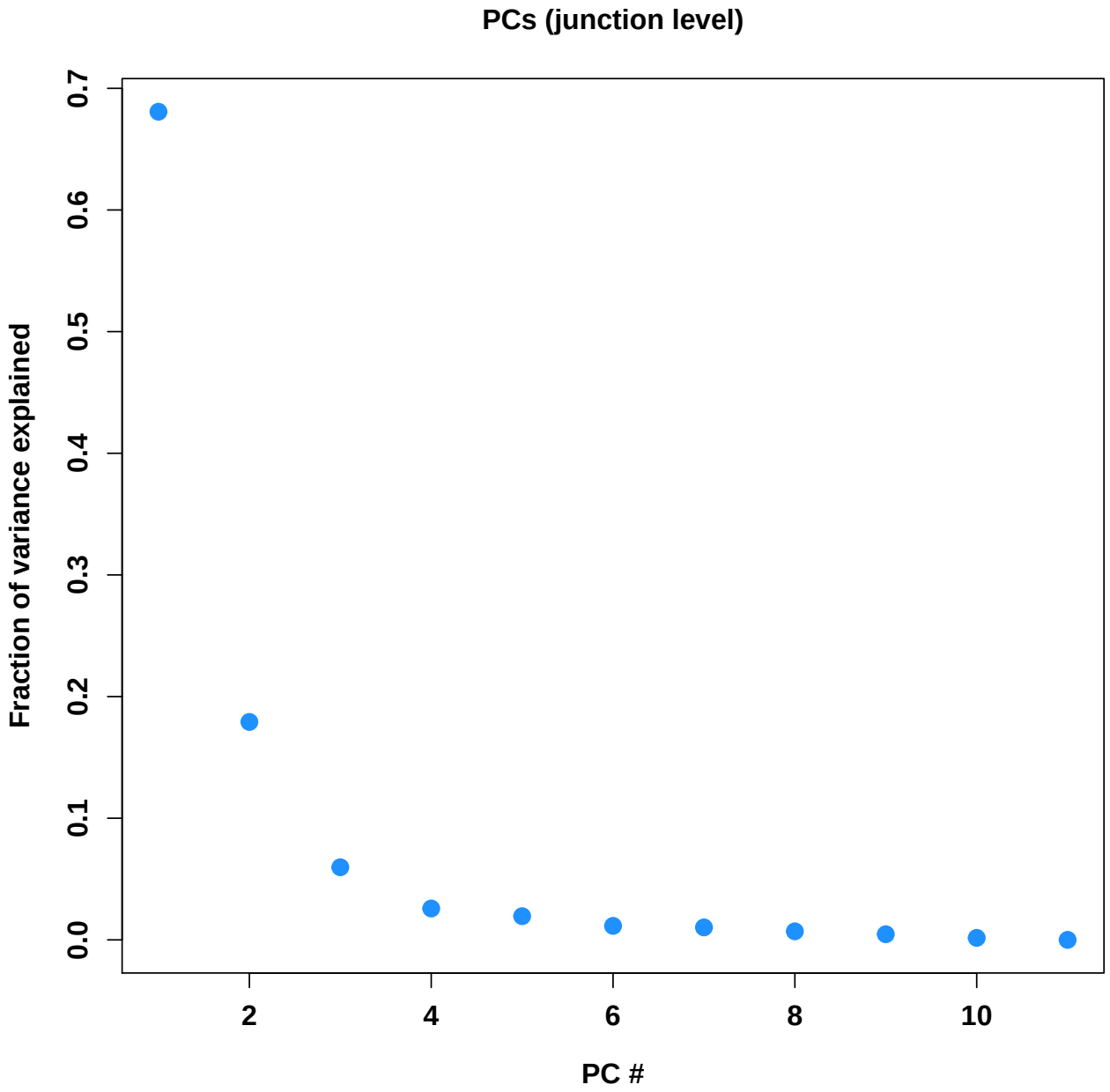
```
## ENSG00000188976.10 0.07438803 0.18966806 0.19445869 0.057628399 0.20608695
## ENSG00000188976.10 0.09298504 0.04123219 0.19445869 0.115256798 0.08586956
## ENSG00000188290.10 0.05579103 0.02473931 0.08642608 0.009604733 0.06869565
## ENSG00000187608.8  0.18597009 0.04123219 0.15124564 0.067233132 0.18891303
## ENSG00000188157.13 0.05579103 0.09071081 0.20526195 0.163280464 0.17173912
## ENSG00000188157.13 0.12088056 0.07421794 0.10803260 0.038418933 0.20608695
## ENSG00000188157.13 0.06508953 0.08246437 0.10803260 0.019209466 0.29195651
##                    SRR1027172
## ENSG00000188976.10  0.5144759
## ENSG00000188976.10  0.2204897
## ENSG00000188976.10  0.3674828
## ENSG00000188976.10  0.5512242
## ENSG00000188976.10  0.4777276
## ENSG00000188290.10  0.3307345
## ENSG00000187608.8   0.5144759
## ENSG00000188157.13  0.3674828
## ENSG00000188157.13  0.5144759
## ENSG00000188157.13  0.5512242
```

As above, to get a better sense of the data, we assess the mean-variance relationship for each junction. 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.

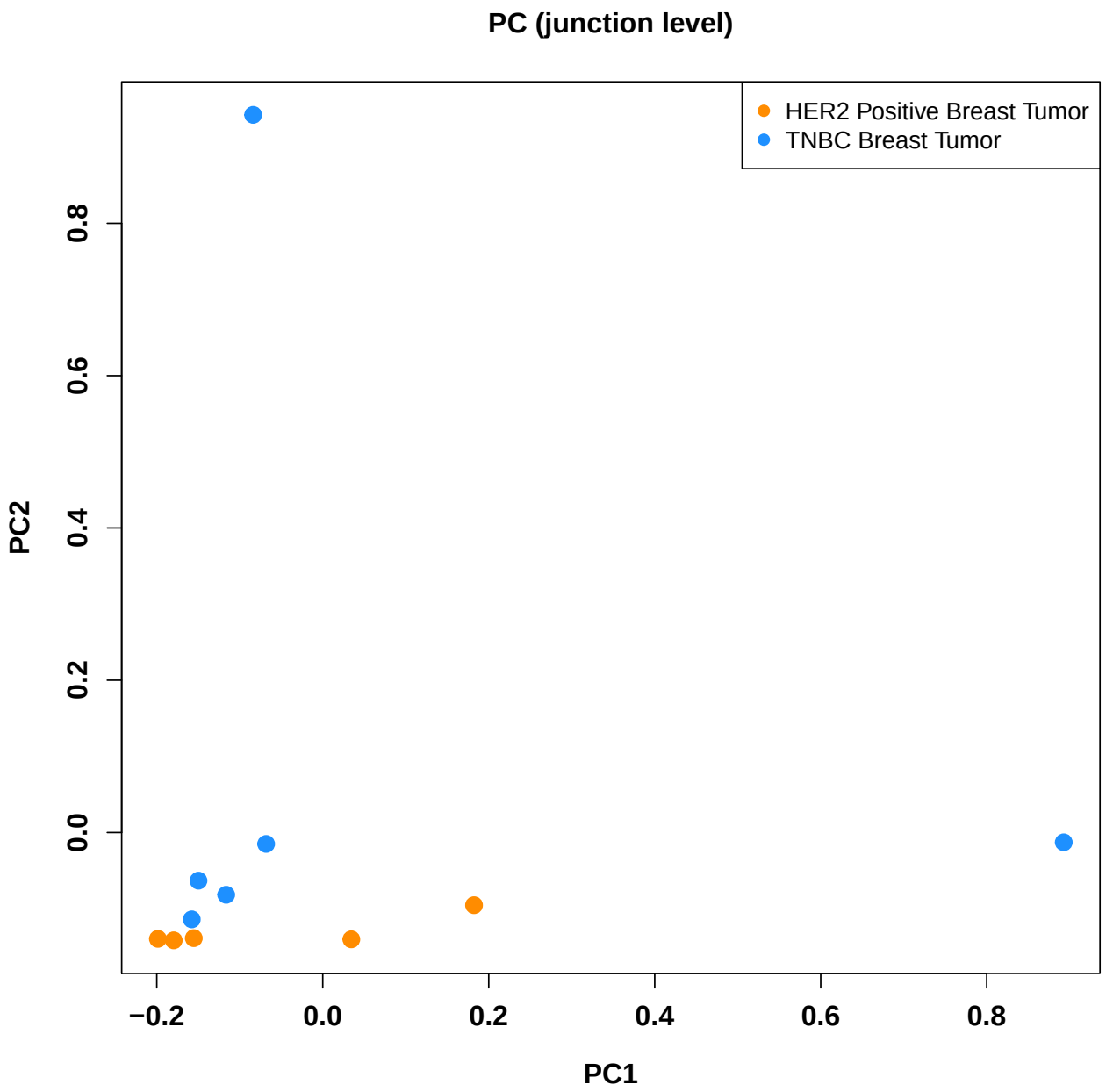
```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse$group))

## 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 (junction 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 (junction level)')
legend('topright', pch = 19, col = trop[c(1, 2)],
       names(summary(as.factor(rse$group))), bg="white")
```



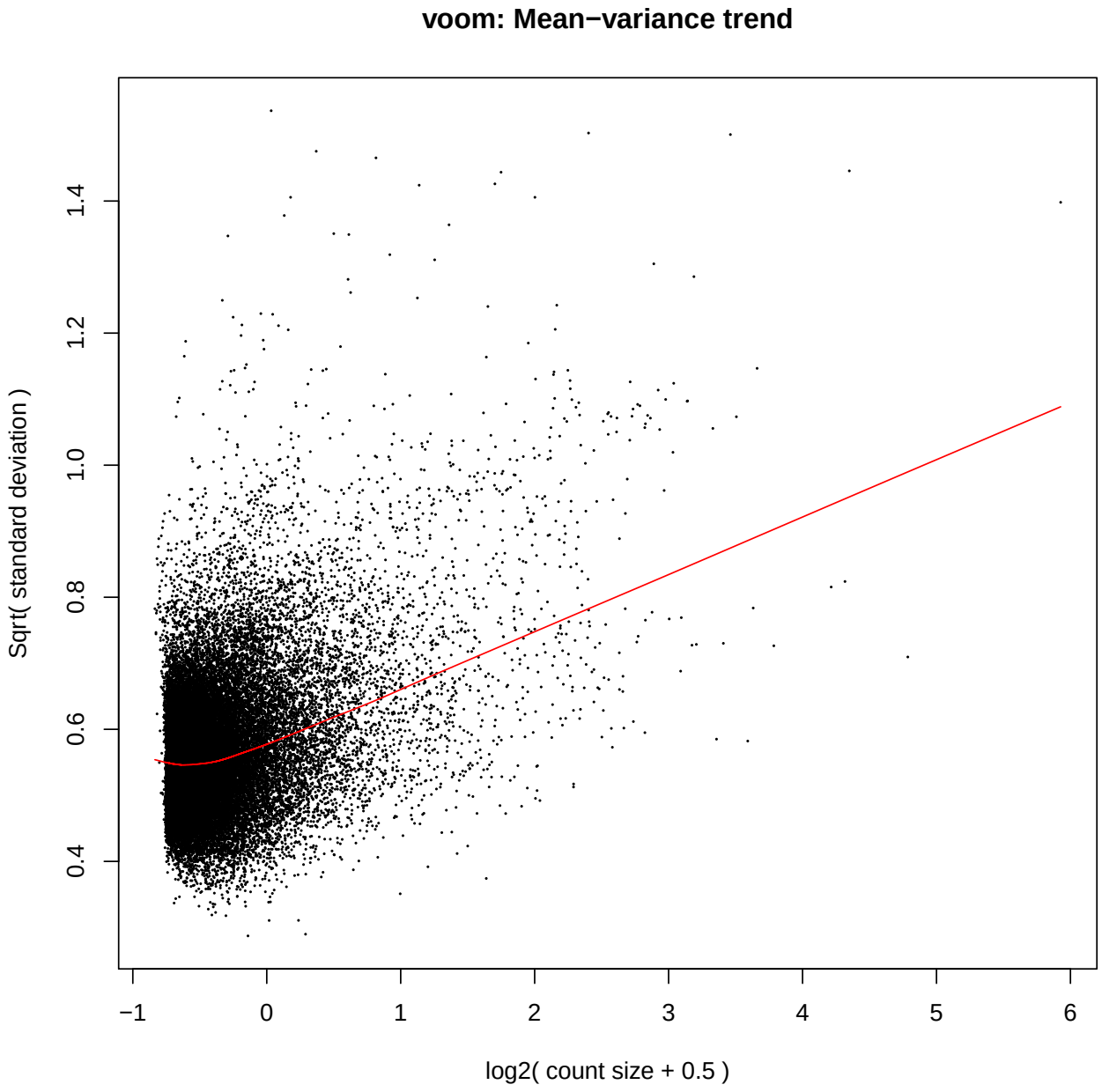
Again, differential expression analysis is carried out using `limma` and `voom`; however, this time at the junction, rather than gene, level. Data are again visualized using a volcano plot to assess the strength [$log_2(fold - change)$ in expression] and its significance [$-log_10(p - value)$] for each junction.

```
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)
```

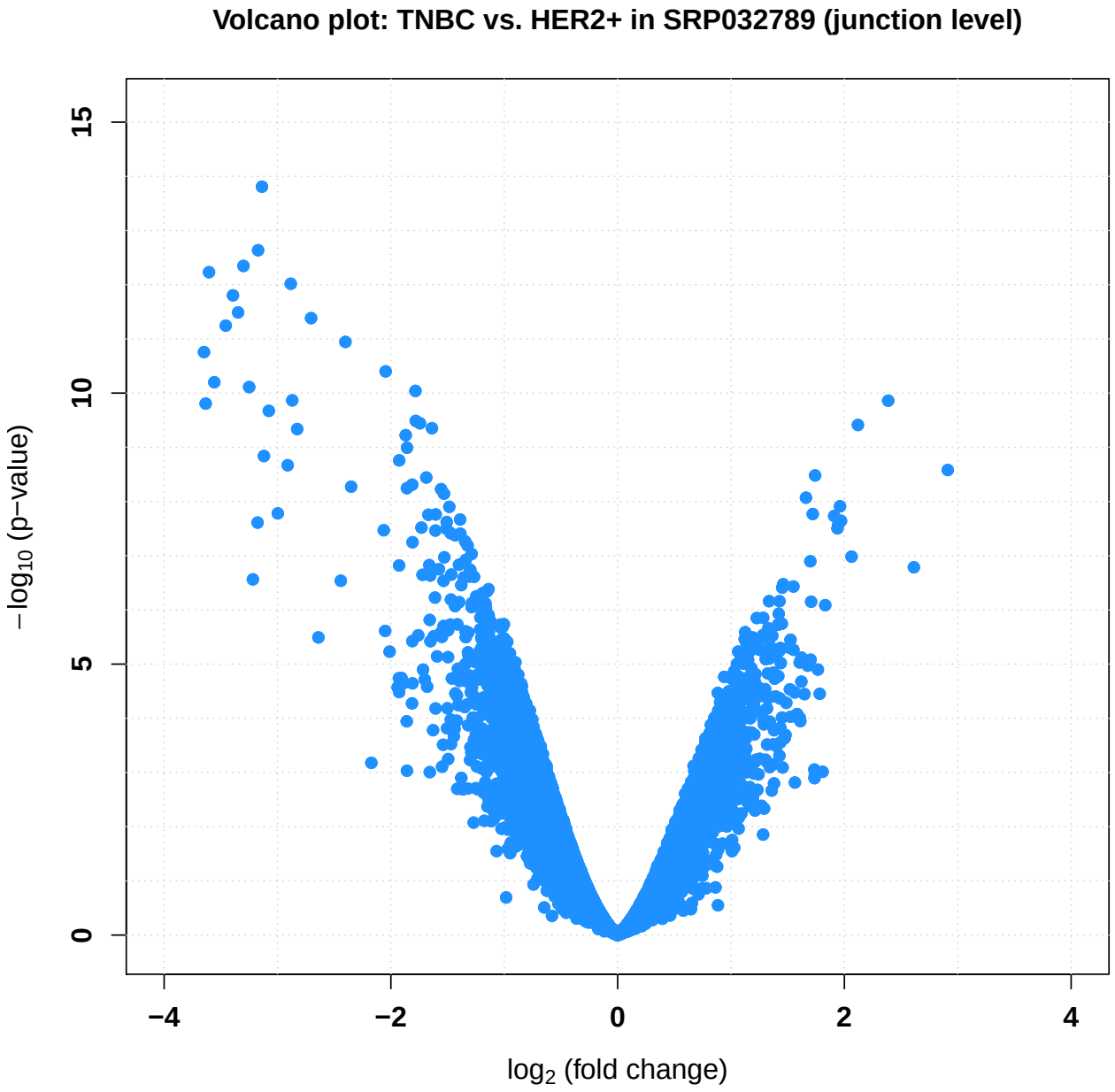



```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC <- fit$coefficients[, 2]
```

```r
p.mod <- fit$p.value[, 2]
q.mod <- qvalue(p.mod)$q
res_jx <- data.frame(log2FC, p.mod, q.mod)

## Determine the number of exons differentially expressed at q<0.05
sum(res_jx$q.mod < 0.05)
```

```
## [1] 23489
```

```r
## 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, 2356, 1), col = 'lightgray', lty = 'dotted')
points(log2FC, -log10(p.mod), pch = 19, col = trop[2])
title('Volcano plot: TNBC vs. HER2+ in SRP032789 (junction level)')
```



5 Comparison of gene, exon, junction, and DER results

To compare findings at the gene, exon, junction, and DER level, we obtained a single exon level [or junction level or DER level] p-value for each gene included at the gene level analysis. To do this, we utilized Simes' rule, such that for each gene included in the gene level analysis, the p-values for exons [or junctions or DERs] within that gene were extracted and sorted. Each exon level [or junction level or DER level] p-value is then multiplied by the number of exons [or junctions or DERs] present within the gene. For each exon [or junction or DER] (1,2...n), this quantity is divided by that exon's rank [or junction's rank or DER's rank] (where 1=most significant exon [or junction or DER] and n=least significant). The minimum value from this calcultion is assigned as the exon level [or junction level or DER level] p-value at each gene. DER results are loaded from the DER analysis report that is described and rendered in `recount_DER_SRP032789.*`

```
## Obtain geneIDs
gene_id <- unique(rownames(counts_exon))

## Calculate p-values for genes with Simes' rule
p_exon_gene <- NULL
for(i in seq_len(length(gene_id))){
    p_exon <- res_exon$p.mod[rownames(counts_exon) %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_exon_gene[i] <- min(p_exon_simes)
}
names(p_exon_gene) <- gene_id

## Determine the number of 'gene level exons' differentially expressed q < 0.05
q_exon_gene <- qvalue(p_exon_gene)$q
sum(q_exon_gene < 0.05)
```

```
## [1] 10977
```

```
## 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 = toens(q_exon_gene),
    geneSelectionFun = interesting,
    annotationFun = annFUN.org, mapping = 'org.Hs.eg.db', ID = 'ensembl')
```

```
##
## Building most specific GOs .....

##  ( 10869 GO terms found. )

##
## Build GO DAG topology ..........

##  ( 14712 GO terms and 34861 relations. )

##
## Annotating nodes ...............

##  ( 14653 genes annotated to the GO terms. )
```

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

```
##
##           -- Weight01 Algorithm --
##
##       the algorithm is scoring 14712 nontrivial nodes
##       parameters:
##           test statistic: ks
##           score order: increasing
```

```
##
##   Level 20:  1 nodes to be scored    (0 eliminated genes)
##
##   Level 19:  8 nodes to be scored    (0 eliminated genes)
##
##   Level 18:  18 nodes to be scored   (1 eliminated genes)
##
##   Level 17:  44 nodes to be scored   (30 eliminated genes)
##
##   Level 16:  119 nodes to be scored  (84 eliminated genes)
##
##   Level 15:  242 nodes to be scored  (178 eliminated genes)
##
##   Level 14:  481 nodes to be scored  (528 eliminated genes)
##
##   Level 13:  834 nodes to be scored  (1215 eliminated genes)
##
##   Level 12:  1212 nodes to be scored (2372 eliminated genes)
##
##   Level 11:  1559 nodes to be scored (4473 eliminated genes)
##
##   Level 10:  1947 nodes to be scored (6232 eliminated genes)
##
##   Level 9:   2057 nodes to be scored (8549 eliminated genes)
##
##   Level 8:   1949 nodes to be scored (10328 eliminated genes)
##
##   Level 7:   1794 nodes to be scored (11631 eliminated genes)
##
##   Level 6:   1315 nodes to be scored (12606 eliminated genes)
##
##   Level 5:   729 nodes to be scored  (13331 eliminated genes)
##
##   Level 4:   304 nodes to be scored  (13908 eliminated genes)
##
##   Level 3:   77 nodes to be scored   (14160 eliminated genes)
##
##   Level 2:   21 nodes to be scored   (14343 eliminated genes)
##
##   Level 1:   1 nodes to be scored    (14427 eliminated genes)
```

```
bptest
```

```
##
## Description: biological process
```

```
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 14712 GO terms scored: 92 terms with p < 0.01
## Annotation data:
##     Annotated genes: 14653
##     Significant genes: 6287
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 14712
```

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

```
##         GO.ID                                                        Term
## 1  GO:0016579                                   protein deubiquitination
## 2  GO:0098609                                         cell-cell adhesion
## 3  GO:0000398                               mRNA splicing, via spliceosome
## 4  GO:0007049                                                   cell cycle
## 5  GO:0051493                     regulation of cytoskeleton organization
## 6  GO:0048025    negative regulation of mRNA splicing, via spliceosome
## 7  GO:0016569                                     chromatin modification
## 8  GO:0000381 regulation of alternative mRNA splicing, via spliceosome
## 9  GO:0016032                                               viral process
## 10 GO:0090503       RNA phosphodiester bond hydrolysis, exonucleolytic
##    Annotated Significant Expected    pval
## 1        115          68    49.34 1.9e-05
## 2       1087         436   466.39 3.3e-05
## 3        292         178   125.29 3.7e-05
## 4       1600         745   686.49 5.5e-05
## 5        390         175   167.33 0.00022
## 6         19          14     8.15 0.00045
## 7        509         271   218.39 0.00051
## 8         36          26    15.45 0.00054
## 9        899         431   385.72 0.00064
## 10        35          23    15.02 0.00069
```

```
## Obtain geneIDs
gene_id <- unique(rownames(counts_jx))

## Calculate p-values for genes with Simes' rule
p_jx_gene <- NULL
for(i in seq_len(length(gene_id))){
    p_jx <- res_jx$p.mod[rownames(counts_jx) %in% gene_id[i]]
    p_jx <- sort(p_jx)
    p_jx_simes <- NULL
    for(j in 1:length(p_jx)){
        p_jx_simes[j] <- length(p_jx) * p_jx[j] / j
    }
    p_jx_gene[i] <- min(p_jx_simes)
}
names(p_jx_gene) <- gene_id


## Determine the number of 'gene leveljunction' differentially expressed q < 0.05
q_jx_gene <- qvalue(p_jx_gene)$q
```

```
sum(q_jx_gene < 0.05)
```

```
## [1] 5366
```

```
## 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 junction p-values)
interesting <- function(x) x < 0.05

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

```
##
## Building most specific GOs .....
##  ( 8132 GO terms found. )
##
## Build GO DAG topology ..........
##  ( 12099 GO terms and 28473 relations. )
##
## Annotating nodes ...............
##  ( 6640 genes annotated to the GO terms. )
```

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

```
##
##           -- Weight01 Algorithm --
##
##       the algorithm is scoring 12099 nontrivial nodes
##       parameters:
##           test statistic: ks
##           score order: increasing
##
##   Level 20:  1 nodes to be scored    (0 eliminated genes)
##
##   Level 19:  5 nodes to be scored    (0 eliminated genes)
##
##   Level 18:  10 nodes to be scored   (1 eliminated genes)
##
##   Level 17:  23 nodes to be scored   (10 eliminated genes)
##
##   Level 16:  84 nodes to be scored   (30 eliminated genes)
##
##   Level 15:  179 nodes to be scored  (67 eliminated genes)
##
##   Level 14:  360 nodes to be scored  (222 eliminated genes)
```

```
##
##   Level 13:  625 nodes to be scored  (599 eliminated genes)
##
##   Level 12:  928 nodes to be scored  (1213 eliminated genes)
##
##   Level 11:  1214 nodes to be scored (2224 eliminated genes)
##
##   Level 10:  1565 nodes to be scored (3116 eliminated genes)
##
##   Level 9:   1701 nodes to be scored (4092 eliminated genes)
##
##   Level 8:   1663 nodes to be scored (4889 eliminated genes)
##
##   Level 7:   1557 nodes to be scored (5477 eliminated genes)
##
##   Level 6:   1149 nodes to be scored (5904 eliminated genes)
##
##   Level 5:   652 nodes to be scored  (6189 eliminated genes)
##
##   Level 4:   285 nodes to be scored  (6372 eliminated genes)
##
##   Level 3:   76 nodes to be scored   (6467 eliminated genes)
##
##   Level 2:   21 nodes to be scored   (6533 eliminated genes)
##
##   Level 1:   1 nodes to be scored    (6558 eliminated genes)
```

```
bptest
```

```
##
## Description: biological process
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 12099 GO terms scored: 51 terms with p < 0.01
## Annotation data:
##     Annotated genes: 6640
##     Significant genes: 4896
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 12099
```

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

```
##         GO.ID
## 1  GO:0006614
## 2  GO:0019083
## 3  GO:0000184
## 4  GO:0006413
```

```
## 5  GO:0098609
## 6  GO:0006364
## 7  GO:0006446
## 8  GO:0043517
## 9  GO:0075522
## 10 GO:0001649
##                                                                                 Term
## 1                           SRP-dependent cotranslational protein targeting to membrane
## 2                                                                    viral transcription
## 3                     nuclear-transcribed mRNA catabolic process, nonsense-mediated decay
## 4                                                              translational initiation
## 5                                                                    cell-cell adhesion
## 6                                                                       rRNA processing
## 7                                                 regulation of translational initiation
## 8  positive regulation of DNA damage response, signal transduction by p53 class mediator
## 9                                          IRES-dependent viral translational initiation
## 10                                                          osteoblast differentiation
##    Annotated Significant Expected    pval
## 1         81          72    59.73 1.7e-13
## 2        140         114   103.23 8.5e-13
## 3         99          85    73.00 1.1e-12
## 4        157         138   115.76 8.2e-12
## 5        554         447   408.49 7.0e-07
## 6        202         154   148.94 1.2e-05
## 7         67          59    49.40  0.0004
## 8          6           6     4.42  0.0012
## 9          7           7     5.16  0.0012
## 10        96          77    70.79  0.0015
```

```
## Load p-values from DER anaysis
load('AnnotatedDERs.Rdata')
p.mod <- annotatedDERs

## Obtain geneIDs
gene_id <- unique(names(p.mod))

## Calculate p-values for genes with Simes' rule
p_DER_gene <- NULL
for(i in seq_len(length(gene_id))){
    p_DER <- p.mod[names(p.mod) %in% gene_id[i]]
    p_DER <- sort(p_DER)
    p_DER_simes <- NULL
    for(j in 1:length(p_DER)){
        p_DER_simes[j] <- length(p_DER) * p_DER[j] / j
    }
    p_DER_gene[i] <- min(p_DER_simes)
}
names(p_DER_gene) <- gene_id

## Determine the number of 'gene level DERs' differentially expressed q < 0.05
q_DER_gene <- qvalue(p_DER_gene)$q
sum(q_DER_gene < 0.05)
```

```
## [1] 6463
```

```
## As above, 'topGO' can be utilized to assign biological function to
## differentially expressed DERs.

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

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

```
##
## Building most specific GOs .....
##  ( 9515 GO terms found. )
##
## Build GO DAG topology ..........
##  ( 13444 GO terms and 31792 relations. )
##
## Annotating nodes ...............
##  ( 9747 genes annotated to the GO terms. )
```

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

```
##
##           -- Weight01 Algorithm --
##
##       the algorithm is scoring 13444 nontrivial nodes
##       parameters:
##           test statistic: ks
##           score order: increasing
##
##   Level 20:  1 nodes to be scored    (0 eliminated genes)
##
##   Level 19:  6 nodes to be scored    (0 eliminated genes)
##
##   Level 18:  15 nodes to be scored   (1 eliminated genes)
##
##   Level 17:  36 nodes to be scored   (18 eliminated genes)
##
##   Level 16:  103 nodes to be scored  (49 eliminated genes)
##
##   Level 15:  209 nodes to be scored  (113 eliminated genes)
##
##   Level 14:  432 nodes to be scored  (341 eliminated genes)
##
##   Level 13:  739 nodes to be scored  (841 eliminated genes)
```

```
##
##   Level 12:  1104 nodes to be scored (1695 eliminated genes)
##
##   Level 11:  1399 nodes to be scored (3175 eliminated genes)
##
##   Level 10:  1767 nodes to be scored (4447 eliminated genes)
##
##   Level 9:   1880 nodes to be scored (5959 eliminated genes)
##
##   Level 8:   1790 nodes to be scored (7126 eliminated genes)
##
##   Level 7:   1659 nodes to be scored (7972 eliminated genes)
##
##   Level 6:   1229 nodes to be scored (8589 eliminated genes)
##
##   Level 5:   683 nodes to be scored  (9023 eliminated genes)
##
##   Level 4:   294 nodes to be scored  (9311 eliminated genes)
##
##   Level 3:   76 nodes to be scored   (9451 eliminated genes)
##
##   Level 2:   21 nodes to be scored   (9565 eliminated genes)
##
##   Level 1:   1 nodes to be scored    (9613 eliminated genes)
```

```
bptest
```

```
##
## Description: biological process
## Ontology: BP
## 'weight01' algorithm with the 'ks' test
## 13444 GO terms scored: 47 terms with p < 0.01
## Annotation data:
##     Annotated genes: 9747
##     Significant genes: 4322
##     Min. no. of genes annotated to a GO: 1
##     Nontrivial nodes: 13444
```

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

```
##         GO.ID                                                  Term
## 1  GO:0000398                          mRNA splicing, via spliceosome
## 2  GO:0090503       RNA phosphodiester bond hydrolysis, exonucleolytic
## 3  GO:0048025    negative regulation of mRNA splicing, via spliceosome
## 4  GO:0051493                 regulation of cytoskeleton organization
## 5  GO:0000244                 spliceosomal tri-snRNP complex assembly
## 6  GO:0042795      snRNA transcription from RNA polymerase II promoter
## 7  GO:0002026             regulation of the force of heart contraction
```

```
## 8  GO:0000381 regulation of alternative mRNA splicing, via spliceosome
## 9  GO:0031325        positive regulation of cellular metabolic process
## 10 GO:0010592            positive regulation of lamellipodium assembly
##    Annotated Significant Expected    pval
## 1        258         161   114.40 0.00013
## 2         32          22    14.19 0.00043
## 3         17          13     7.54 0.00064
## 4        299         150   132.58 0.00071
## 5         11           9     4.88 0.00114
## 6         67          40    29.71 0.00118
## 7         15          10     6.65 0.00121
## 8         26          17    11.53 0.00138
## 9       1854         838   822.10 0.00141
## 10        11          10     4.88 0.00150
```

To determine the concordance between the gene level and (exon, junction, DER) 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.

```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2, 3)]

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

## Obtain and sort p-values for genes derived from exons
p.mod2 <- p_exon_gene
p.mod2.sort <- p.mod2[order(p.mod2)]

## Obtain and sort p-values for genes derived from junctions
p.mod3 <- p_jx_gene
p.mod3.sort <- p.mod3[order(p.mod3)]

## Obtain and sort p-values for genes derived from DER
p.mod4 <- p_DER_gene
p.mod4.sort <- p.mod4[order(p.mod4)]


## Overlap of features:
    ## gene level and exon level
table(names(p.mod1.sort) %in% names(p.mod2.sort))
```

```
##
##  TRUE
## 26742
```

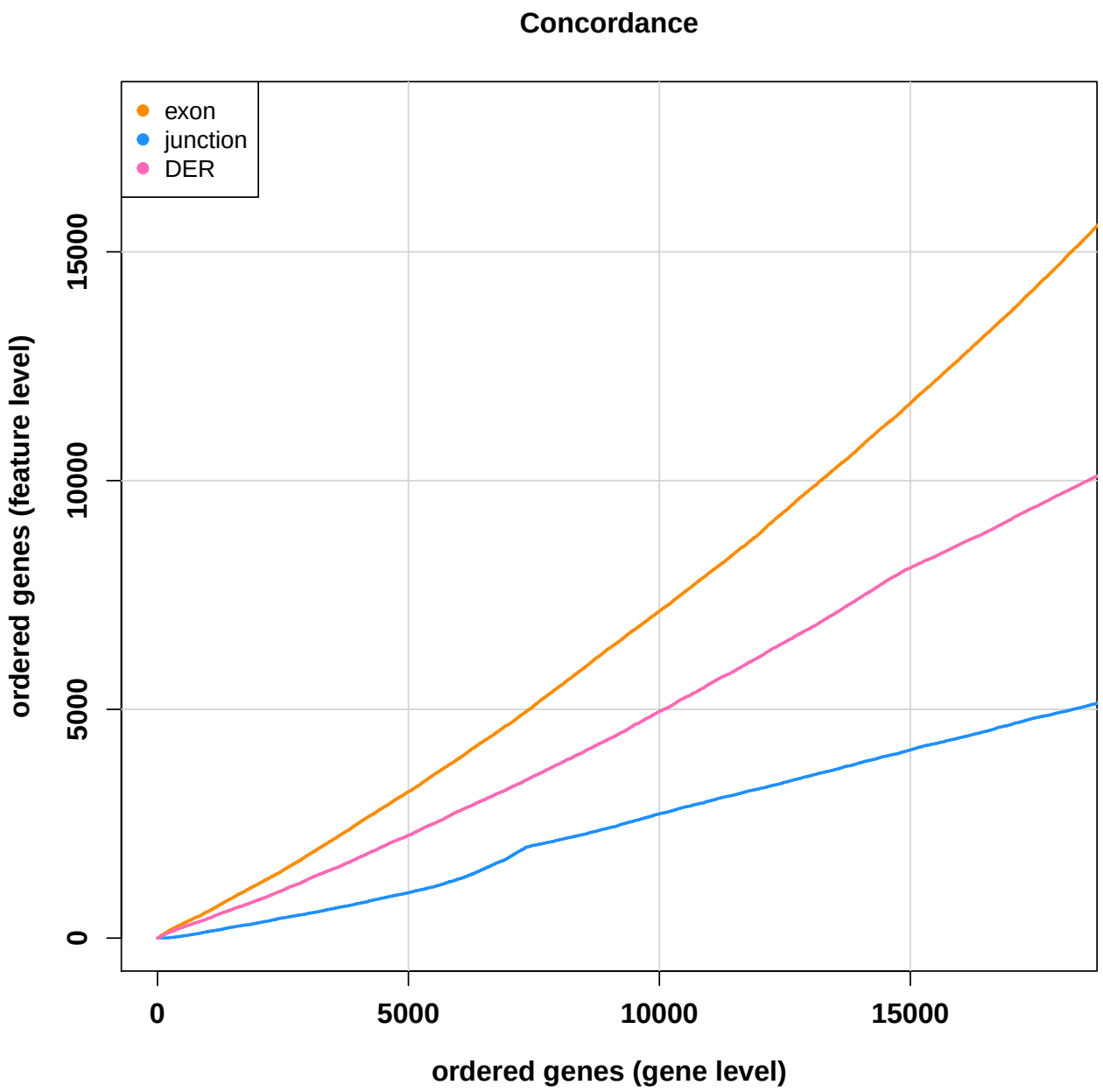
```
    ## gene level and junction level
table(names(p.mod1.sort) %in% names(p.mod3.sort))
```

```
##
## FALSE  TRUE
## 19406  7336
```

```
    ## gene level and DER level
table(names(p.mod1.sort) %in% names(p.mod4.sort))
```

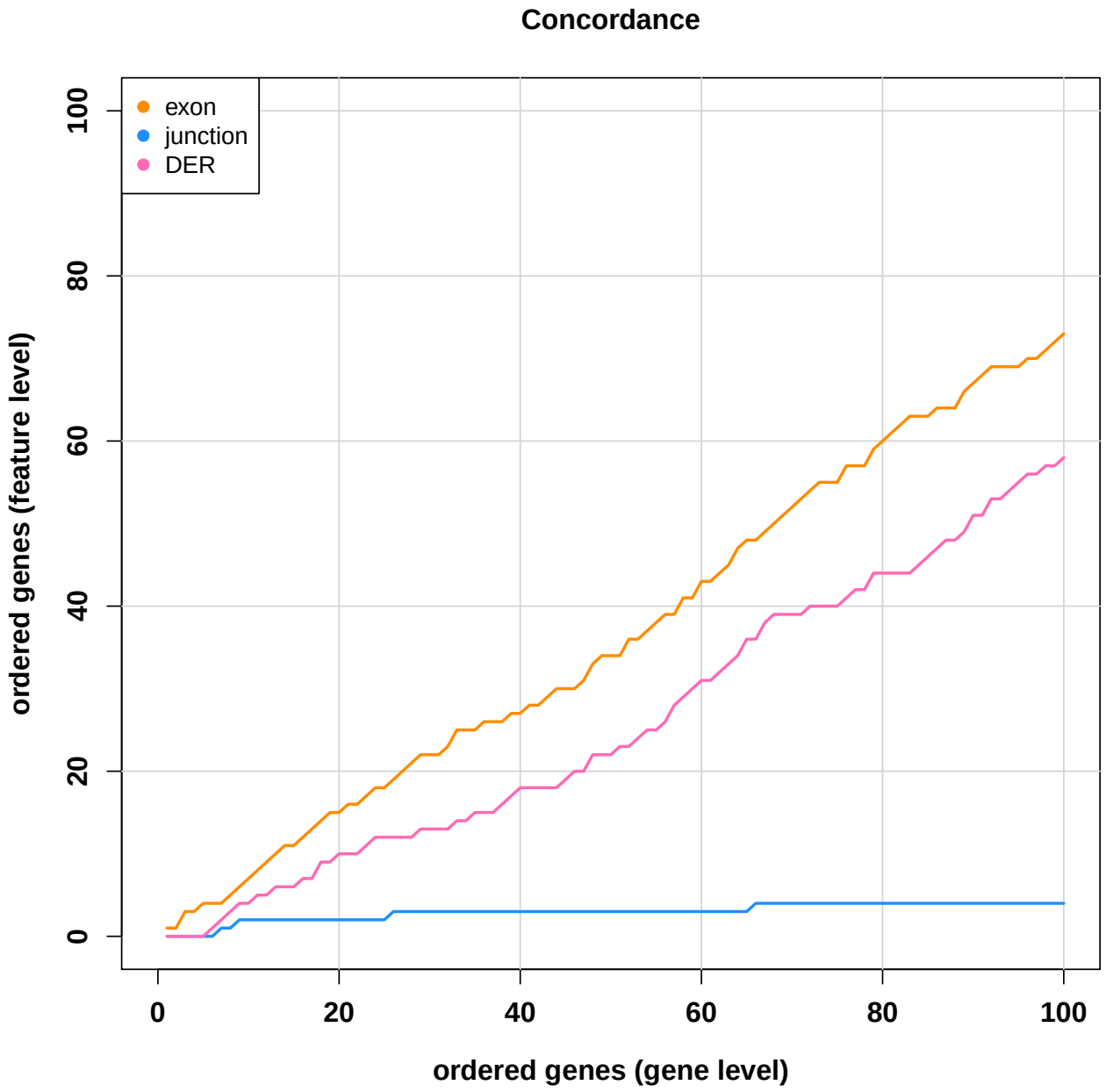
```
##
## FALSE  TRUE
## 12229 14513
```

```
conc_exon <- NULL
conc_jx <- NULL
conc_DER <- NULL
for(i in seq_len(length(p.mod1.sort))) {
    conc_exon[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod2.sort)[1:i])
    conc_jx[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod3.sort)[1:i])
    conc_DER[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod4.sort)[1:i])
}


## All genes
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort)), conc_exon,
     type = 'l', las = 0,
     xlim = c(0, 18000),
     ylim = c(0, 18000),
     xlab = 'ordered genes (gene level)',
     ylab = 'ordered genes (feature level)',
     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')
}
points(seq(1:length(p.mod1.sort)), conc_jx, type = 'l', lwd = 2, col = trop[2])
lines(seq(1:length(p.mod1.sort)), conc_exon, lwd = 2,  col = trop[1])
lines(seq(1:length(p.mod1.sort)), conc_DER, lwd = 2,  col = trop[3])
legend('topleft', pch = 19, col = trop[c(1, 2, 3)], c("exon", "junction", "DER"), bg="white")
```




```
## Top 100 genes
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort[1:100])), conc_exon[1:100],
     type = 'l',
     xlim = c(0, 100),
     ylim = c(0, 100),
     xlab = 'ordered genes (gene level)',
     ylab = 'ordered genes (feature level)',
     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')
}
points(seq(1:length(p.mod1.sort[1:100])), conc_jx[1:100], type = 'l', lwd = 2, col = trop[2])
lines(seq(1:length(p.mod1.sort[1:100])), conc_exon[1:100], lwd = 2,  col = trop[1])
```

```
lines(seq(1:length(p.mod1.sort[1:100])), conc_DER[1:100], lwd = 2,  col = trop[3])
legend('topleft', pch = 19, col = trop[c(1, 2, 3)], c("exon", "junction", "DER"), bg="white")
```




```
## Numbers at 100 on the x-axis
conc_jx[100]
```

```
## [1] 4
```

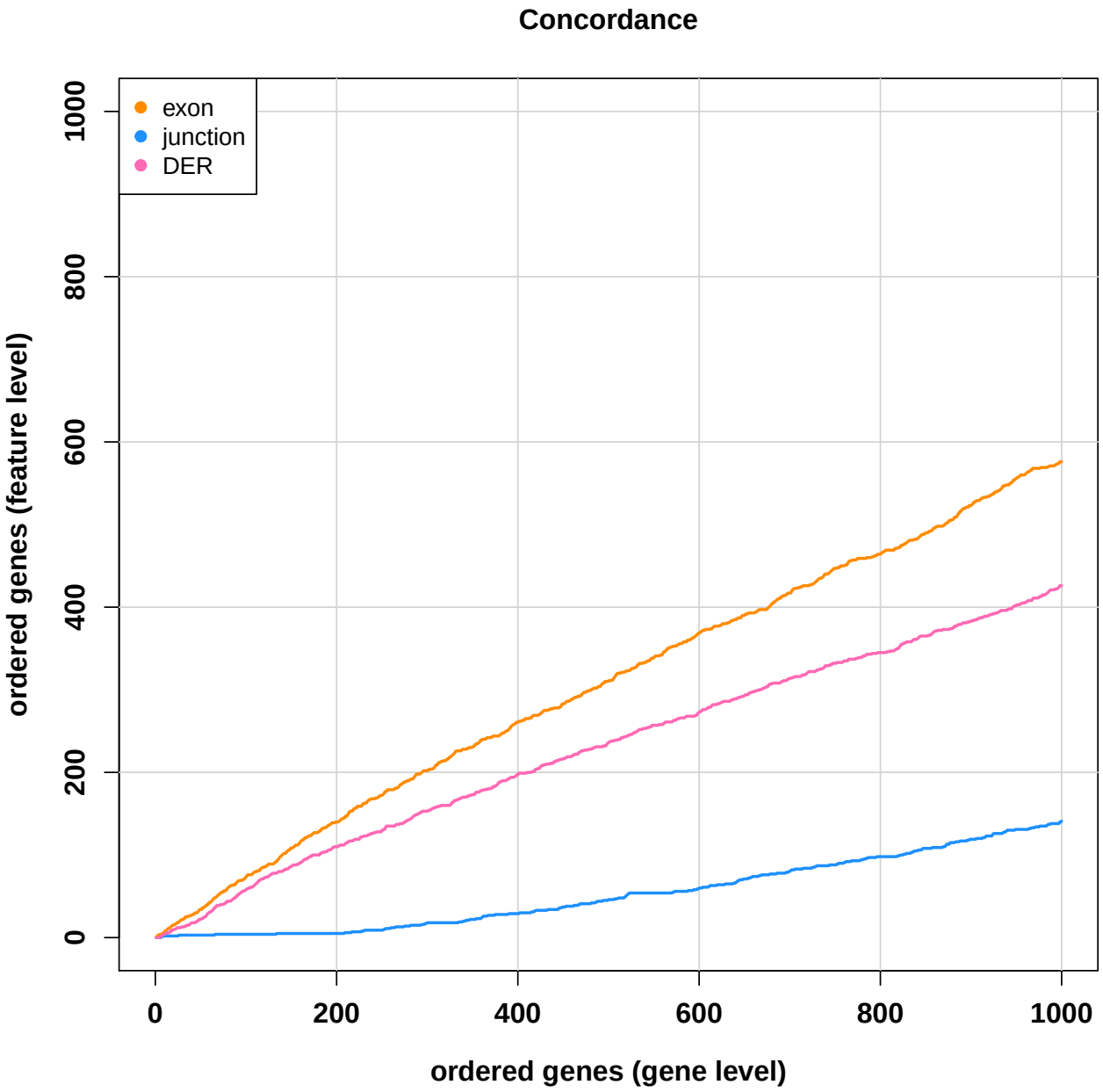
```
conc_exon[100]
```

```
## [1] 73
```

```
conc_DER[100]
```

```
## [1] 58
```

```
## Top 1,000 genes
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod1.sort[1:1000])), conc_exon[1:1000],
     type = 'l',
     xlim = c(0, 1000),
     ylim = c(0, 1000),
     xlab = 'ordered genes (gene level)',
     ylab = 'ordered genes (feature level)',
     main = 'Concordance')
for(k in 1:5){
    abline(v = k * 200, cex = 0.5, col = 'lightgrey')
    abline(h = k * 200, cex = 0.5, col = 'lightgrey')
}
points(seq(1:length(p.mod1.sort[1:1000])), conc_jx[1:1000], type = 'l', lwd = 2, col = trop[2])
lines(seq(1:length(p.mod1.sort[1:1000])), conc_exon[1:1000], lwd = 2,  col = trop[1] )
lines(seq(1:length(p.mod1.sort[1:1000])), conc_DER[1:1000], lwd = 2,  col = trop[3])
legend('topleft', pch = 19, col = trop[c(1, 2, 3)], c("exon", "junction", "DER"), bg="white")
```



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.

6 Reproducibility

This analysis report was made possible thanks to:

- R (R Core Team, 2016)
- *BiocStyle* (Oleś, Morgan, and Huber, 2017)
- *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, Nellore, Kammers, Ellis, et al., 2016)
- *rmarkdown* (Allaire, Cheng, Xie, McPherson, et al., 2017)
- *RSkittleBrewer* (Frazee, 2017)
- *SummarizedExperiment* (Morgan, Obenchain, Hester, and Pagès, 2017)
- *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 2.27.0. 2016.

[2] J. Allaire, J. Cheng, Y. Xie, J. McPherson, et al. rmarkdown: Dynamic Documents for R. R package version 1.3. 2017. URL: http://rmarkdown.rstudio.com.

[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 2.7.0. 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 0.51.0. 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, A. Nellore, A. C. Frazee, C. Wilks, et al. "Flexible expressed region analysis for RNA-seq with derfinder". In: Nucl. Acids Res. (2016). DOI: 10.1093/nar/gkw852. URL: http://nar.oxfordjournals.org/content/early/2016/09/29/nar.gkw852.

[7] L. Collado-Torres, A. Nellore, K. Kammers, S. E. Ellis, et al. "recount: A large-scale resource of analysis-ready RNA-seq expression data". In: bioRxiv (2016). DOI: 10.1101/068478. URL: http://biorxiv.org/content/early/2016/08/08/068478.

[8] A. Frazee. RSkittleBrewer: Fun with R Colors. R package version 1.1. 2017. URL: https://github.com/alyssafrazee/RSkittlel

[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 1.5.6. 2017.

[11] A. Oleś, M. Morgan and W. Huber. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.3.30. 2017. 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 1.12.0. 2016. URL: https://CRAN.R-project.org/package=devtools.

```
## Time spent creating this report:
diff(c(timestart, Sys.time()))
```

```
## Time difference of 30.4198 mins
```

```
## Date this report was generated
message(Sys.time())
```

```
## 2017-02-14 18:32:11
```

```
## Reproducibility info
options(width = 120)
devtools::session_info()
```

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

```
##  setting  value
##  version  R Under development (unstable) (2016-10-26 r71594)
##  system   x86_64, darwin13.4.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2017-02-14
```

```
## Packages -----------------------------------------------------------------------------------------------
```

```
##  package              * version  date       source
##  acepack                1.4.1    2016-10-29 CRAN (R 3.4.0)
##  AnnotationDbi        * 1.37.3   2017-02-09 Bioconductor
##  assertthat             0.1      2013-12-06 CRAN (R 3.4.0)
##  backports              1.0.5    2017-01-18 CRAN (R 3.4.0)
##  base64enc              0.1-3    2015-07-28 CRAN (R 3.4.0)
##  bibtex                 0.4.0    2014-12-31 CRAN (R 3.4.0)
##  Biobase              * 2.35.0   2016-10-23 Bioconductor
##  BiocGenerics         * 0.21.3   2017-01-12 Bioconductor
##  BiocParallel           1.9.5    2017-01-24 Bioconductor
##  BiocStyle            * 2.3.30   2017-01-27 Bioconductor
##  biomaRt                2.31.4   2017-01-13 Bioconductor
##  Biostrings             2.43.4   2017-02-02 Bioconductor
##  bitops                 1.0-6    2013-08-17 CRAN (R 3.4.0)
##  BSgenome               1.43.5   2017-02-02 Bioconductor
##  bumphunter             1.15.0   2016-10-23 Bioconductor
##  checkmate              1.8.2    2016-11-02 CRAN (R 3.4.0)
##  cluster                2.0.5    2016-10-08 CRAN (R 3.4.0)
##  codetools              0.2-15   2016-10-05 CRAN (R 3.4.0)
##  colorout             * 1.1-2    2016-11-15 Github (jalvesaq/colorout@6d84420)
##  colorspace             1.3-2    2016-12-14 CRAN (R 3.4.0)
##  data.table             1.10.4   2017-02-01 CRAN (R 3.4.0)
##  DBI                    0.5-1    2016-09-10 CRAN (R 3.4.0)
##  DelayedArray         * 0.1.6    2017-02-10 cran (@0.1.6)
##  derfinder            * 1.9.6    2017-01-13 Bioconductor
##  derfinderHelper        1.9.3    2016-11-29 Bioconductor
##  devtools               1.12.0   2016-12-05 CRAN (R 3.4.0)
##  digest                 0.6.12   2017-01-27 CRAN (R 3.4.0)
##  doRNG                  1.6      2014-03-07 CRAN (R 3.4.0)
##  downloader             0.4      2015-07-09 CRAN (R 3.4.0)
##  edgeR                * 3.17.5   2016-12-13 Bioconductor
##  evaluate               0.10     2016-10-11 CRAN (R 3.4.0)
##  foreach                1.4.3    2015-10-13 CRAN (R 3.4.0)
##  foreign                0.8-67   2016-09-13 CRAN (R 3.4.0)
##  Formula                1.2-1    2015-04-07 CRAN (R 3.4.0)
##  GenomeInfoDb         * 1.11.9   2017-02-08 Bioconductor
##  GenomeInfoDbData       0.99.0   2017-02-14 Bioconductor
##  GenomicAlignments      1.11.9   2017-02-02 Bioconductor
```

```
##  GenomicFeatures       1.27.6   2016-12-17 Bioconductor
##  GenomicFiles          1.11.3   2016-11-29 Bioconductor
##  GenomicRanges       * 1.27.22  2017-02-02 Bioconductor
##  GEOquery              2.41.0   2016-10-25 Bioconductor
##  ggplot2               2.2.1    2016-12-30 CRAN (R 3.4.0)
##  GO.db               * 3.4.0    2016-11-15 Bioconductor
##  graph               * 1.53.0   2016-10-23 Bioconductor
##  gridExtra             2.2.1    2016-02-29 CRAN (R 3.4.0)
##  gtable                0.2.0    2016-02-26 CRAN (R 3.4.0)
##  Hmisc                 4.0-2    2016-12-31 CRAN (R 3.4.0)
##  htmlTable             1.9      2017-01-26 CRAN (R 3.4.0)
##  htmltools             0.3.5    2016-03-21 CRAN (R 3.4.0)
##  htmlwidgets           0.8      2016-11-09 CRAN (R 3.4.0)
##  httr                  1.2.1    2016-07-03 CRAN (R 3.4.0)
##  IRanges             * 2.9.18   2017-02-02 Bioconductor
##  iterators             1.0.8    2015-10-13 CRAN (R 3.4.0)
##  jsonlite              1.2      2016-12-31 CRAN (R 3.4.0)
##  knitcitations       * 1.0.7    2015-10-28 CRAN (R 3.4.0)
##  knitr                 1.15.1   2016-11-22 CRAN (R 3.4.0)
##  lattice               0.20-34  2016-09-06 CRAN (R 3.4.0)
##  latticeExtra          0.6-28   2016-02-09 CRAN (R 3.4.0)
##  lazyeval              0.2.0    2016-06-12 CRAN (R 3.4.0)
##  limma               * 3.31.13  2017-02-09 Bioconductor
##  locfit                1.5-9.1  2013-04-20 CRAN (R 3.4.0)
##  lubridate             1.6.0    2016-09-13 CRAN (R 3.4.0)
##  magrittr              1.5      2014-11-22 CRAN (R 3.4.0)
##  Matrix                1.2-8    2017-01-20 CRAN (R 3.4.0)
##  matrixStats         * 0.51.0   2016-10-09 CRAN (R 3.4.0)
##  memoise               1.0.0    2016-01-29 CRAN (R 3.4.0)
##  munsell               0.4.3    2016-02-13 CRAN (R 3.4.0)
##  nnet                  7.3-12   2016-02-02 CRAN (R 3.4.0)
##  org.Hs.eg.db        * 3.4.0    2016-11-15 Bioconductor
##  pkgmaker              0.22     2014-05-14 CRAN (R 3.4.0)
##  plyr                  1.8.4    2016-06-08 CRAN (R 3.4.0)
##  qvalue              * 2.7.0    2016-10-23 Bioconductor
##  R6                    2.2.0    2016-10-05 CRAN (R 3.4.0)
##  RColorBrewer          1.1-2    2014-12-07 CRAN (R 3.4.0)
##  Rcpp                  0.12.9   2017-01-14 CRAN (R 3.4.0)
##  RCurl                 1.95-4.8 2016-03-01 CRAN (R 3.4.0)
##  recount             * 1.1.17   2017-02-14 Github (leekgroup/recount@fa5f3ea)
##  RefManageR            0.13.1   2016-11-13 CRAN (R 3.4.0)
##  registry              0.3      2015-07-08 CRAN (R 3.4.0)
##  rentrez               1.0.4    2016-10-26 CRAN (R 3.4.0)
##  reshape2              1.4.2    2016-10-22 CRAN (R 3.4.0)
##  RJSONIO               1.3-0    2014-07-28 CRAN (R 3.4.0)
##  rmarkdown           * 1.3      2017-01-20 Github (rstudio/rmarkdown@5b74148)
##  rngtools              1.2.4    2014-03-06 CRAN (R 3.4.0)
##  rpart                 4.1-10   2015-06-29 CRAN (R 3.4.0)
##  rprojroot             1.2      2017-01-16 CRAN (R 3.4.0)
##  Rsamtools             1.27.12  2017-01-24 Bioconductor
##  RSkittleBrewer      * 1.1      2016-11-15 Github (alyssafrazee/RSkittleBrewer@0088112)
##  RSQLite               1.1-2    2017-01-08 CRAN (R 3.4.0)
##  rtracklayer           1.35.5   2017-02-02 Bioconductor
##  S4Vectors           * 0.13.15  2017-02-14 cran (@0.13.15)
```

```
##  scales                 0.4.1    2016-11-09 CRAN (R 3.4.0)
##  SparseM              * 1.74     2016-11-10 CRAN (R 3.4.0)
##  stringi                1.1.2    2016-10-01 CRAN (R 3.4.0)
##  stringr                1.1.0    2016-08-19 CRAN (R 3.4.0)
##  SummarizedExperiment * 1.5.6    2017-02-10 cran (@1.5.6)
##  survival               2.40-1   2016-10-30 CRAN (R 3.4.0)
##  tibble                 1.2      2016-08-26 CRAN (R 3.4.0)
##  topGO                * 2.27.0   2016-10-23 Bioconductor
##  VariantAnnotation      1.21.17  2017-02-12 Bioconductor
##  withr                  1.0.2    2016-06-20 CRAN (R 3.4.0)
##  XML                    3.98-1.5 2016-11-10 CRAN (R 3.4.0)
##  xtable                 1.8-2    2016-02-05 CRAN (R 3.4.0)
##  XVector                0.15.2   2017-02-02 Bioconductor
##  yaml                   2.1.14   2016-11-12 CRAN (R 3.4.0)
##  zlibbioc               1.21.0   2016-10-23 Bioconductor
```