

recount (overlay two studies)

Kai Kammers and Shannon Ellis

July 22, 2016

Contents

Load R-packages	1
Gene level analysis	1
Independence hypotheses weighting	17
Concordance across studies	21
p-values from both studies	21
p-values IHW vs. raw p-values from both study	24
Reproducibility	27

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 18 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_aligned
##                           <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_aligned 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_tissue sharq_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  bigwig_file

```

```

## <integer> <character>
## SRR791043 100 SRR791043.bw
## SRR791044 100 SRR791044.bw
## SRR791045 100 SRR791045.bw
## SRR791046 100 SRR791046.bw
## SRR791047 100 SRR791047.bw
## ... ...
## SRR791070 100 SRR791070.bw
## SRR791071 100 SRR791071.bw
## SRR791072 100 SRR791072.bw
## SRR791073 100 SRR791073.bw
## SRR791074 100 SRR791074.bw

## Gene info
rowData(rse_gene1)

```

```

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

```

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

```

## Scale counts by taking into account the total coverage per sample
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: 23779 16
## metadata(0):
## assays(1): counts
## rownames(23779): 1 10 ... 9994 9997
## rowData names(2): gene_id bp_length
## colnames(16): SRR791051 SRR791052 ... SRR791065 SRR791074
## colData names(19): project sample ... bigwig_file 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] 18333    16

```

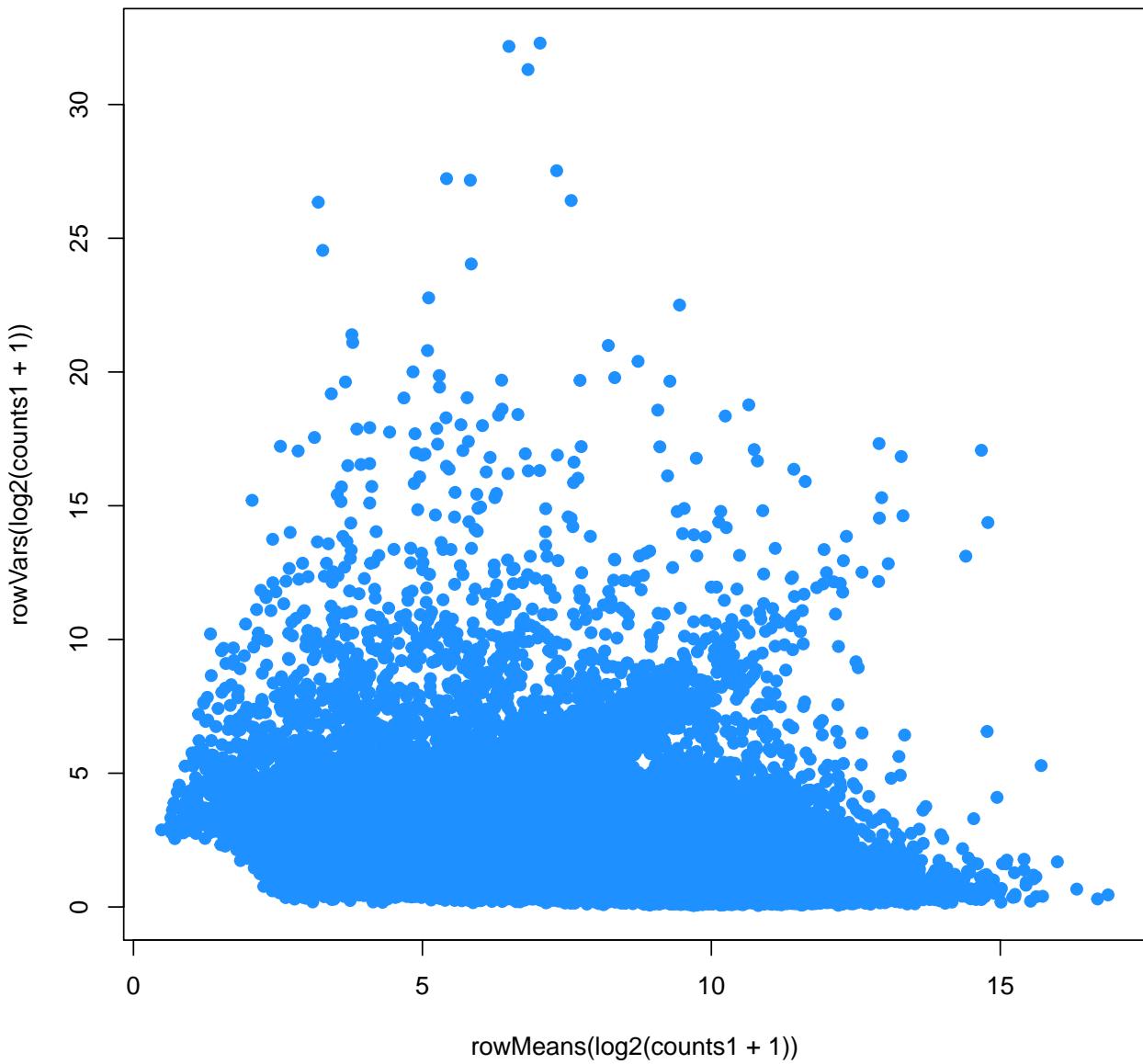
To get a better sense of the data, we plot the mean-variance relationship for each gene. Similarly, we run principal component analysis (PCA) to identify any sample outliers within the data. We assess the variance explained by each of the first 11 PCs as well as visualize the relationship of each sample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

```

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

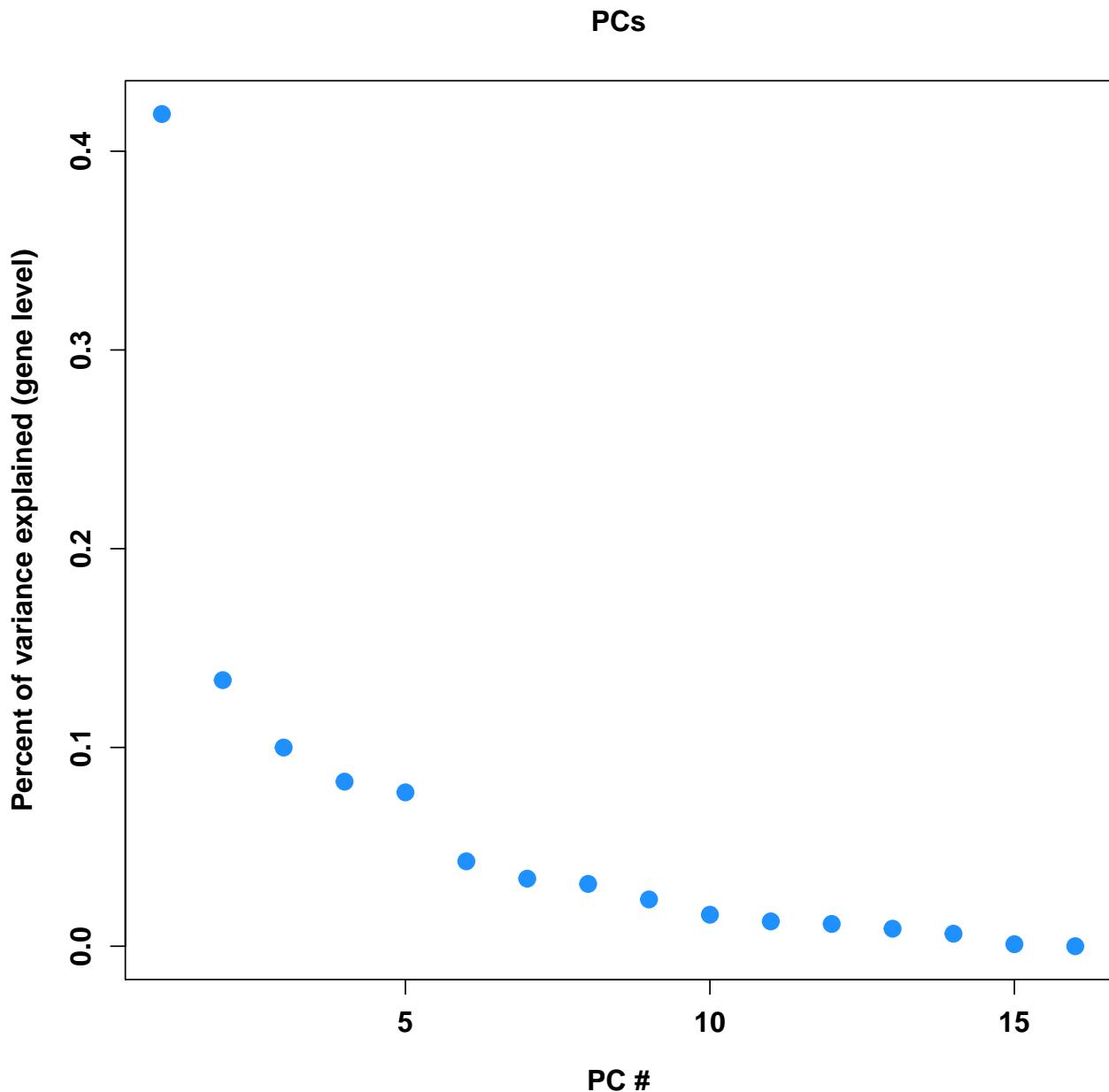
```

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

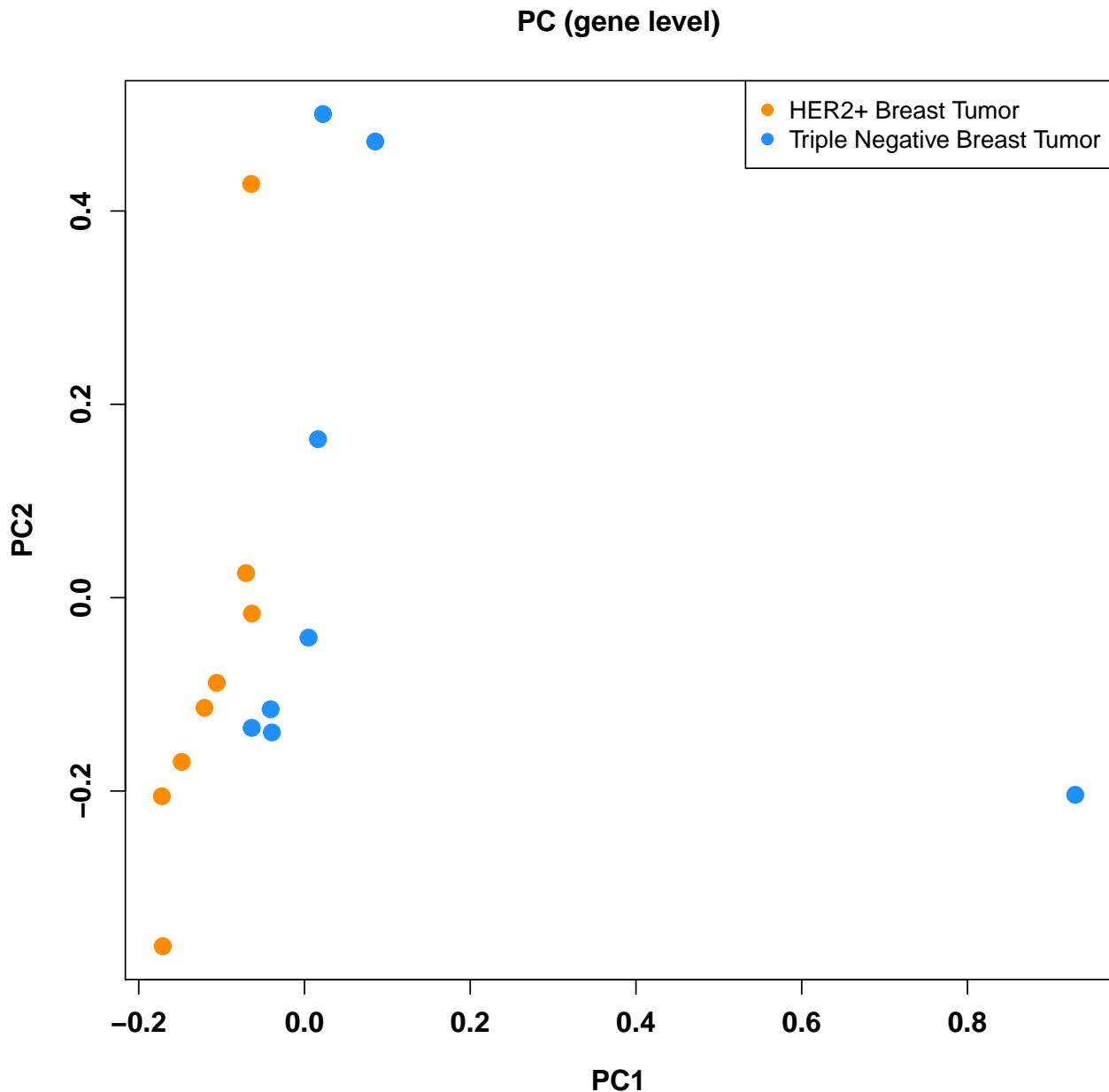


```
## Calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))

## 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
##      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: 23779 16
## metadata(0):
## assays(1): counts
## rownames(23779): 1 10 ... 9994 9997
## rowData names(2): gene_id bp_length
## colnames(16): SRR791051 SRR791052 ... SRR791065 SRR791074
## colData names(19): project sample ... bigwig_file group

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

## class: RangedSummarizedExperiment
## dim: 23779 15
## metadata(0):
## assays(1): counts
## rownames(23779): 1 10 ... 9994 9997
## rowData names(2): gene_id bp_length
## colnames(15): SRR791051 SRR791052 ... SRR791064 SRR791074
## colData names(19): project sample ... bigwig_file 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] 18370    15

```

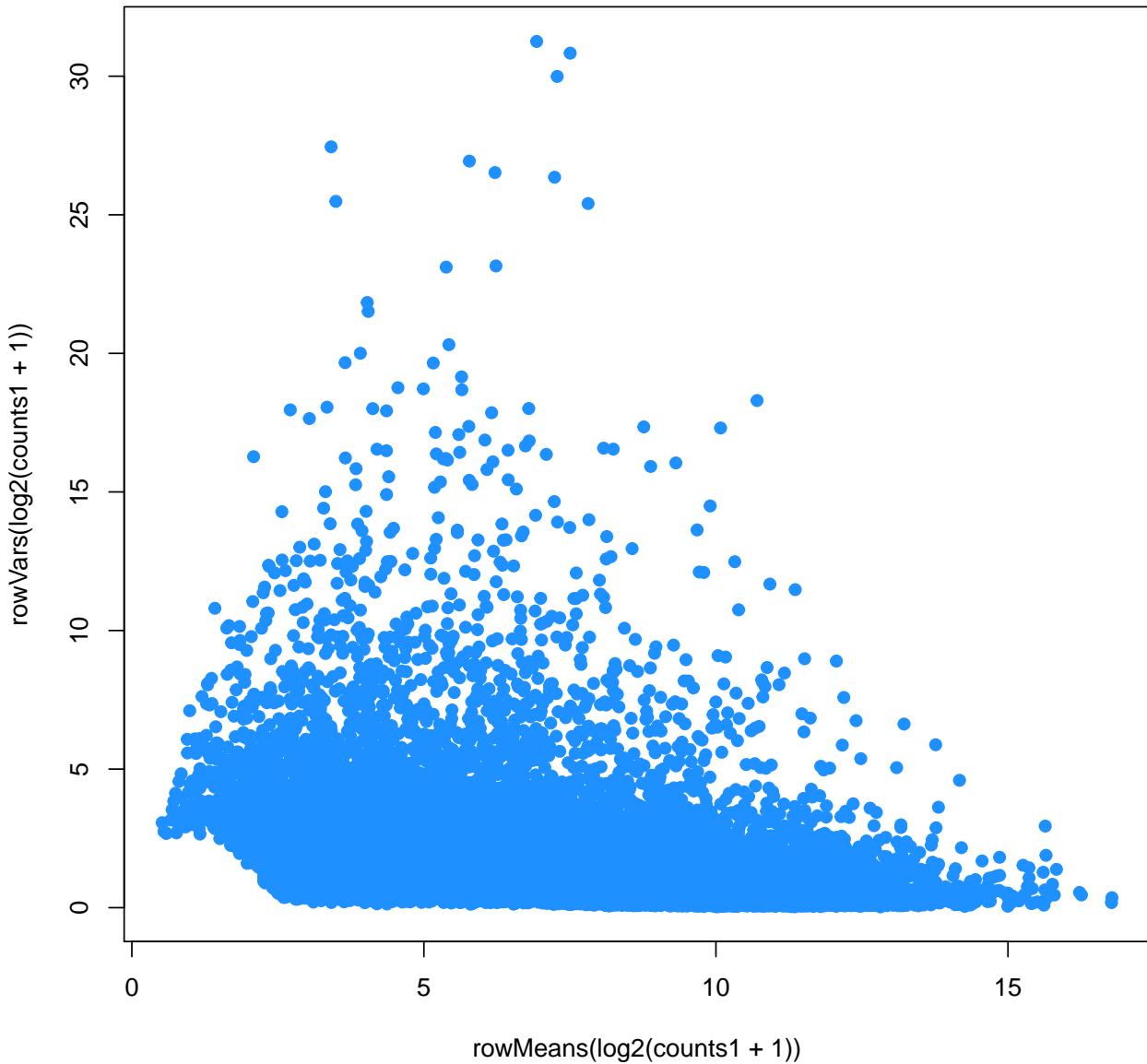
After sample outlier removal, PCA is again run to obtain a global understanding of the mean-variance relationship at each gene and the global relationship between samples included for study.

```

## 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])

```

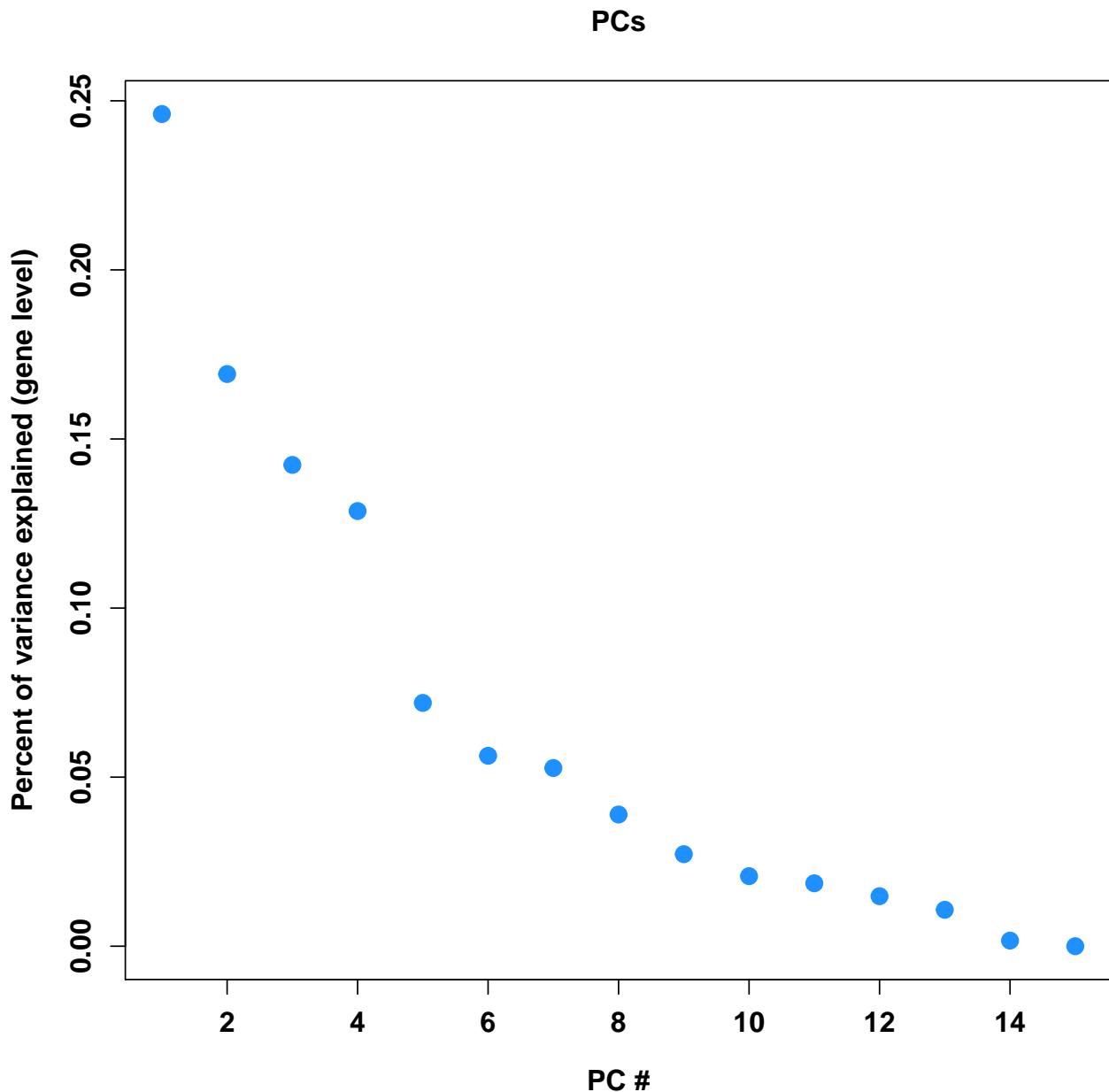


```

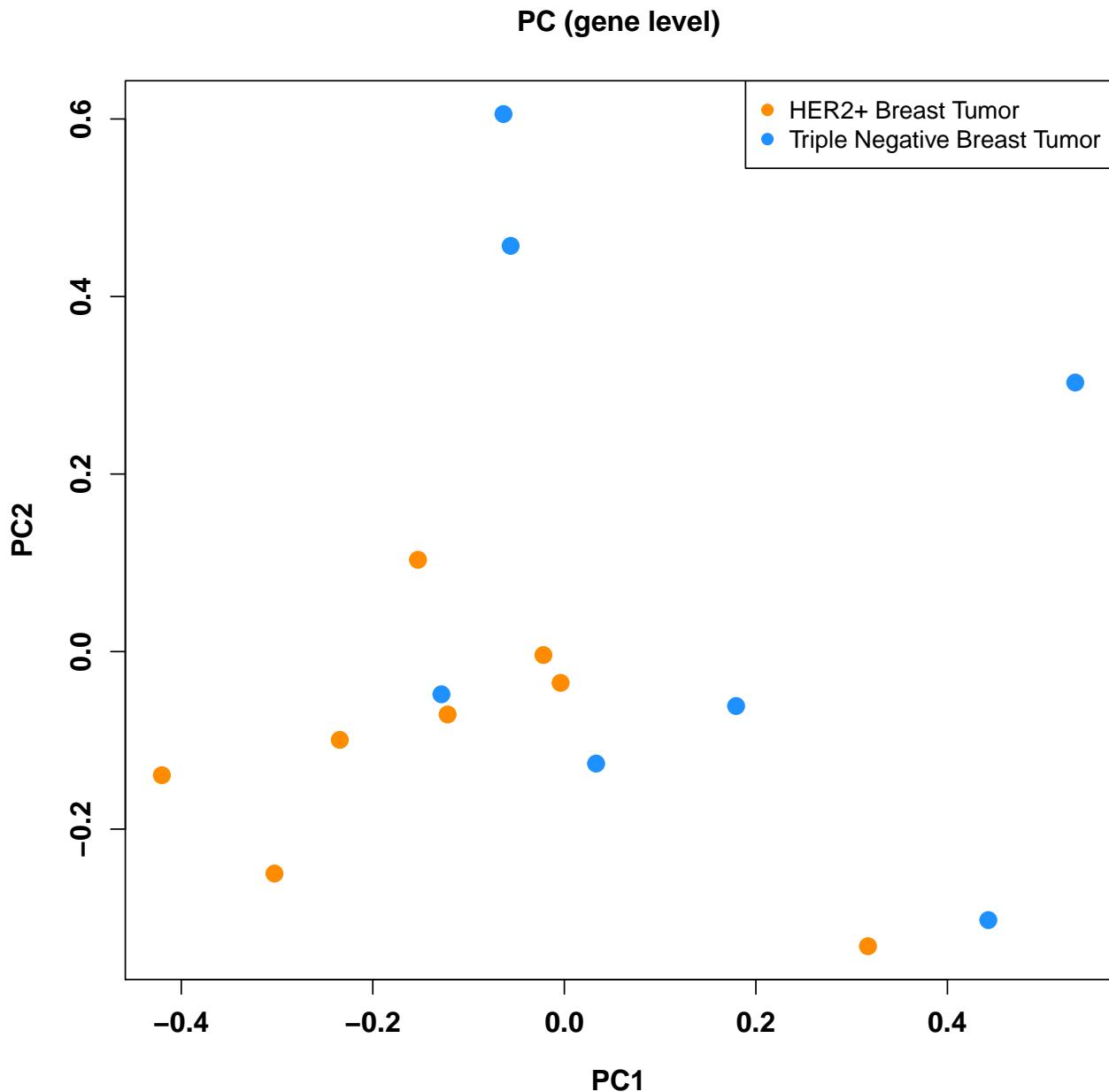
## Calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))

## 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 [$-\log_{10}(p - value)$].

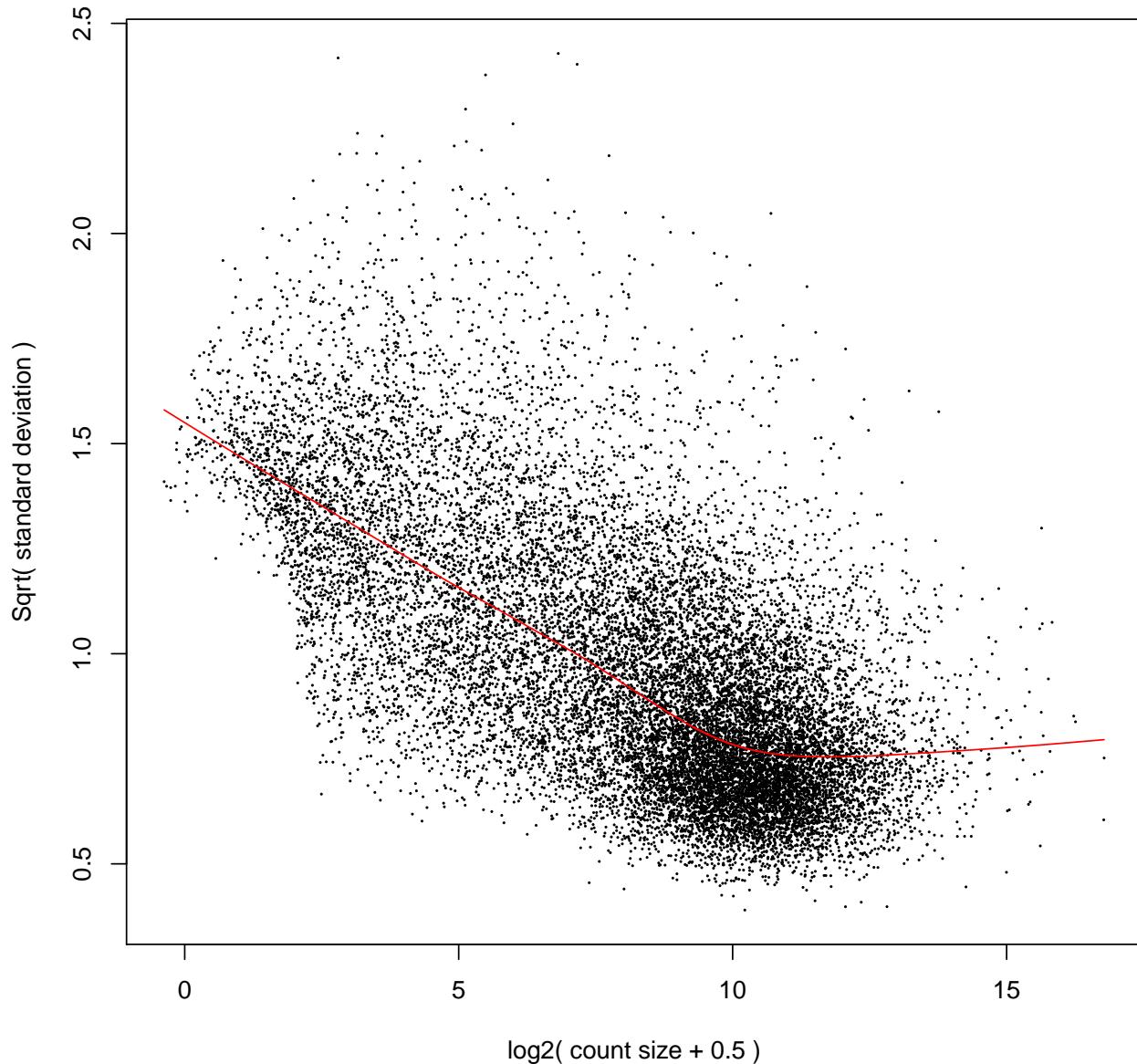
```
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse1$group)
design
```

```
##      (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)
```

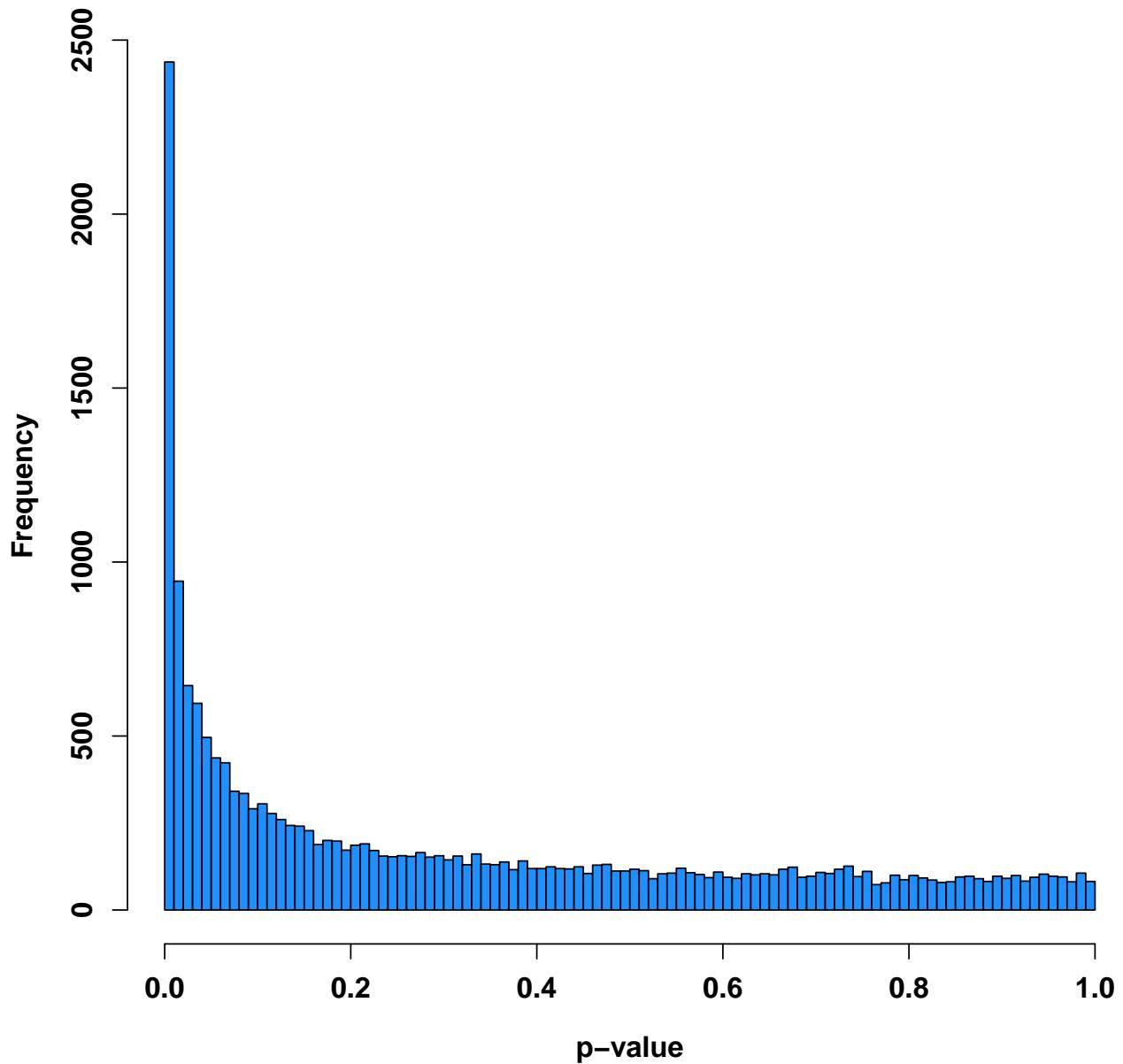
voom: Mean–variance trend



```
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

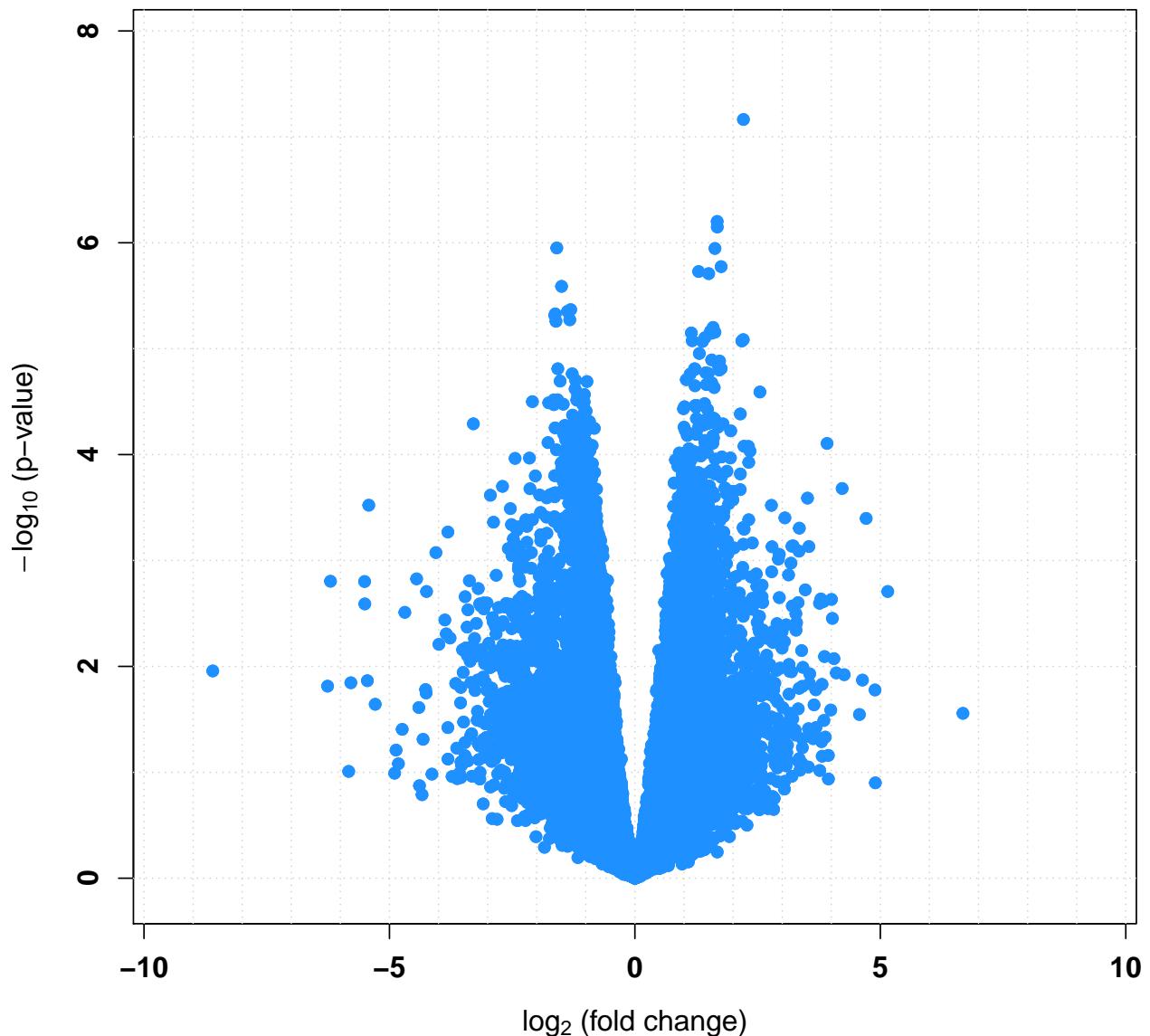
## Histogram of p-values
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
hist(p.mod1, col = trop[2], xlab = 'p-value',
     main = 'Histogramm of p-values', breaks = 100)
```

Histogramm of p-values



```
## Volcano plot
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
rx2 <- c(-1, 1) * 1.1 * max(abs(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)')
```

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 expression 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: 23779 11
## metadata(0):
## assays(1): counts
## rownames(23779): 1 10 ... 9994 9997
## rowData names(2): gene_id bp_length
## colnames(11): SRR1027171 SRR1027173 ... SRR1027187 SRR1027172
## colData names(19): project sample ... bigwig_file group
```

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

```
## [1] 23779    11
```

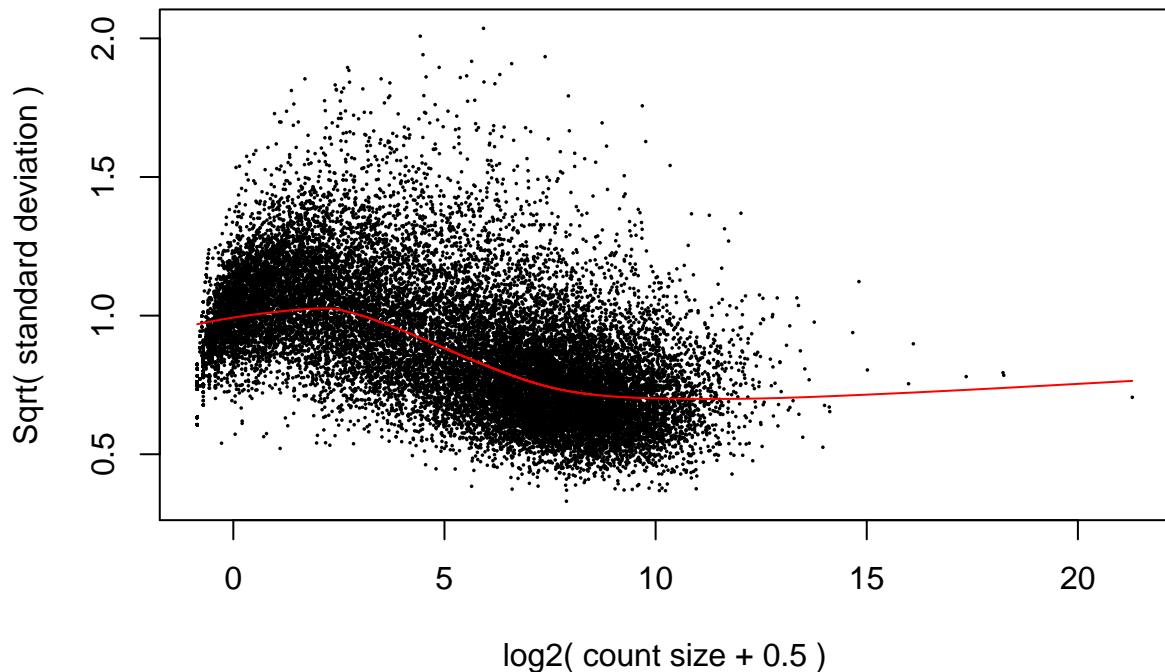
Differential gene expression is performed as it was done previously (`recount_SRP032789.Rmd`). Genes found in study 2 (SRP032798) that are also present in study 1 (SRP019936) are included for analysis. Independence hypotheses weighting (IHW) allows for the use of previous findings to be applied as priors 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)
```

voom: Mean–variance trend



```

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 covariate
## 1      0.53356404  0.7112896 1.000000   0.533564036     9 1.8505302
## 10     0.46942017  0.6590179 1.000000   0.469420172     7 1.3749147
## 100    0.58806782  0.6503988 1.281798   0.458783634     4 0.7023154
## 1000   0.87146306  0.9383792 1.000000   0.871463063     5 0.8411674
## 100008589 0.01187256  0.0724467 1.281798   0.009262426     4 0.7600378
## 100009676 0.22427859  0.5565072 0.639021   0.350972165     7 1.3856221
##          fold
## 1        2
## 10       1
## 100      3
## 1000     2
## 100008589 3
## 100009676 3

```

```

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

## [1] 5117

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

## [1] 3197

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

## [1] 5088

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

## [1] 2582

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

## [1] 5117

sum(q.mod1 < 0.05)

## [1] 3197

#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 concordance 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] 17874    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
## 1132 17238

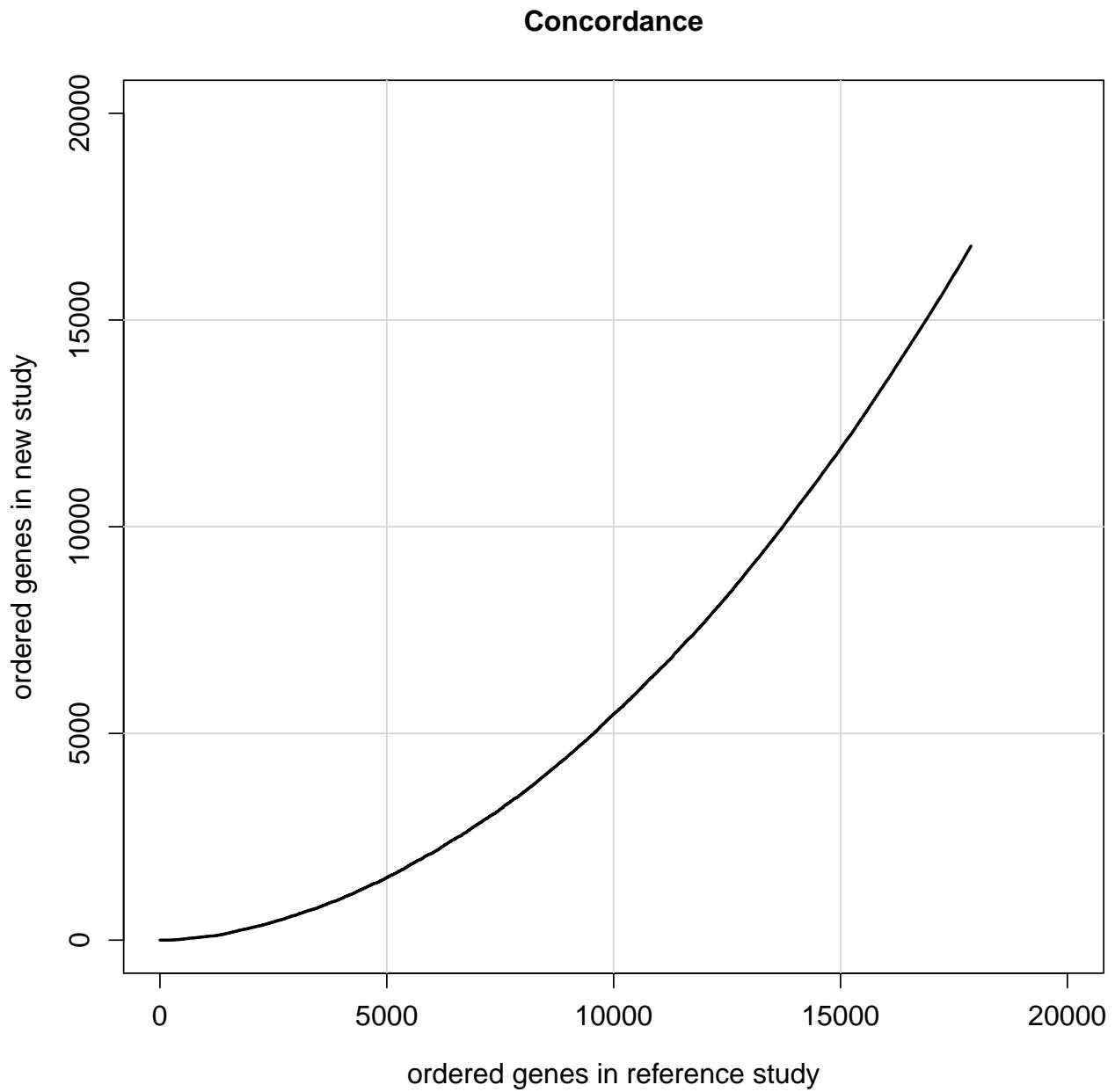
table(names(p.mod2.sort) %in% names(p.mod1.sort))

## 
## FALSE  TRUE
## 636 17238

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)

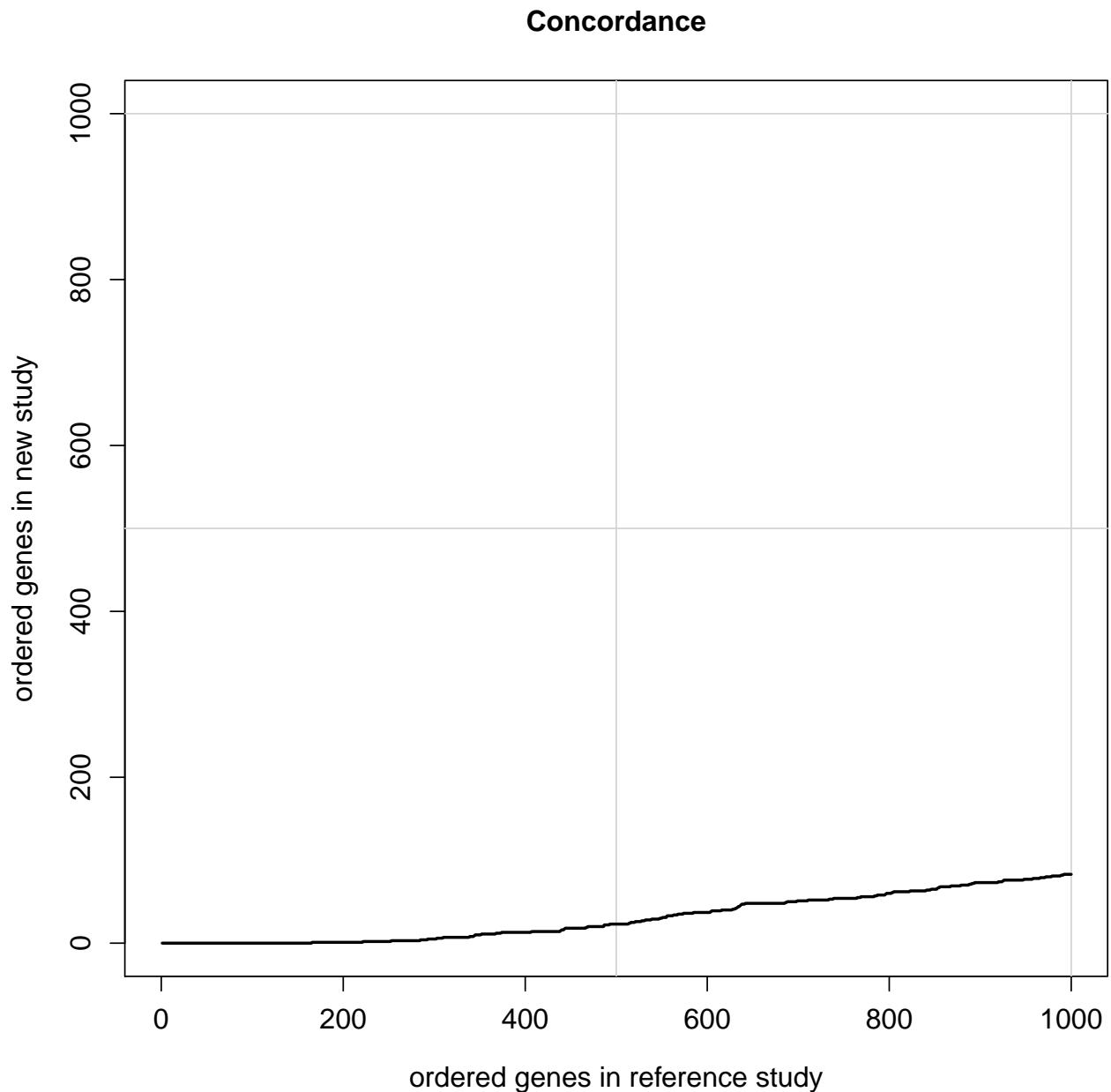
```



```

## 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

```

```

## 1132 17238

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

## 
## FALSE TRUE
## 636 17238

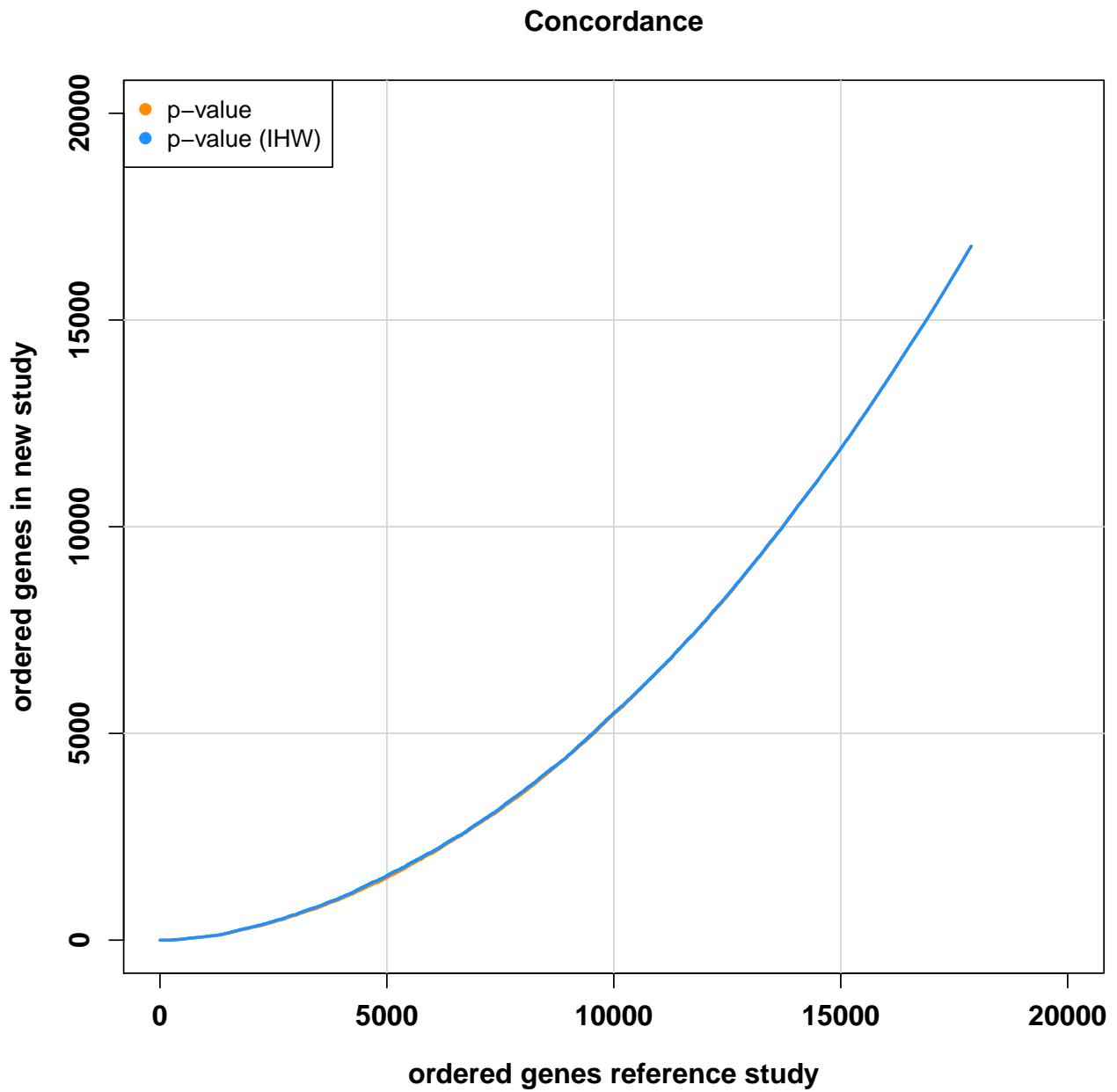
table(names(p.mod1.sort) %in% names(p.ihw.sort))

## 
## TRUE
## 18370

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)"))

```



```

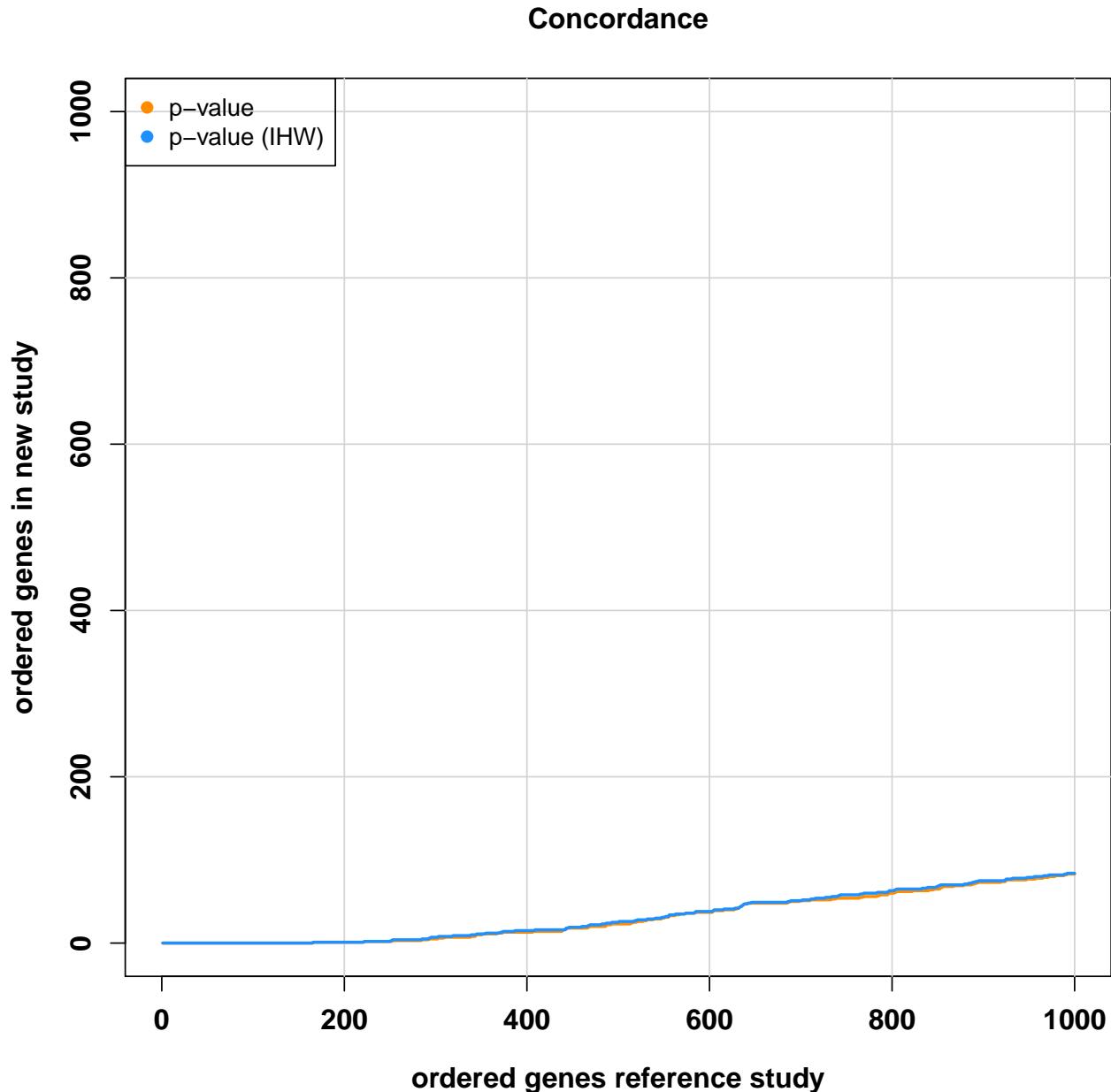
## top 1000 genes
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
plot(seq(1:length(p.mod2.sort[1:1000])), conc_p.mod[1:1000],
     type = 'l', las = 0,
     xlim = c(0, 1000),
     ylim = c(0, 1000),
     xlab = 'ordered genes reference study',
     ylab = 'ordered genes in new study',
     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 (Ignatiadis, Klaus, Zaugg, and Huber, 2015)
- *BiocStyle* (R Core Team, 2016)
- *derfinder* (Allaire, Cheng, Xie, McPherson, et al., 2016)
- *devtools* (with contributions from Andrew J. Bass, Dabney, and Robinson, 2015)
- *edgeR* (Bengtsson, 2016)

- *IHW* (Boettiger, 2016)
- *knitcitations* (Collado-Torres, Nellore, Frazee, Wilks, et al., 2016)
- *matrixStats* (Collado-Torres, Nellore, and Leek, 2016)
- *qvalue* (Frazee, 2016)
- *recount* (Morgan, Obenchain, Hester, and Pagès, 2016)
- *rmarkdown* (Oleś, Morgan, and Huber, 2016)
- *RSkittleBrewer* (Robinson, McCarthy, and Smyth, 2010)
- *SummarizedExperiment* (Law, Chen, Shi, and Smyth, 2014)
- *limma* (Wickham and Chang, 2016)

Bibliography file

- [1] J. Allaire, J. Cheng, Y. Xie, J. McPherson, et al. *rmarkdown*: Dynamic Documents for R. R package version 1.0. 2016. URL: <https://CRAN.R-project.org/package=rmarkdown>.
- [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 2.5.2. 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 0.50.2. 2016. URL: <https://CRAN.R-project.org/package=matrixStats>.
- [4] C. Boettiger. *knitcitations*: Citations for ‘Knitr’ Markdown Files. R package version 1.0.7.1. 2016. URL: <https://github.com/cboettig/knitcitations>.
- [5] L. Collado-Torres, A. Nellore, A. C. Frazee, C. Wilks, et al. “Flexible expressed region analysis for RNA-seq with derfinder”. In: bioRxiv (2016). DOI: 10.1101/015370. URL: <http://biorkxiv.org/content/early/2016/05/07/015370>.
- [6] L. Collado-Torres, A. Nellore and J. T. Leek. *recount*: Explore and download data from the recount project. R package version 0.99.0. 2016. URL: <https://github.com/leekgroup/recount>.
- [7] A. Frazee. *RSkittleBrewer*: Fun with R Colors. R package version 1.1. 2016. URL: <https://github.com/alyssafrzee/RSkittleBrewer>.
- [8] N. Ignatiadis, B. Klaus, J. Zaugg and W. Huber. “Data-driven hypothesis weighting increases detection power in big data analytics”. In: bioRxiv (2015). DOI: 10.1101/034330. URL: <http://dx.doi.org/10.1101/034330>.
- [9] C. Law, Y. Chen, W. Shi and G. Smyth. “Voom: precision weights unlock linear model analysis tools for RNA-seq read counts”. In: Genome Biology 15 (2014), p. R29.
- [10] M. Morgan, V. Obenchain, J. Hester and H. Pagès. *SummarizedExperiment*: SummarizedExperiment container. R package version 1.3.7. 2016.
- [11] A. Oleś, M. Morgan and W. Huber. *BiocStyle*: Standard styles for vignettes and other Bioconductor documents. R package version 2.1.19. 2016. URL: <https://github.com/Bioconductor/BiocStyle>.
- [12] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: <https://www.R-project.org/>.
- [13] M. D. Robinson, D. J. McCarthy and G. K. Smyth. “edgeR: a Bioconductor package for differential expression analysis of digital gene expression data”. In: Bioinformatics 26 (2010), pp. -1.
- [14] H. Wickham and W. Chang. *devtools*: Tools to Make Developing R Packages Easier. R package version 1.12.0. 2016. URL: <https://CRAN.R-project.org/package=devtools>.

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

```
## Time difference of 1.327059 mins
```

```

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

## 2016-07-22 20:26:27

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

## Session info -----
## setting value
## version R version 3.3.0 (2016-05-03)
## system x86_64, darwin13.4.0
## ui      RStudio (0.99.902)
## language (EN)
## collate en_US.UTF-8
## tz      America/New_York
## date    2016-07-22

## Packages -----
## package      * version  date      source
## bibtex       0.4.0    2014-12-31 CRAN (R 3.3.0)
## Biobase      * 2.33.0   2016-05-27 Bioconductor
## BiocGenerics * 0.19.2   2016-07-08 Bioconductor
## BiocStyle     * 2.1.19   2016-07-11 Bioconductor
## bitops        1.0-6    2013-08-17 CRAN (R 3.3.0)
## colorspace    1.2-6    2015-03-11 CRAN (R 3.1.3)
## devtools      1.12.0   2016-06-24 CRAN (R 3.3.0)
## digest        0.6.9    2016-01-08 CRAN (R 3.3.0)
## edgeR         * 3.15.2   2016-07-08 Bioconductor
## evaluate      0.9      2016-04-29 CRAN (R 3.3.0)
## fdrtool       1.2.15   2015-07-08 CRAN (R 3.3.0)
## formatR       1.4      2016-05-09 CRAN (R 3.3.0)
## GenomeInfoDb  * 1.9.4    2016-07-14 Bioconductor
## GenomicRanges * 1.25.9   2016-06-26 Bioconductor
## ggplot2        2.1.0    2016-03-01 CRAN (R 3.2.4)
## gtable        0.2.0    2016-02-26 CRAN (R 3.2.3)
## htmltools      0.3.5    2016-03-21 CRAN (R 3.3.0)
## httr          1.2.1    2016-07-03 CRAN (R 3.3.0)
## IHW           * 1.1.3    2016-06-17 Bioconductor
## IRanges        * 2.7.11   2016-06-22 Bioconductor
## knitr          * 1.0.7.1  2016-06-06 Github (cboettig/knitr@76c128a)
## knitr          1.13     2016-05-09 CRAN (R 3.3.0)
## lattice        0.20-33   2015-07-14 CRAN (R 3.3.0)
## limma          * 3.29.16  2016-07-21 Bioconductor
## locfit         1.5-9.1   2013-04-20 CRAN (R 3.1.0)
## lpsymphony     1.1.2    2016-05-27 Bioconductor (R 3.3.0)
## lubridate      1.5.6    2016-04-06 CRAN (R 3.3.0)
## magrittr       1.5      2014-11-22 CRAN (R 3.3.0)
## Matrix         1.2-6    2016-05-02 CRAN (R 3.3.0)

```

```

##  matrixStats           * 0.50.2   2016-04-24 CRAN (R 3.2.5)
##  memoise                1.0.0    2016-01-29 CRAN (R 3.2.3)
##  munsell                 0.4.3    2016-02-13 CRAN (R 3.2.3)
##  plyr                     1.8.4    2016-06-08 CRAN (R 3.3.0)
##  qvalue                   * 2.5.2    2016-05-27 Bioconductor
##  R6                        2.1.2    2016-01-26 CRAN (R 3.2.3)
##  Rcpp                      0.12.6   2016-07-19 CRAN (R 3.3.0)
##  RCurl                     1.95-4.8 2016-03-01 CRAN (R 3.2.4)
##  recount                  * 0.99.0   2016-06-06 Github (leekgroup/recount@bfa4681)
##  RefManageR                0.10.13   2016-04-04 CRAN (R 3.3.0)
##  reshape2                  1.4.1    2014-12-06 CRAN (R 3.1.2)
##  RJSONIO                   1.3-0     2014-07-28 CRAN (R 3.2.0)
##  rmarkdown                  * 1.0      2016-07-08 CRAN (R 3.3.0)
##  RSkittleBrewer             * 1.1      2016-06-07 Github (alyssafrazee/RSkittleBrewer@230d1d0)
##  rstudioapi                  0.6      2016-06-27 CRAN (R 3.3.0)
##  S4Vectors                  * 0.11.10   2016-07-21 Bioconductor
##  scales                      0.4.0    2016-02-26 CRAN (R 3.2.3)
##  slam                         0.1-35   2016-06-14 CRAN (R 3.3.0)
##  stringi                     1.1.1    2016-05-27 CRAN (R 3.3.0)
##  stringr                     1.0.0    2015-04-30 CRAN (R 3.3.0)
##  SummarizedExperiment        * 1.3.7    2016-07-08 Bioconductor
##  withr                        1.0.2    2016-06-20 CRAN (R 3.3.0)
##  XML                          3.98-1.4 2016-03-01 CRAN (R 3.2.4)
##  XVector                     0.13.6   2016-07-08 Bioconductor
##  yaml                         2.1.13   2014-06-12 CRAN (R 3.3.0)
##  zlibbioc                    1.19.0   2016-05-27 Bioconductor

```