recount (overlay two studies)

Kai Kammers and Shannon Ellis

July 22, 2016

Contents

In these analyses, we compare differential expression findings from two different studies, each of which looked to compare the transcriptomes of human breast cancer samples.

Data herein are labeled as follows:

- study1 = SRP019936 (This is the 'new' study)
- study2 = SRP032798 (This is the 'reference' study that is also used for gene, exon, junction, and differential expressed region (DER) analyses)

Load R-packages

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

We first download data for the project of interest (SRP019936), obtaining expression data. Data can be summarized across samples and genes using `colData()` and `rowData()`, respectively.

Gene level analysis

```
## Find the project of interest (SRP019936), e.g. with parts of the abstract
project_info1 <- abstract_search('model for HER2 positive breast tumors')
project_info1
```

```
##     number_samples species
## 640             32   human
##
## 640 The goal of our study is to build an integrated transcriptome landscape model for HER2 positive
##       project
## 640 SRP019936
```

```
## Download the gene-level RangedSummarizedExperiment data
if(!file.exists(file.path('SRP019936', 'rse_gene.Rdata'))) {
    download_study(project_info1$project)
}

## Load the data
load(file.path(project_info1$project, 'rse_gene.Rdata'))
rse_gene1 <- rse_gene

## Browse the project at SRA
browse_study(project_info1$project)

## This is the sample phenotype data provided by the recount project
colData(rse_gene1)
```

```
## DataFrame with 32 rows and 21 columns
##               project      sample  experiment         run
##           <character> <character> <character> <character>
## SRR791043   SRP019936   SRS403393   SRX254189   SRR791043
## SRR791044   SRP019936   SRS403394   SRX254190   SRR791044
## SRR791045   SRP019936   SRS403395   SRX254191   SRR791045
## SRR791046   SRP019936   SRS403396   SRX254192   SRR791046
## SRR791047   SRP019936   SRS403397   SRX254193   SRR791047
## ...               ...         ...         ...         ...
## SRR791070   SRP019936   SRS403420   SRX254216   SRR791070
## SRR791071   SRP019936   SRS403421   SRX254217   SRR791071
## SRR791072   SRP019936   SRS403422   SRX254218   SRR791072
## SRR791073   SRP019936   SRS403423   SRX254219   SRR791073
## SRR791074   SRP019936   SRS403424   SRX254220   SRR791074
##           read_count_as_reported_by_sra reads_downloaded
##                               <integer>        <integer>
## SRR791043                     105978126        105978126
## SRR791044                      95333634         95333634
## SRR791045                      99628684         99628684
## SRR791046                     101469302        101469302
## SRR791047                     105908146        105908146
## ...                                 ...              ...
## SRR791070                      48067496         48067496
## SRR791071                      43365202         43365202
## SRR791072                      40514396         40514396
## SRR791073                      43101642         43101642
## SRR791074                      57371422         57371422
##           proportion_of_reads_reported_by_sra_downloaded paired_end
##                                                <numeric>  <logical>
## SRR791043                                              1       TRUE
## SRR791044                                              1       TRUE
## SRR791045                                              1       TRUE
## SRR791046                                              1       TRUE
```

```
## SRR791047                                          1       TRUE
## ...                                              ...        ...
## SRR791070                                          1       TRUE
## SRR791071                                          1       TRUE
## SRR791072                                          1       TRUE
## SRR791073                                          1       TRUE
## SRR791074                                          1       TRUE
##           sra_misreported_paired_end mapped_read_count        auc
##                            <logical>         <integer>  <numeric>
## SRR791043                      FALSE         104178189 5176960114
## SRR791044                      FALSE          93149787 4618143004
## SRR791045                      FALSE          94582133 4693785750
## SRR791046                      FALSE          91480736 4532961517
## SRR791047                      FALSE         103567907 5140850573
## ...                              ...               ...        ...
## SRR791070                      FALSE          47433080 2360019131
## SRR791071                      FALSE          42745233 2119904288
## SRR791072                      FALSE          39881890 1980303339
## SRR791073                      FALSE          42431338 2108493246
## SRR791074                      FALSE          55132710 2714772264
##           sharq_beta_tissue sharq_beta_cell_type biosample_submission_date
##                 <character>          <character>                <character>
## SRR791043           breast                  esc   2013-03-22T11:37:31.457
## SRR791044           breast                  esc   2013-03-22T11:37:31.533
## SRR791045           breast                  esc   2013-03-22T11:37:31.583
## SRR791046           breast                  esc   2013-03-22T11:37:31.623
## SRR791047           breast                  esc   2013-03-22T11:37:31.663
## ...                    ...                  ...                       ...
## SRR791070           breast                  esc   2013-03-22T11:37:32.697
## SRR791071           breast                  esc   2013-03-22T11:37:32.747
## SRR791072           breast                  esc   2013-03-22T11:37:32.783
## SRR791073           breast                  esc   2013-03-22T11:37:32.817
## SRR791074           breast                  esc   2013-03-22T11:37:32.853
##           biosample_publication_date   biosample_update_date
##                          <character>             <character>
## SRR791043   2013-12-07T01:12:55.003 2014-03-06T17:06:22.413
## SRR791044   2013-12-07T01:12:57.767 2014-03-06T17:06:22.445
## SRR791045   2013-12-07T01:13:02.953 2014-03-06T17:06:22.483
## SRR791046   2013-12-07T01:13:00.473 2014-03-06T17:06:22.515
## SRR791047   2013-12-07T01:18:43.917 2014-03-06T17:06:22.546
## ...                            ...                     ...
## SRR791070   2013-12-07T01:18:31.917 2014-03-06T17:06:23.633
## SRR791071   2013-12-07T01:18:27.567 2014-03-06T17:06:23.664
## SRR791072   2013-12-07T01:18:26.017 2014-03-06T17:06:23.696
## SRR791073   2013-12-07T01:18:39.517 2014-03-06T17:06:23.728
## SRR791074   2013-12-07T01:18:28.853 2014-03-06T17:06:23.765
##           avg_read_length geo_accession  bigwig_file       title
##                 <integer>   <character>  <character> <character>
## SRR791043             100    GSM1103987 SRR791043.bw  BCT04_mRNA
## SRR791044             100    GSM1103988 SRR791044.bw  BCT12_mRNA
## SRR791045             100    GSM1103989 SRR791045.bw  BCT14_mRNA
## SRR791046             100    GSM1103990 SRR791046.bw  BCT16_mRNA
## SRR791047             100    GSM1103991 SRR791047.bw  BCT22_mRNA
## ...                   ...           ...          ...         ...
```

```
## SRR791070             100    GSM1104014 SRR791070.bw BSO32N_mRNA
## SRR791071             100    GSM1104015 SRR791071.bw  BSO36_mRNA
## SRR791072             100    GSM1104016 SRR791072.bw  BSO37_mRNA
## SRR791073             100    GSM1104017 SRR791073.bw DHF168_mRNA
## SRR791074             100    GSM1104018 SRR791074.bw OP-535_mRNA
##                                      characteristics
##                                      <CharacterList>
## SRR791043             tissue type: ER+ Breast Tumor
## SRR791044             tissue type: ER+ Breast Tumor
## SRR791045             tissue type: ER+ Breast Tumor
## SRR791046             tissue type: ER+ Breast Tumor
## SRR791047             tissue type: ER+ Breast Tumor
## ...                                             ...
## SRR791070     tissue type: Benign cell lines (HMEC)
## SRR791071     tissue type: Benign cell lines (HMEC)
## SRR791072     tissue type: Benign cell lines (HMEC)
## SRR791073     tissue type: Benign cell lines (HMEC)
## SRR791074 tissue type: Triple Negative Breast Tumor
```

```
## Gene info
rowData(rse_gene1)
```

```
## 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
rse1 <- scale_counts(rse_gene1)

## Download pheno data from
## http://trace.ncbi.nlm.nih.gov/Traces/study/?acc=SRP019936
pheno1 <- read.table('SraRunTable_SRP019936.txt', sep = '\t',
    header=TRUE,
    stringsAsFactors = FALSE)

## Obtain correct order for pheno data
pheno1 <- pheno1[match(rse1$run, pheno1$Run_s), ]
identical(pheno1$Run_s, rse1$run)
```

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

```
head(cbind(pheno1$Run_s, rse1$run))
```

```
##      [,1]        [,2]
## [1,] "SRR791043" "SRR791043"
## [2,] "SRR791044" "SRR791044"
## [3,] "SRR791045" "SRR791045"
## [4,] "SRR791046" "SRR791046"
## [5,] "SRR791047" "SRR791047"
## [6,] "SRR791048" "SRR791048"
```

```
## Obtain grouping information
colData(rse1)$group <- pheno1$tissue_s
table(colData(rse1)$group)
```

```
##
##     Benign cell lines (HMEC)             ER+ Breast Tumor
##                            8                            8
##           HER2+ Breast Tumor Triple Negative Breast Tumor
##                            8                            8
```

```
## subset data to HER2 and TNBC types
rse1 <- rse1[, rse1$group %in% c('HER2+ Breast Tumor', 'Triple Negative Breast Tumor')]
rse1
```

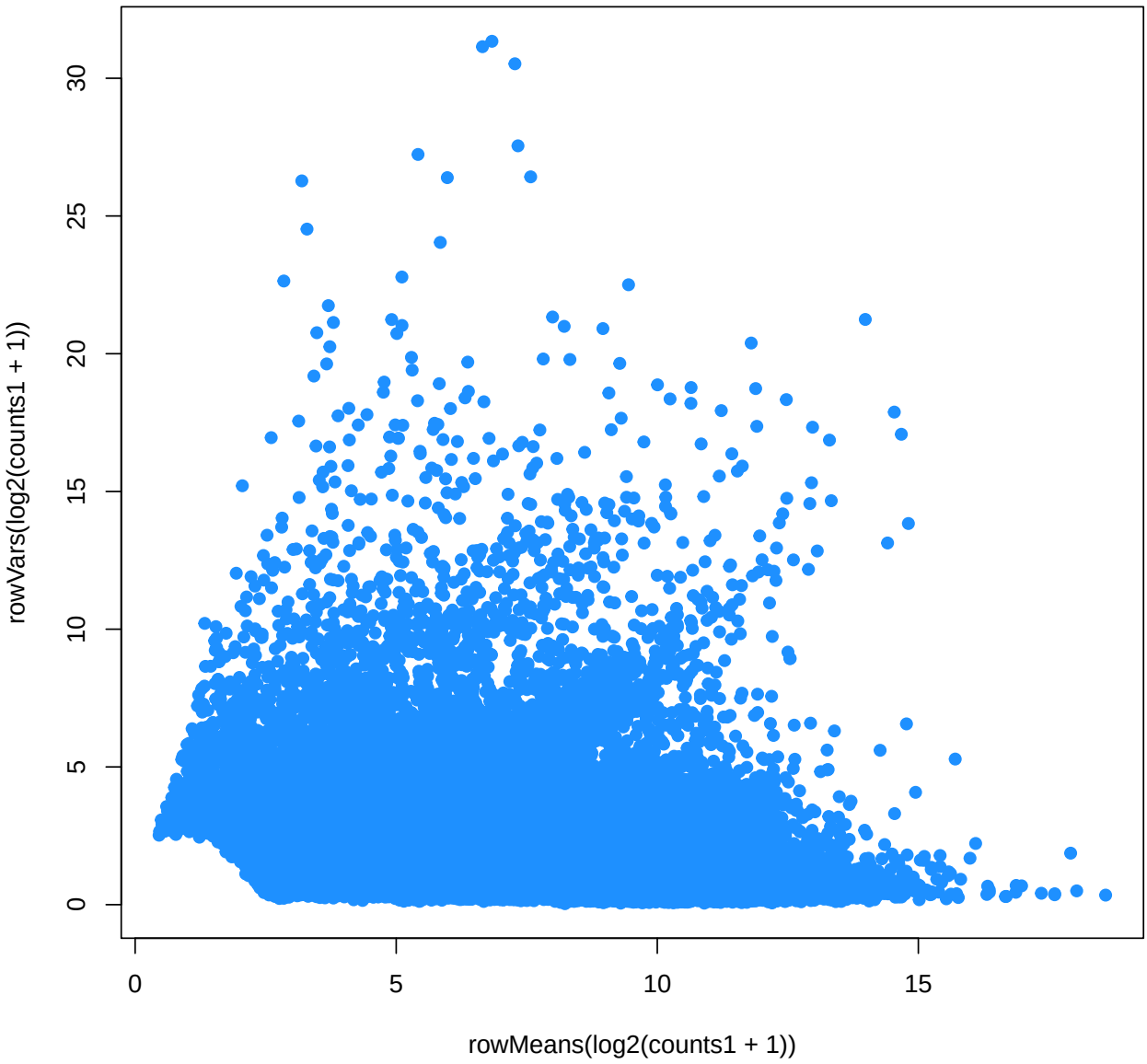
```
## class: RangedSummarizedExperiment
## dim: 58037 16
## metadata(0):
## assays(1): counts
## rownames(58037): ENSG00000000003.14 ENSG00000000005.5 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(3): gene_id bp_length symbol
## colnames(16): SRR791051 SRR791052 ... SRR791065 SRR791074
## colData names(22): project sample ... characteristics group
```

```
## Obtain count matrix
counts1 <- assays(rse1)$counts

## Filter count matrix
filter <- apply(counts1, 1, function(x) mean(x) > 5)
counts1 <- counts1[filter, ]
dim(counts1)
```

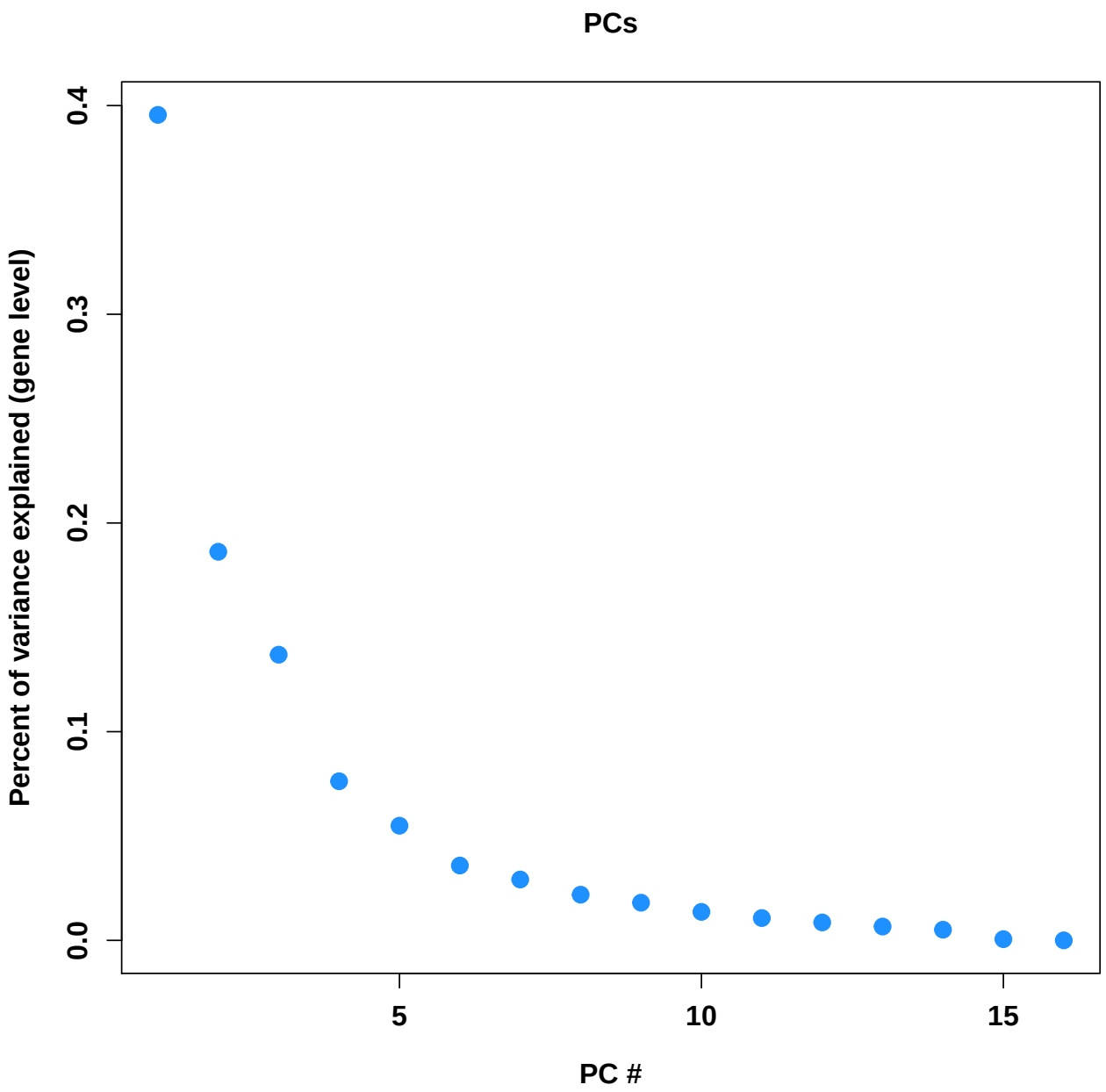
```
## [1] 29163    16
```

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.

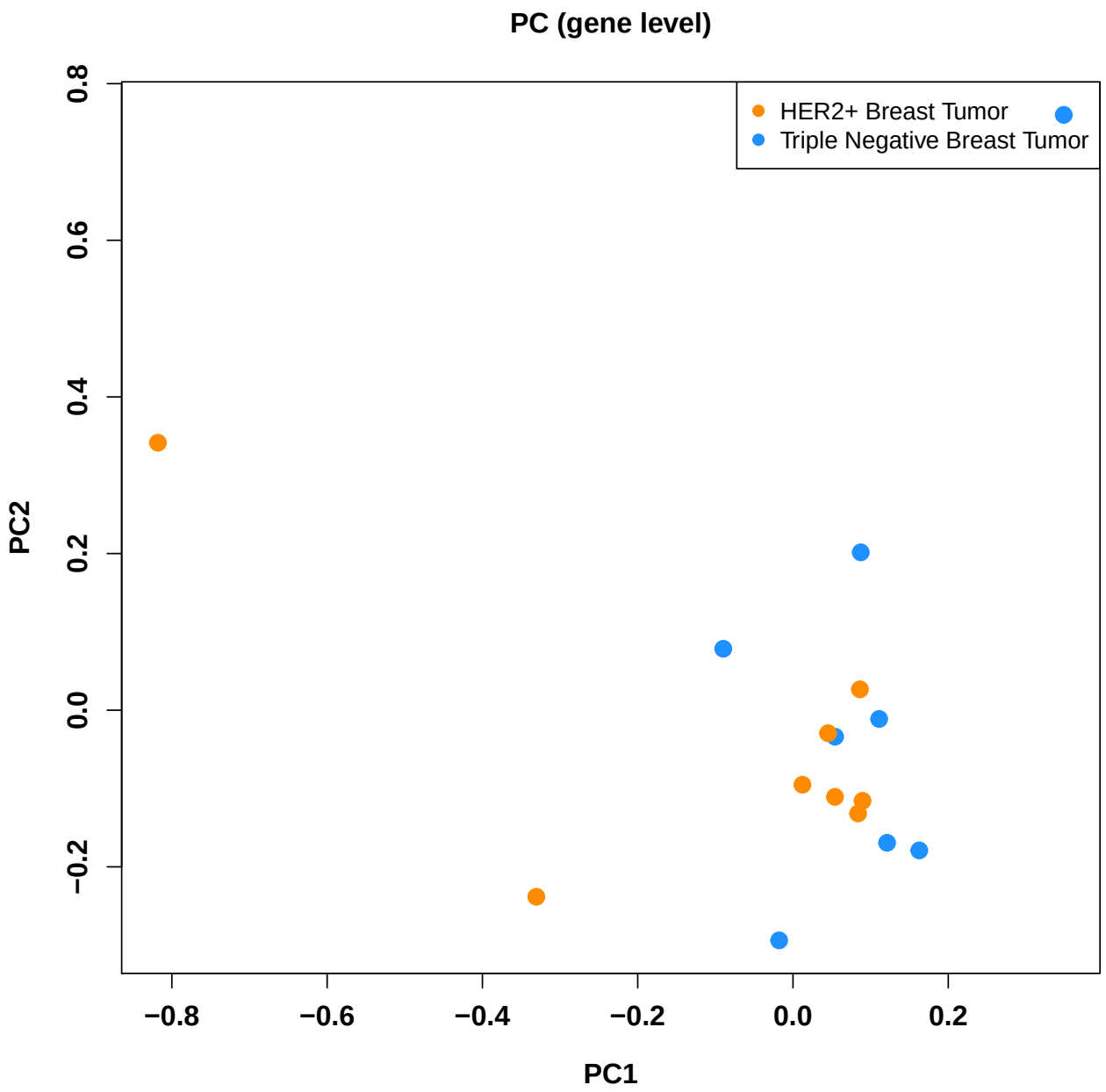
```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse1$group))

## Look at mean variance relationship
plot(rowMeans(log2(counts1 + 1)), rowVars(log2(counts1 + 1)),
     pch = 19, col = trop[2])
```




```
## Calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))

## 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 = 'Percent 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(rse1$group))))
```



PCA identifies a clear sample outlier in these data. This sample is removed from analysis prior to moving forward with differential expression analyses. As mentioned previously, prior to differential gene expression analysis, count data are 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
rse1 <- scale_counts(rse_gene1)

## Download pheno data from
## http://trace.ncbi.nlm.nih.gov/Traces/study/?acc=SRP019936
pheno1 <- read.table('SraRunTable_SRP019936.txt', sep = '\t',
    header=TRUE,
    stringsAsFactors = FALSE)

## Obtain correct order for pheno data
```

```
pheno1 <- pheno1[match(rse1$run, pheno1$Run_s), ]
identical(pheno1$Run_s, rse1$run)
```

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

```
head(cbind(pheno1$Run_s, rse1$run))
```

```
##      [,1]        [,2]
## [1,] "SRR791043" "SRR791043"
## [2,] "SRR791044" "SRR791044"
## [3,] "SRR791045" "SRR791045"
## [4,] "SRR791046" "SRR791046"
## [5,] "SRR791047" "SRR791047"
## [6,] "SRR791048" "SRR791048"
```

```
## Obtain grouping information
colData(rse1)$group <- pheno1$tissue_s
table(colData(rse1)$group)
```

```
##
##     Benign cell lines (HMEC)             ER+ Breast Tumor
##                            8                            8
##           HER2+ Breast Tumor Triple Negative Breast Tumor
##                            8                            8
```

```
## Subset data to HER2 and TNBC types
rse1 <- rse1[, rse1$group %in% c('HER2+ Breast Tumor', 'Triple Negative Breast Tumor')]
rse1
```

```
## class: RangedSummarizedExperiment
## dim: 58037 16
## metadata(0):
## assays(1): counts
## rownames(58037): ENSG00000000003.14 ENSG00000000005.5 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(3): gene_id bp_length symbol
## colnames(16): SRR791051 SRR791052 ... SRR791065 SRR791074
## colData names(22): project sample ... characteristics group
```

```
## Remove outlier sample
rse1 <- rse1[, -15]
rse1
```

```
## class: RangedSummarizedExperiment
## dim: 58037 15
## metadata(0):
## assays(1): counts
## rownames(58037): ENSG00000000003.14 ENSG00000000005.5 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(3): gene_id bp_length symbol
## colnames(15): SRR791051 SRR791052 ... SRR791064 SRR791074
## colData names(22): project sample ... characteristics group
```

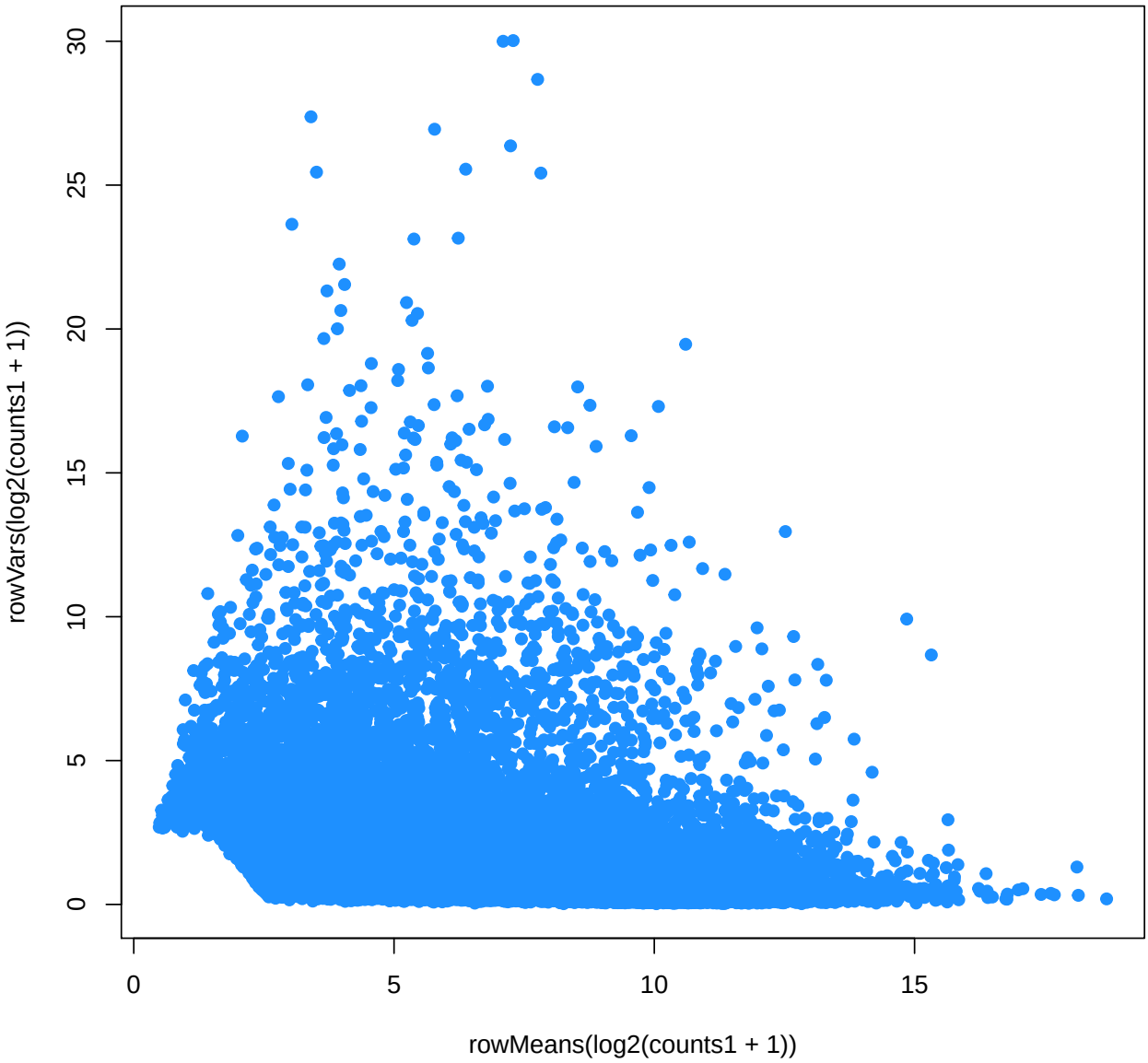
```
## Obtain count matrix
counts1 <- assays(rse1)$counts

## Filter count matrix
filter <- apply(counts1, 1, function(x) mean(x) > 5)
```

```
counts1 <- counts1[filter, ]
dim(counts1)
```

```
## [1] 29323    15
```

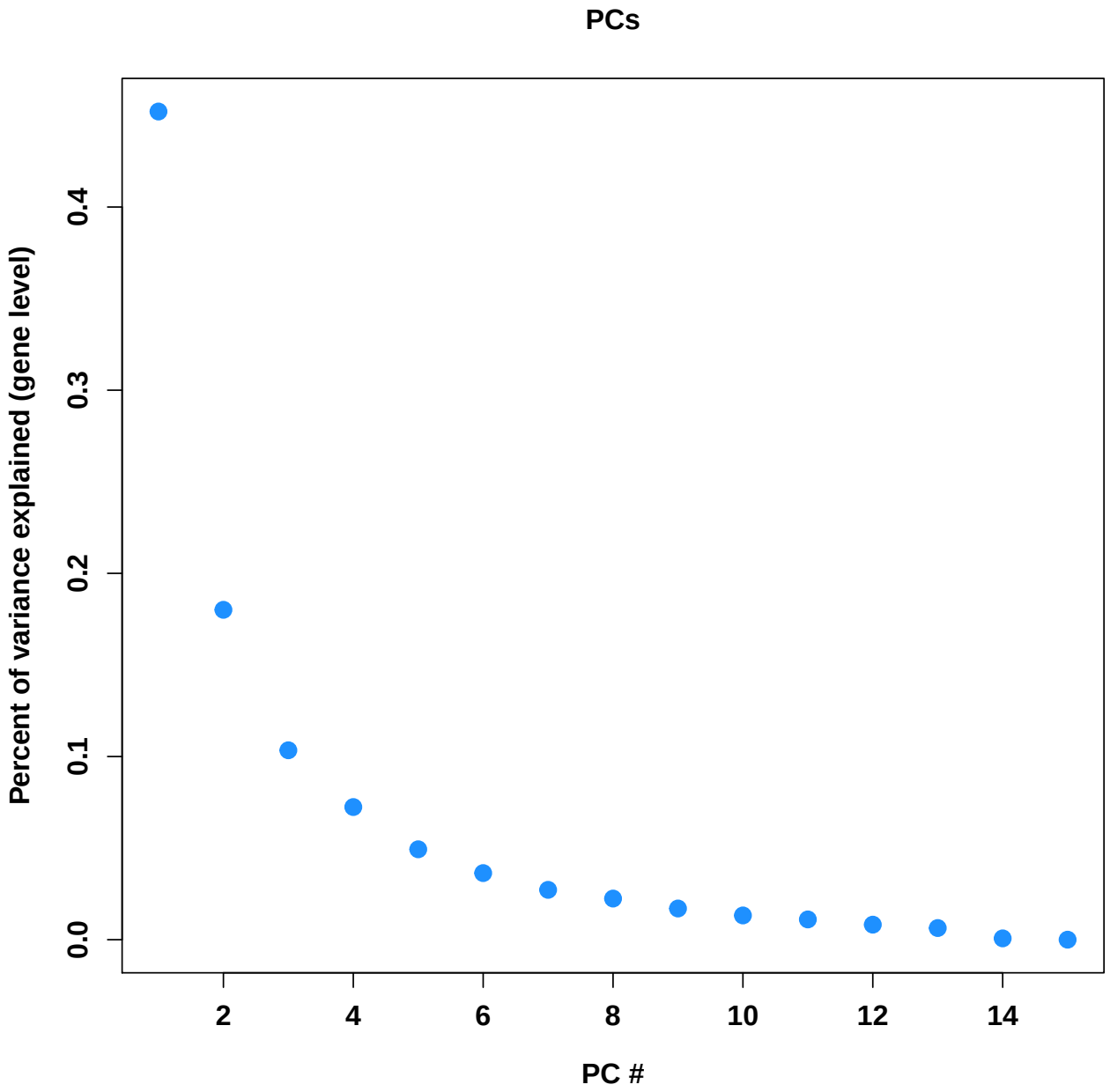
After sample outlier removal, PCA is again run to obtain a global understanding of the mean-variance relationship at each gene and the global relationship between samples included for study.

```
## Set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse1$group))

## Look at mean variance relationship
plot(rowMeans(log2(counts1 + 1)), rowVars(log2(counts1 + 1)),
     pch = 19, col = trop[2])
```

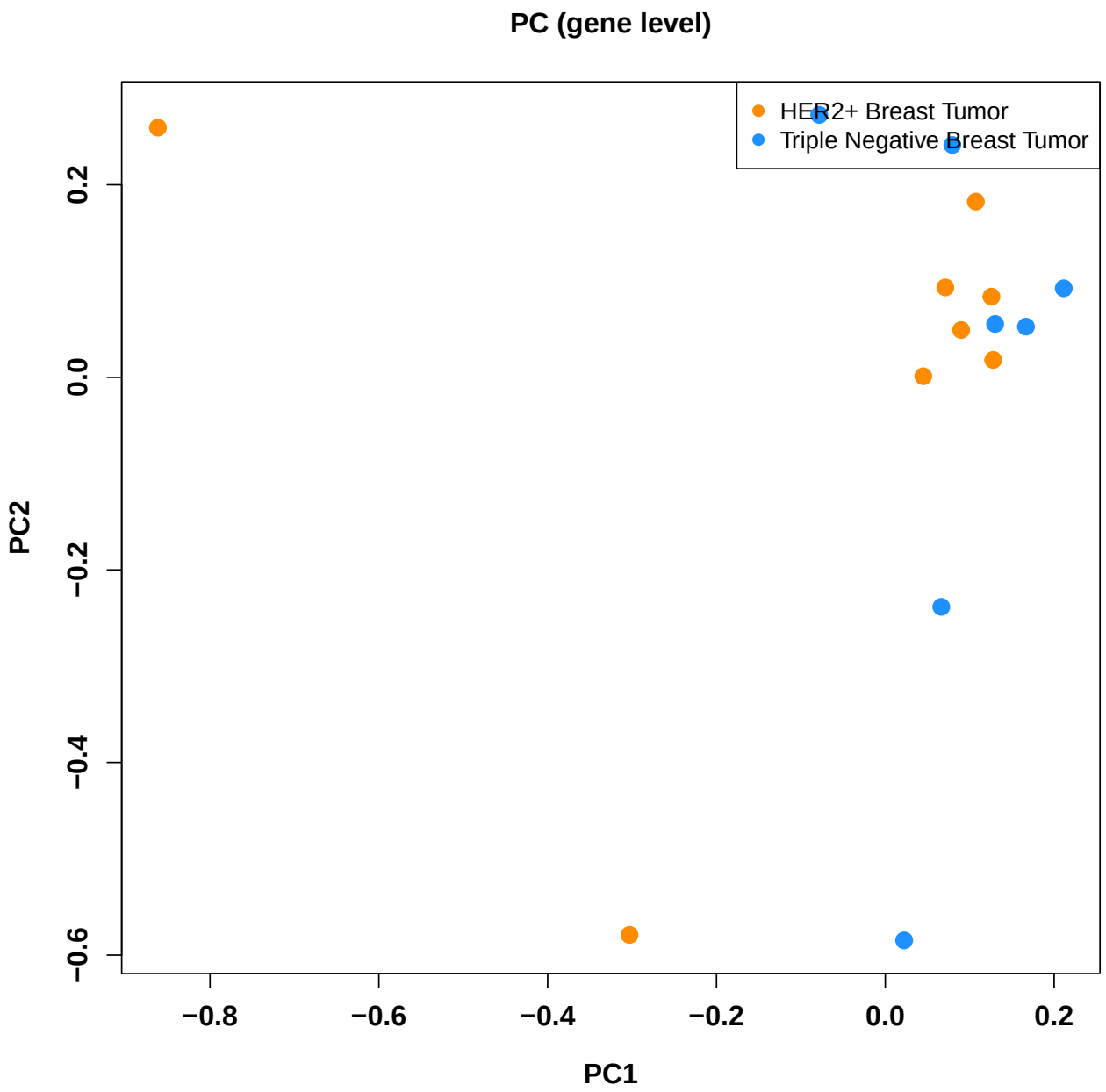



```
## Calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))
```

```
## 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 = 'Percent 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(rse1$group))))
```



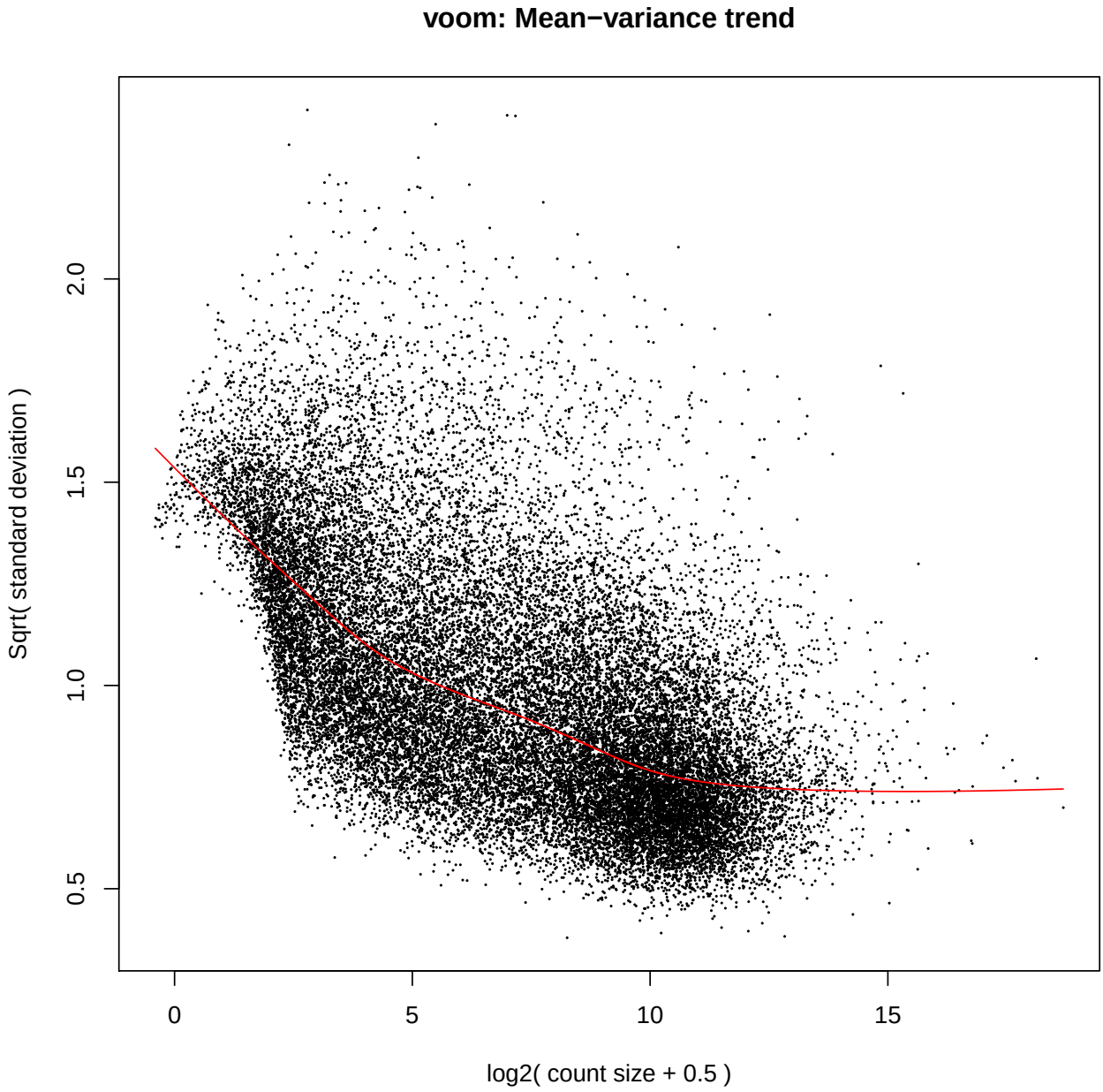
Differential gene expression between TNBC and HER2-positive samples is determined using `limma` and `voom`. Differentially expressed genes are visualized using a volcano plot to compare the effect size of the differential expression [as measured by the $log_2(foldchange)$ in expression] and its significance [$-log_10(p-value)$].

```
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse1$group)
design
```

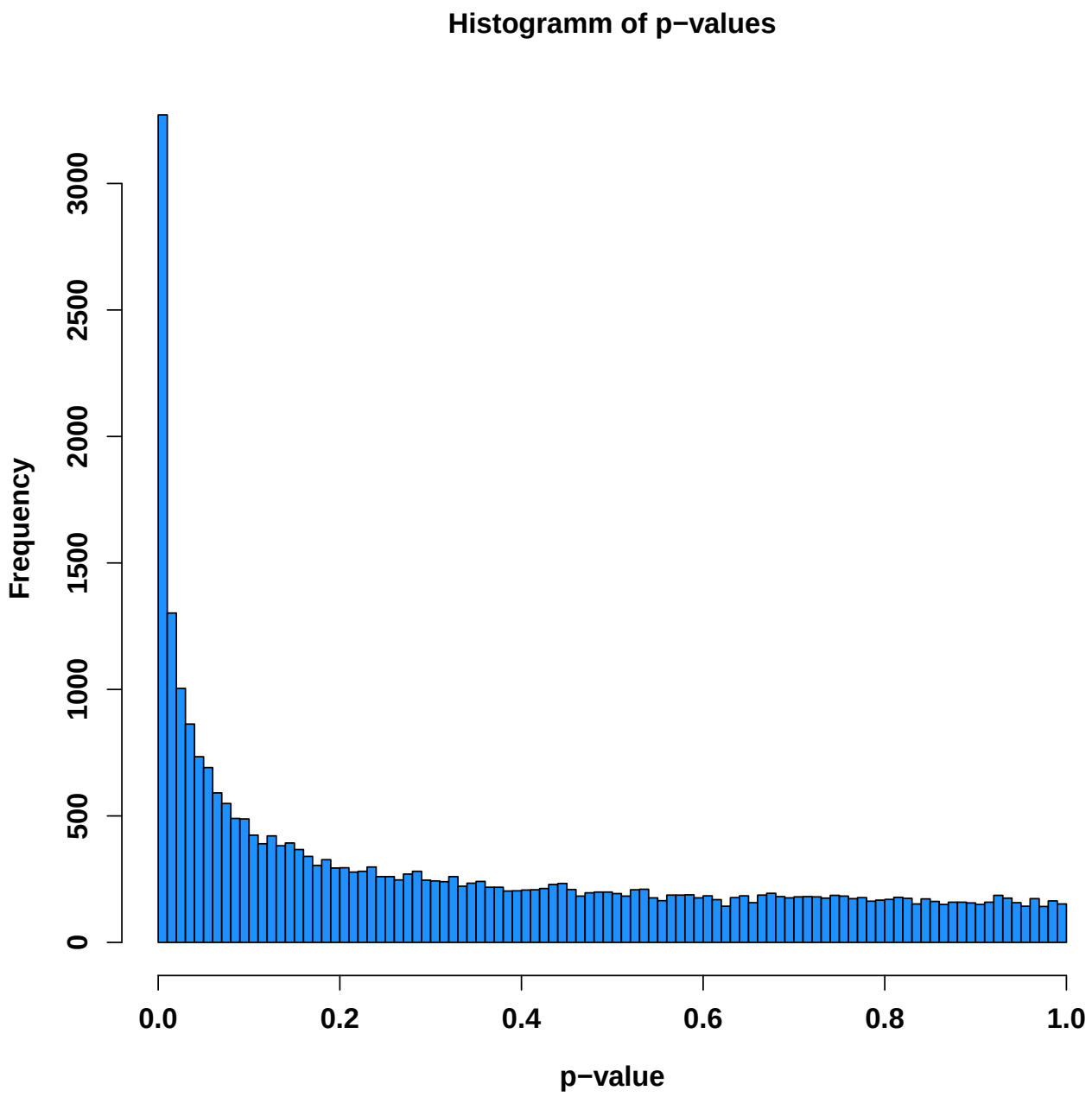
```
##    (Intercept) rse1$groupTriple Negative Breast Tumor
## 1            1                                     1
## 2            1                                     0
## 3            1                                     1
## 4            1                                     0
## 5            1                                     1
## 6            1                                     0
## 7            1                                     0
```

```
## 8           1                                       0
## 9           1                                       0
## 10          1                                       1
## 11          1                                       0
## 12          1                                       0
## 13          1                                       1
## 14          1                                       1
## 15          1                                       1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`rse1$group`
## [1] "contr.treatment"
```

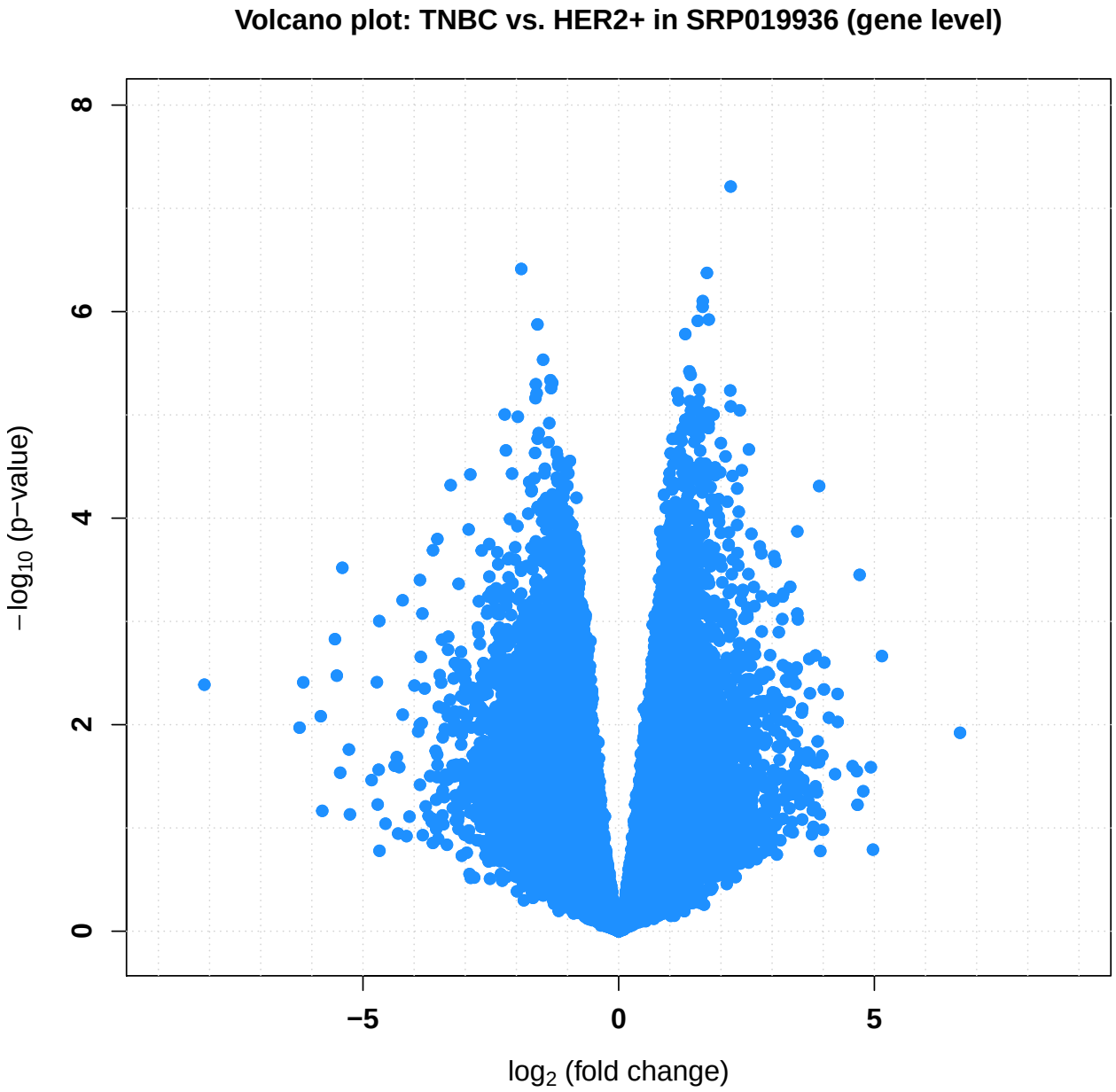
```
dge <- DGEList(counts = counts1)
dge <- calcNormFactors(dge)
v <- voom(dge, design,plot = TRUE)
```




```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC1 <- fit$coefficients[, 2]
t.mod1 <- fit$t[, 2]
p.mod1 <- fit$p.value[, 2]
q.mod1 <- qvalue(p.mod1)$q

## Histogram of p-values
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
hist(p.mod1, 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(log2FC1))
ry2 <- c(-0.1, max(-log10(p.mod1))) * 1.1
plot(log2FC1, -log10(p.mod1),
     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(log2FC1, -log10(p.mod1), pch = 19, col = trop[2])
title('Volcano plot: TNBC vs. HER2+ in SRP019936 (gene level)')
```



To compare these findings back to the breast cancer transcriptome data used to identify differential gene, exon, expressed region, and junction (SRP032798), we must again acquire these data, filter the read counts, and summarize gene expreesion as explained previously.

Independence hypotheses weighting

```
## Find second project of interest (SRP032789), e.g. with parts of the abstract
project_info2 <- abstract_search('To define the digital transcriptome of three breast cancer')

## Download the gene-level RangedSummarizedExperiment data
if(!file.exists(file.path('SRP032789', 'rse_gene.Rdata'))) {
    download_study(project_info2$project)
}
```

```
## Load the data
load(file.path(project_info2$project, 'rse_gene.Rdata'))
rse_gene2 <- rse_gene

## Scale counts by taking into account the total coverage per sample
rse2 <- scale_counts(rse_gene2)

## Download additional phenotype data from
## http://trace.ncbi.nlm.nih.gov/Traces/study/?acc=SRP032789
pheno2 <- read.table('SraRunTable_SRP032789.txt', sep = '\t',
                     header=TRUE,
                     stringsAsFactors = FALSE)

## Obtain correct order for pheno data
pheno2 <- pheno2[match(rse2$run, pheno2$Run_s), ]
identical(pheno2$Run_s, rse2$run)
```

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

```
head(cbind(pheno2$Run_s, rse2$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(rse2)$group <- pheno2$tumor_type_s
table(colData(rse2)$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
rse2 <- rse2[, rse2$group %in% c('HER2 Positive Breast Tumor', 'TNBC Breast Tumor')]
rse2
```

```
## class: RangedSummarizedExperiment
## dim: 58037 11
## metadata(0):
## assays(1): counts
## rownames(58037): ENSG00000000003.14 ENSG00000000005.5 ...
##   ENSG00000283698.1 ENSG00000283699.1
## rowData names(3): gene_id bp_length symbol
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(22): project sample ... characteristics group
```

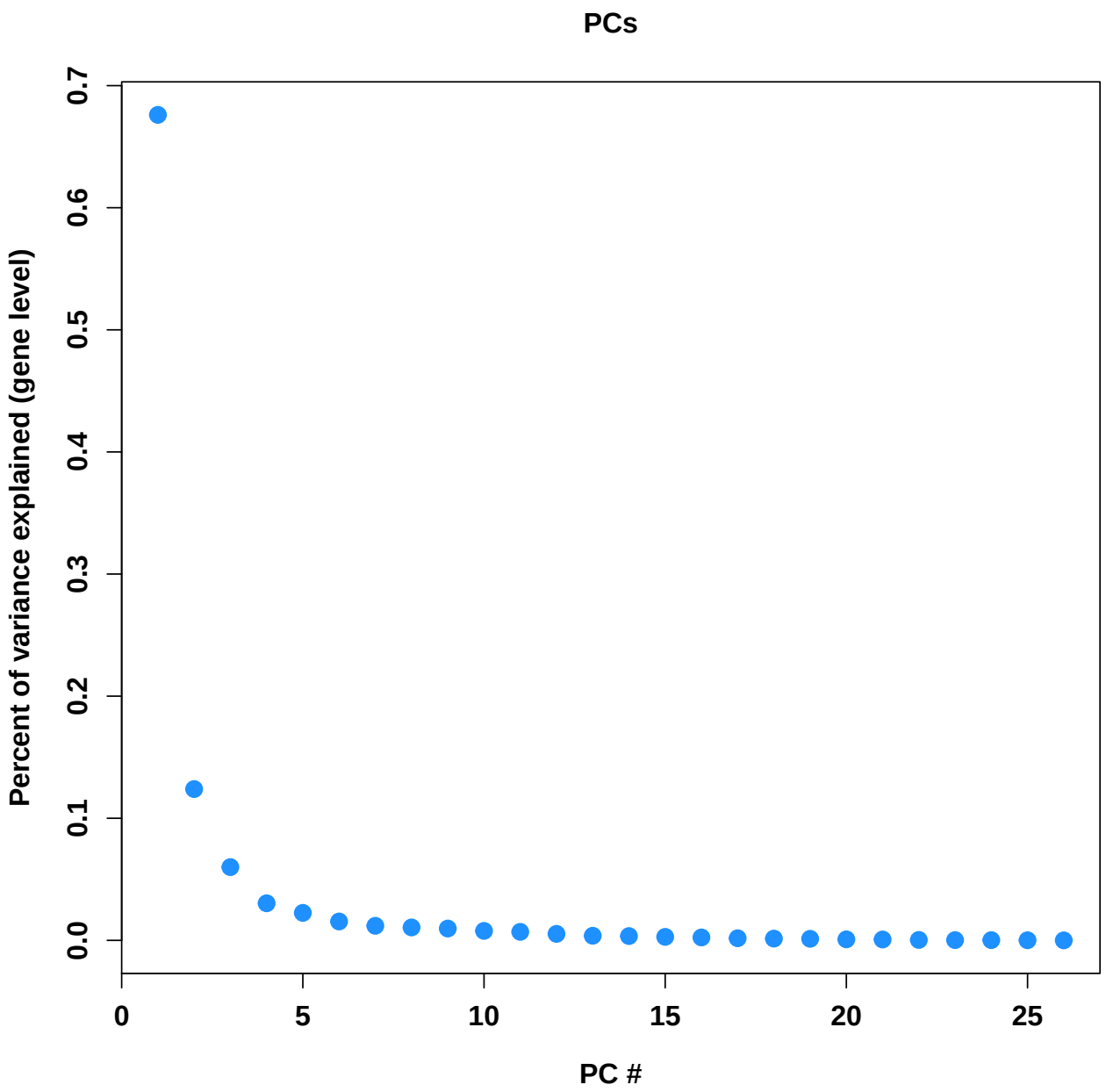
```
## Obtain count matrix without filtering
counts2 <- assays(rse2)$counts
dim(counts2)
```

```
## [1] 58037    11
```

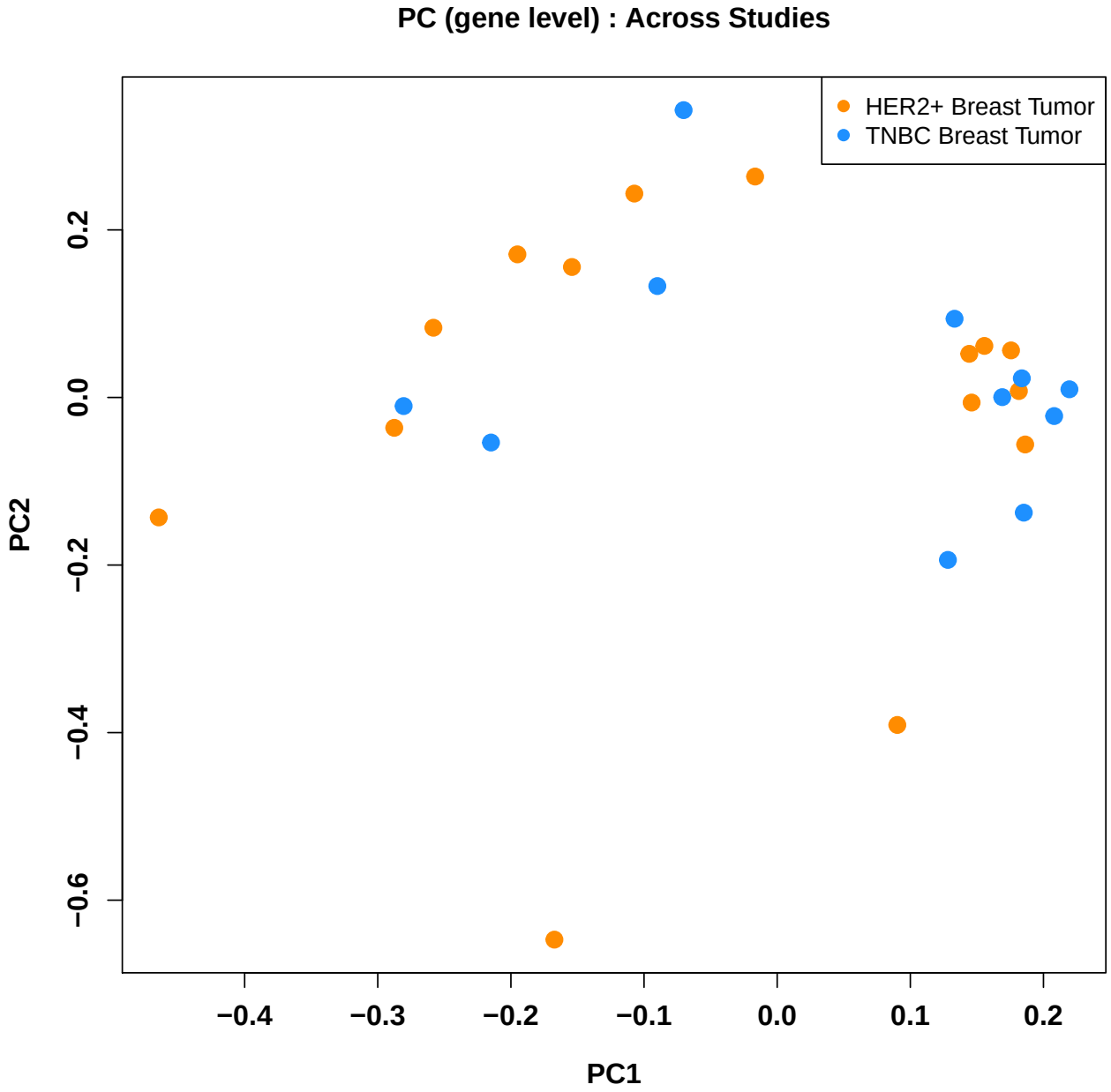
With count data from both studies, we will run PCA to assess global expression patterns across studies and samples.

Cross-study PCA

```
## Combine expression data across studies
combined_counts <- merge(counts1, counts2, by="row.names")
rownames(combined_counts) <- combined_counts$Row.names
combined_counts <- combined_counts[,-1]

#make sure phenotypes are annotated the same way
combined_pheno <- c(rse1$group,rse2$group)
combined_pheno <- gsub("Triple Negative Breast Tumor","TNBC Breast Tumor",combined_pheno)
combined_pheno <- gsub("HER2 Positive Breast Tumor","HER2+ Breast Tumor",combined_pheno)

## Calculate PCs with svd function
expr.pca <- svd(combined_counts - rowMeans(combined_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 = 'Percent 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) : Across Studies')
legend('topright', pch = 19, col = trop[c(1, 2)],
       names(summary(as.factor(combined_pheno)))
)
```



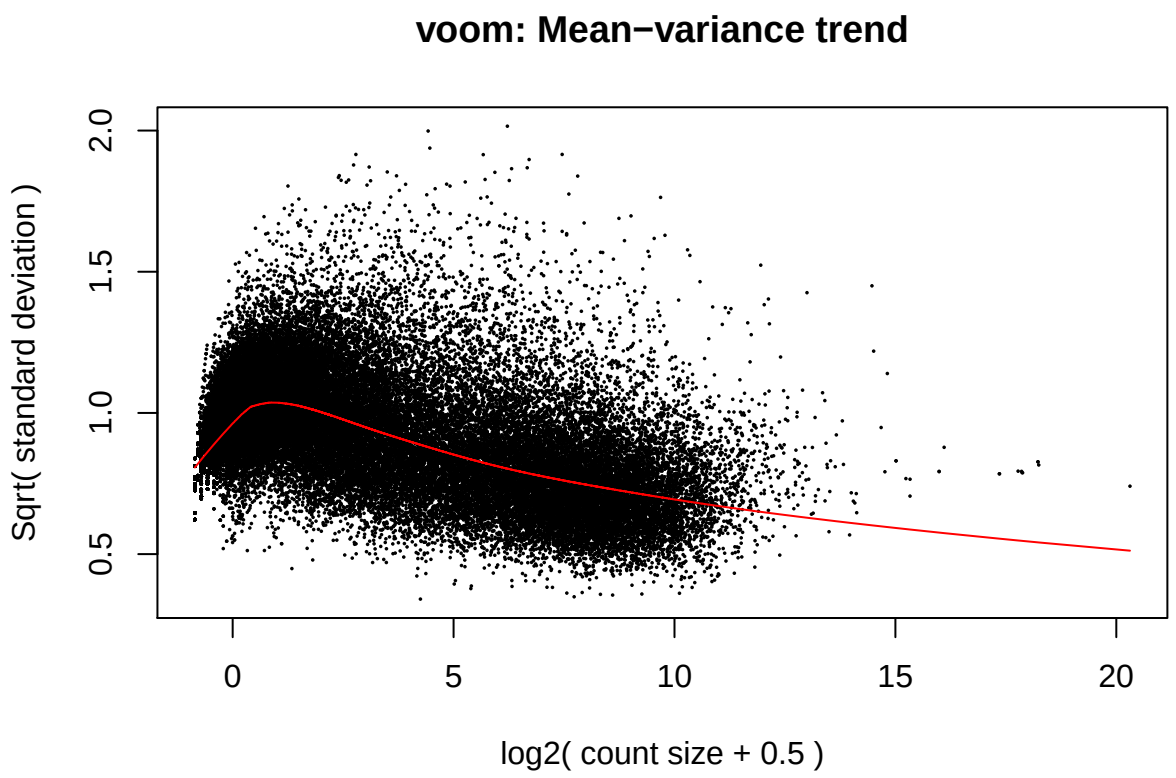
Differential gene expression is performed as it was done previously (`recount_SRP032789.Rmd`). Genes found in study 2 (SRP032798) that are also present in study 1 (SRP019936) are included for analysis. Independence hypotheses weighting (IHW) allows for the use of previous findings to be applied as priors to a current analysis as a means to improve power in the current study. Here, absolute values of the test statistic from study 2 were used as weights for the differential expression analysis in study 1 and p-value distributions of the differential expression analysis before and after applying IHW are compared.

```
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse2$group)
design
```

```
##    (Intercept) rse2$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")$`rse2$group`
## [1] "contr.treatment"
```

```
dge <- DGEList(counts = counts2)
dge <- calcNormFactors(dge)
v <- voom(dge, design,plot = TRUE)
```




```
fit <- lmFit(v, design)
fit <- eBayes(fit)
t.mod2 <- fit$t[, 2]
log2FC2 <- fit$coefficients[, 2]
p.mod2 <- fit$p.value[, 2]
q.mod2 <- qvalue(p.mod2)$q

## Use those genes from study 2 that are kept in study 1
t.mod2 <- t.mod2[names(t.mod1)]

## Use values of the test statistic from study 2 as weights for study 1
ihw.res <- ihw(p.mod1 ~ abs(t.mod2), alpha = 0.05)
head(ihw.res@df)
```

```
##                         pvalue adj_pvalue    weight weighted_pvalue group
## ENSG00000000003.14 0.12868605  0.3251598 1.0880948      0.11826731     9
## ENSG00000000005.5  0.29651780  0.5365931 1.0188440      0.29103355    12
## ENSG00000000419.12 0.04568144  0.2318954 0.7294672      0.06262301     2
## ENSG00000000457.13 0.02561366  0.1253512 1.3258948      0.01931802    18
## ENSG00000000460.16 0.20608374  0.3655948 1.4024446      0.14694609    18
## ENSG00000000938.12 0.18385023  0.5306072 0.6446970      0.28517308     7
##                    covariate fold
## ENSG00000000003.14 1.0979784    2
## ENSG00000000005.5  1.4830027    5
## ENSG00000000419.12 0.1913238    3
## ENSG00000000457.13 3.2957640    2
## ENSG00000000460.16 3.4305911    5
## ENSG00000000938.12 0.8287128    3
```

```
## Raw (orignial) p-values = p.mod1
p.ihw.raw <- ihw.res@df$pvalue
sum(p.ihw.raw < 0.05)
```

```
## [1] 7174
```

```
## Raw (orignial) q-values = q.mod1
q.ihw.raw <- qvalue(p.ihw.raw)$q
sum(q.ihw.raw < 0.05)
```

```
## [1] 3434
```

```
## Weighted p-values
p.ihw <- ihw.res@df$weighted_pvalue
sum(p.ihw < 0.05)
```

```
## [1] 7137
```

```
## q-values obtained from weighted p-values
q.ihw <- qvalue(p.ihw)$q
sum(q.ihw < 0.05)
```

```
## [1] 1648
```

```
## Recall: Differential expression summary statistics before IHW
sum(p.mod1 < 0.05)
```

```
## [1] 7174
```

```
sum(q.mod1 < 0.05)
```

```
## [1] 3434
```

```
#plot(p.ihw.raw, p.ihw)
```

To determine the concordance across studies, p-values are ranked and compared across genes present in both studies. Results are plotted such that the points falling along the identity line would indicate complete concodance between the two studies.

Concordance across studies

p-values from both studies

```
## filter count matrix for study 2
filter <- apply(counts2, 1, function(x) mean(x) > 5)
counts2 <- counts2[filter, ]
dim(counts2)
```

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

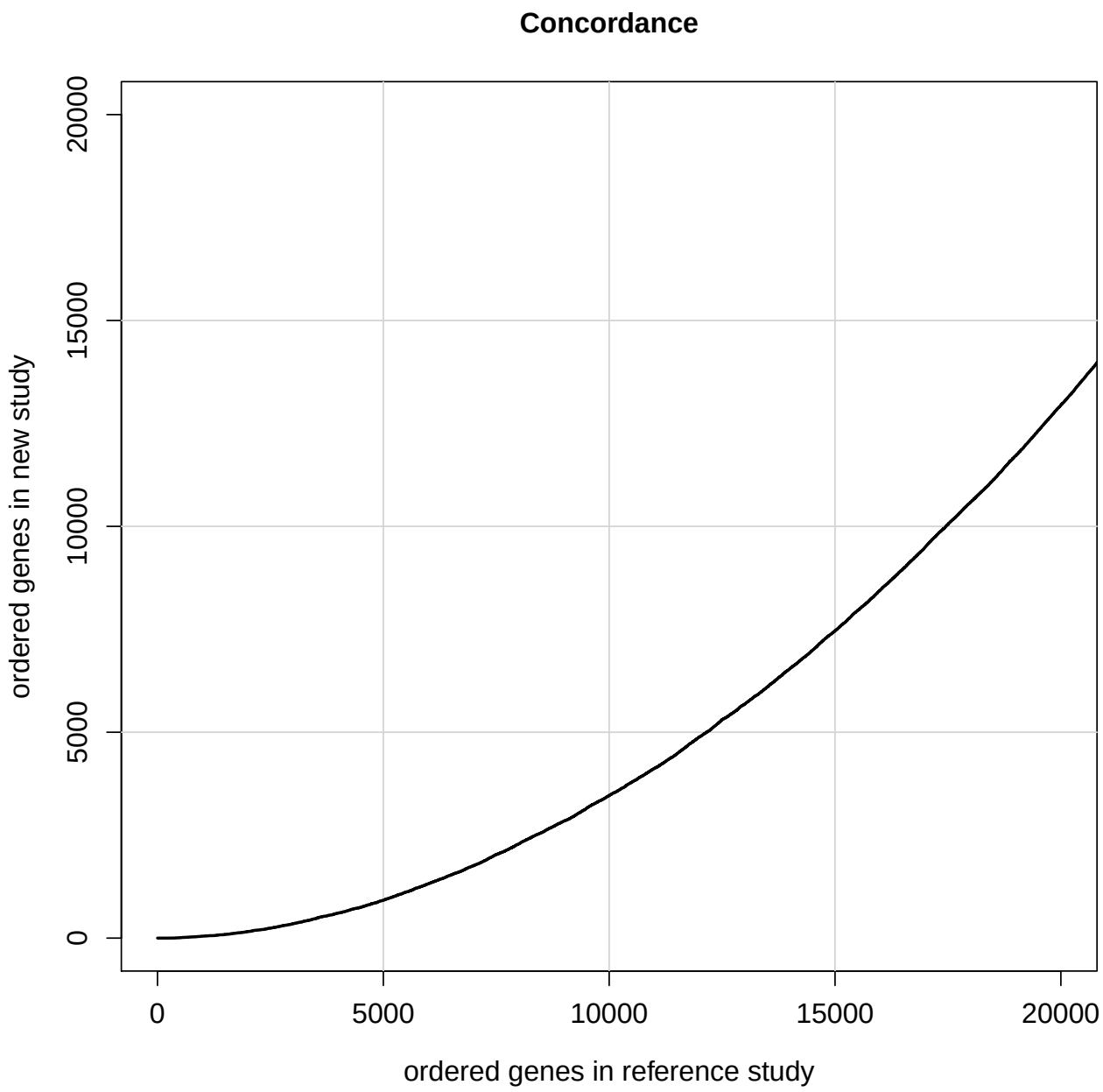
```
## filter p-values for study 2 (was not filtered before)
p.mod2 <- p.mod2[rownames(counts2)]

## sort p-values
p.mod1.sort <- p.mod1[order(p.mod1)]
p.mod2.sort <- p.mod2[order(p.mod2)]

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

```
##
## FALSE  TRUE
##  4559 24764
```

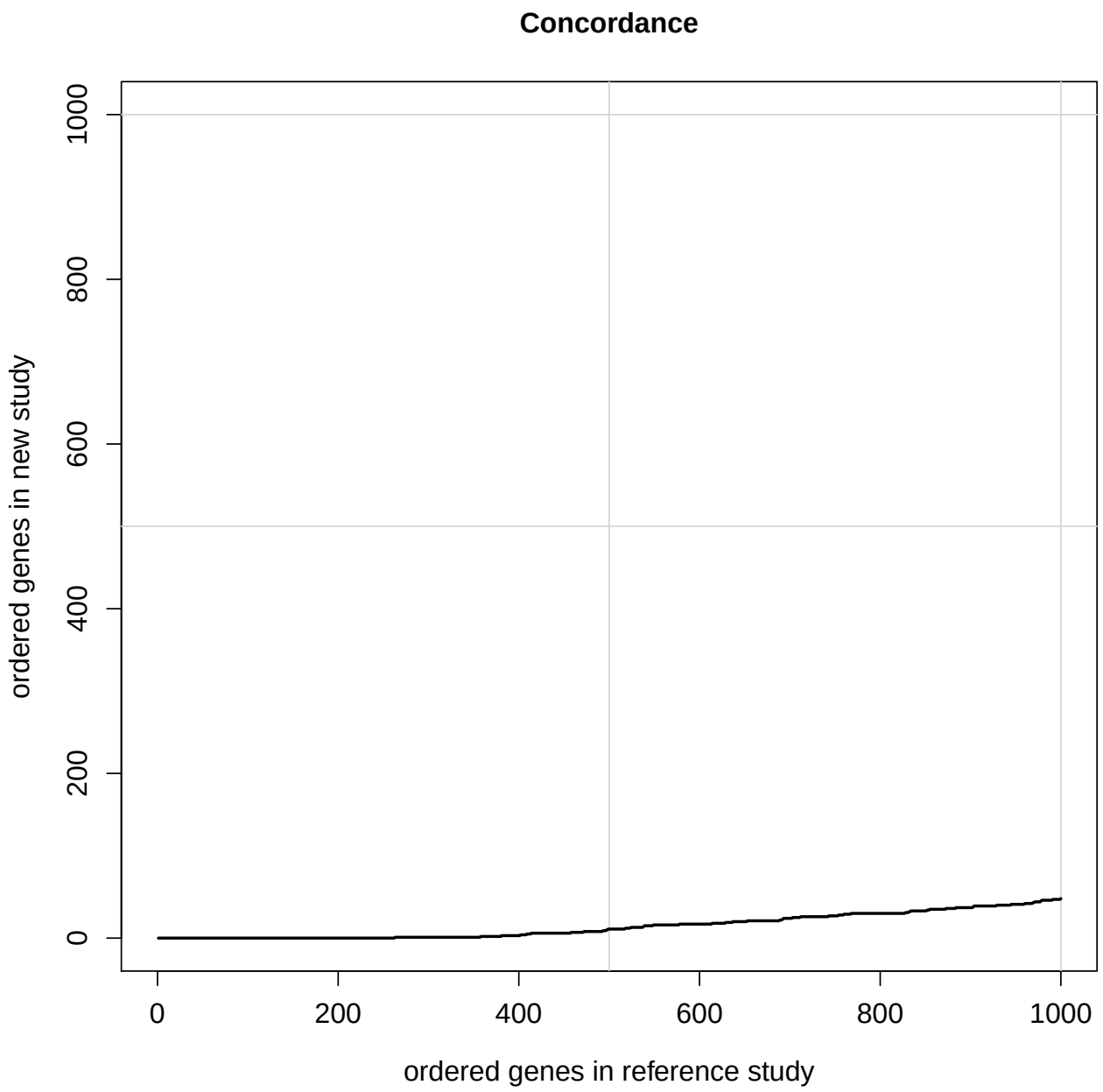
```
table(names(p.mod2.sort) %in% names(p.mod1.sort))
```

```
##
## FALSE  TRUE
##  1978 24764
```

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

## all genes
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5, cex.axis = 1.2)
plot(seq(1:length(p.mod2.sort)), conc,
     type = 'l', las = 0,
     xlim = c(0, 20000),
     ylim = c(0, 20000),
     xlab = 'ordered genes in reference study',
     ylab = 'ordered genes in new study',
     main = 'Concordance')
for(k in 1:3){
    abline(v = k * 5000, cex = 0.5, col = 'lightgrey')
    abline(h = k * 5000, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod2.sort)), conc, col = 'black', lwd = 2)
```




```
## top 1000 genes
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5, cex.axis = 1.2)
plot(seq(1:1000), conc[1:1000],
     type = 'l', las = 0,
     xlim = c(0, 1000),
     ylim = c(0, 1000),
     xlab = 'ordered genes in reference study',
     ylab = 'ordered genes in new study',
     main = 'Concordance')
for(k in 1:2){
    abline(v = k * 500, cex = 0.5, col = 'lightgrey')
    abline(h = k * 500, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:1000), conc[1:1000], col = 'black', lwd = 2)
```



p-values IHW vs. raw p-values from both study

```
## sort p-values (ihw procedure)
## p.mod1 and p.mod2 are sorted
names(p.ihw) <- rownames(ihw.res@df)
p.ihw.sort <- p.ihw[order(p.ihw)]

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

```
##
## FALSE  TRUE
##  4559 24764
```

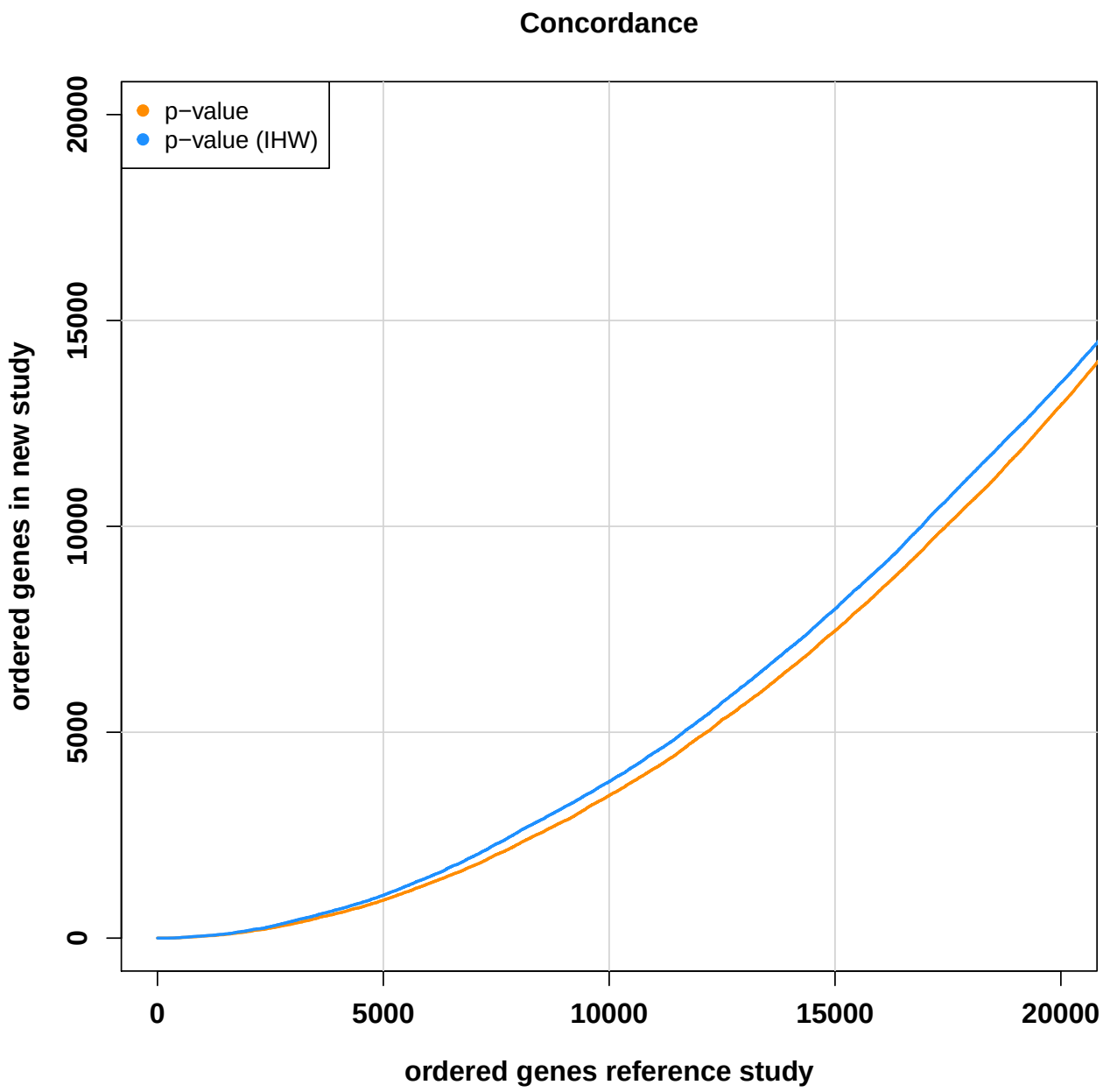
```r
table(names(p.mod2.sort) %in% names(p.mod1.sort))
```

```
##
## FALSE  TRUE
##  1978 24764
```

```r
table(names(p.mod1.sort) %in% names(p.ihw.sort))
```

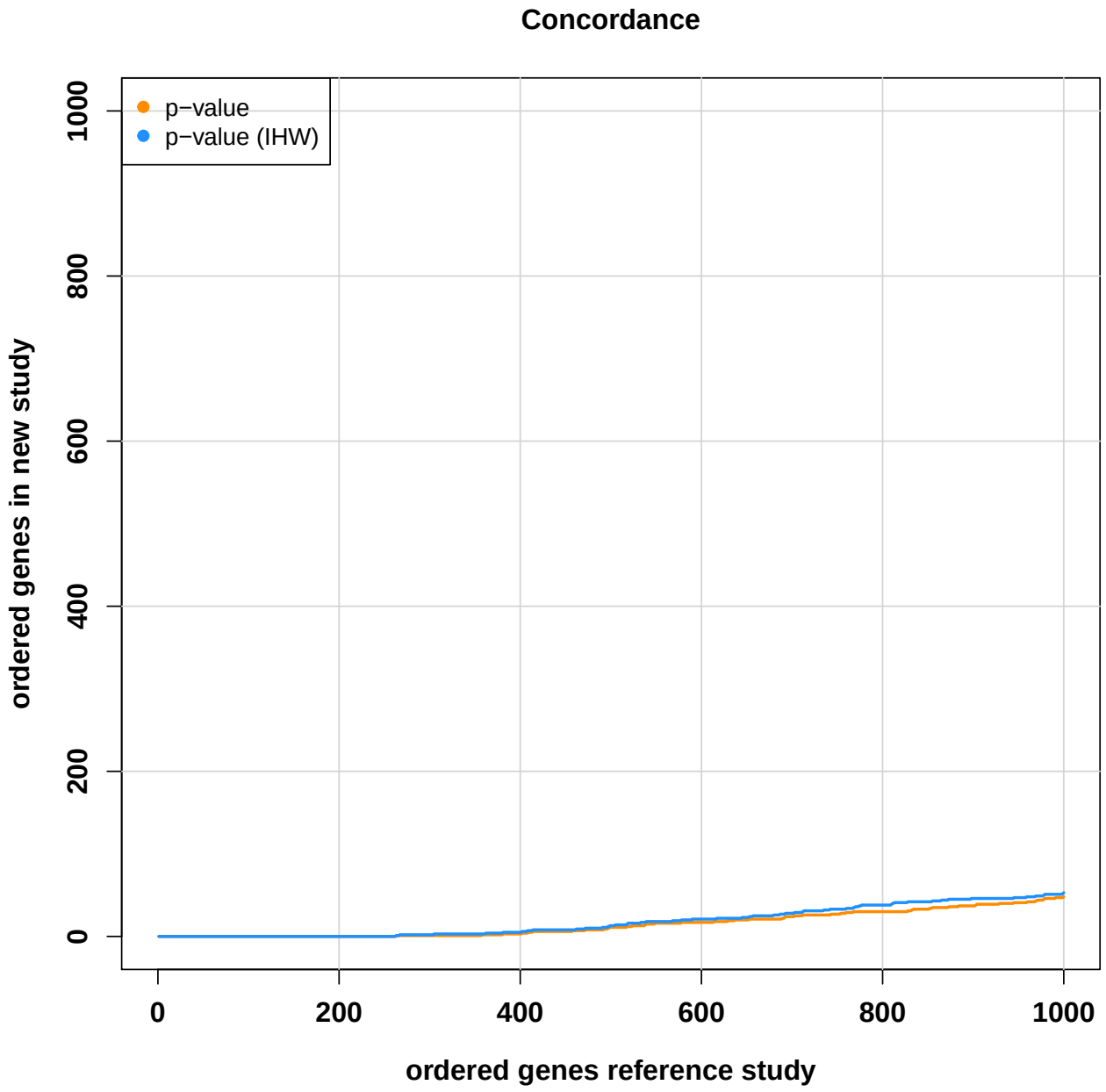
```
##
##  TRUE
## 29323
```

```r
conc_p.mod <- NULL
conc_p.ihw <- NULL
for(i in 1:length(p.mod2.sort)){
    conc_p.mod[i] <- sum(names(p.mod2.sort)[1:i] %in% names(p.mod1.sort)[1:i])
    conc_p.ihw[i] <- sum(names(p.mod2.sort)[1:i] %in% names(p.ihw.sort)[1:i])
}

## Concordance plot
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod2.sort)), conc_p.mod,
     type = 'l', las = 0,
     xlim = c(0, 20000),
     ylim = c(0, 20000),
     xlab = 'ordered genes reference study',
     ylab = 'ordered genes in new study',
     main = 'Concordance')
for(k in 1:3){
    abline(v = k * 5000, cex = 0.5, col = 'lightgrey')
    abline(h = k * 5000, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod2.sort)), conc_p.mod, col = trop[1], type = 'l', lwd = 2)
points(seq(1:length(p.mod2.sort)), conc_p.ihw, col = trop[2], type = 'l', lwd = 2)
legend('topleft', pch = 19, col = trop[c(1, 2)], c("p-value", "p-value (IHW)"))
```




```
## top 1000 genes
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod2.sort[1:1000])), conc_p.mod[1:1000],
     type = 'l', las = 0,
     xlim = c(0, 1000),
     ylim = c(0, 1000),
     xlab = 'ordered genes reference study',
     ylab = 'ordered genes in new study',
     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')
}
lines(seq(1:length(p.mod2.sort[1:1000])), conc_p.mod[1:1000], col = trop[1], type = 'l', lwd = 2)
points(seq(1:length(p.mod2.sort[1:1000])), conc_p.ihw[1:1000], col = trop[2], type = 'l', lwd = 2)
```

```
legend('topleft', pch = 19, col = trop[c(1, 2)], c("p-value", "p-value (IHW)"))
```



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)
- *IHW*
- *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, 2016)
- *limma* (Law, Chen, Shi, and Smyth, 2014)

Bibliography file

[1] 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.

[2] 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.

[3] 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.

[4] C. Boettiger. knitcitations: Citations for 'Knitr' Markdown Files. R package version 1.0.7. 2015. URL: https://CRAN.R-project.org/package=knitcitations.

[5] 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.

[6] 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.

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

[8] 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.

[9] M. Morgan, V. Obenchain, J. Hester and H. Pagès. SummarizedExperiment: SummarizedExperiment container. R package version 1.5.3. 2016.

[10] 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.

[11] 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/.

[12] 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.

[13] 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 4.682282 mins
```

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

```
## 2017-01-30 15:07:29
```

```
## 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     2017-01-30
## Packages -----------------------------------------------------------------------------------------
##  package              * version  date       source
##  acepack                1.4.1    2016-10-29 CRAN (R 3.4.0)
##  AnnotationDbi          1.37.1   2017-01-13 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             2.43.3   2017-01-24 Bioconductor
##  bitops                 1.0-6    2013-08-17 CRAN (R 3.4.0)
##  BSgenome               1.43.4   2017-01-20 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             1.10.0   2016-12-03 CRAN (R 3.4.0)
##  DBI                    0.5-1    2016-09-10 CRAN (R 3.4.0)
##  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)
##  fdrtool                1.2.15   2015-07-08 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         * 1.11.6   2016-11-17 Bioconductor
##  GenomicAlignments      1.11.8   2017-01-24 Bioconductor
##  GenomicFeatures        1.27.6   2016-12-17 Bioconductor
##  GenomicFiles           1.11.3   2016-11-29 Bioconductor
##  GenomicRanges        * 1.27.21  2017-01-20 Bioconductor
##  GEOquery               2.41.0   2016-10-25 Bioconductor
##  ggplot2                2.2.1    2016-12-30 CRAN (R 3.4.0)
```

```
##  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)
##  IHW                  * 1.3.1    2017-01-20 Bioconductor
##  IRanges              * 2.9.16   2017-01-28 cran (@2.9.16)
##  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                * 3.31.10  2017-01-26 Bioconductor
##  locfit                 1.5-9.1  2013-04-20 CRAN (R 3.4.0)
##  lpsymphony             1.3.0    2016-10-23 Bioconductor (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)
##  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              * 1.1.14   2017-01-30 Github (leekgroup/recount@009bb32)
##  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            1.35.3   2017-01-30 Github (Bioconductor-mirror/rtracklayer@5f195a1)
##  S4Vectors            * 0.13.11  2017-01-28 cran (@0.13.11)
##  scales                 0.4.1    2016-11-09 CRAN (R 3.4.0)
##  slam                   0.1-40   2016-12-01 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 * 1.5.3    2016-11-11 Bioconductor
##  survival               2.40-1   2016-10-30 CRAN (R 3.4.0)
```

```
##  tibble                1.2      2016-08-26 CRAN (R 3.4.0)
##  VariantAnnotation     1.21.15  2017-01-20 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               0.15.1   2017-01-24 Bioconductor
##  yaml                  2.1.14   2016-11-12 CRAN (R 3.4.0)
##  zlibbioc              1.21.0   2016-10-23 Bioconductor
```