recount (overlay two studies)

Kai Kammers and Shannon Ellis

June 09, 2016

Contents

Here we label in the following way:

- study1 = SRP019936 (that is the 'new' study)
- study2 = SRP032789 (this is the study that we used for gene, exon, and expressed region analyses)

Load R-packages

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

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
```

```
## 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
```

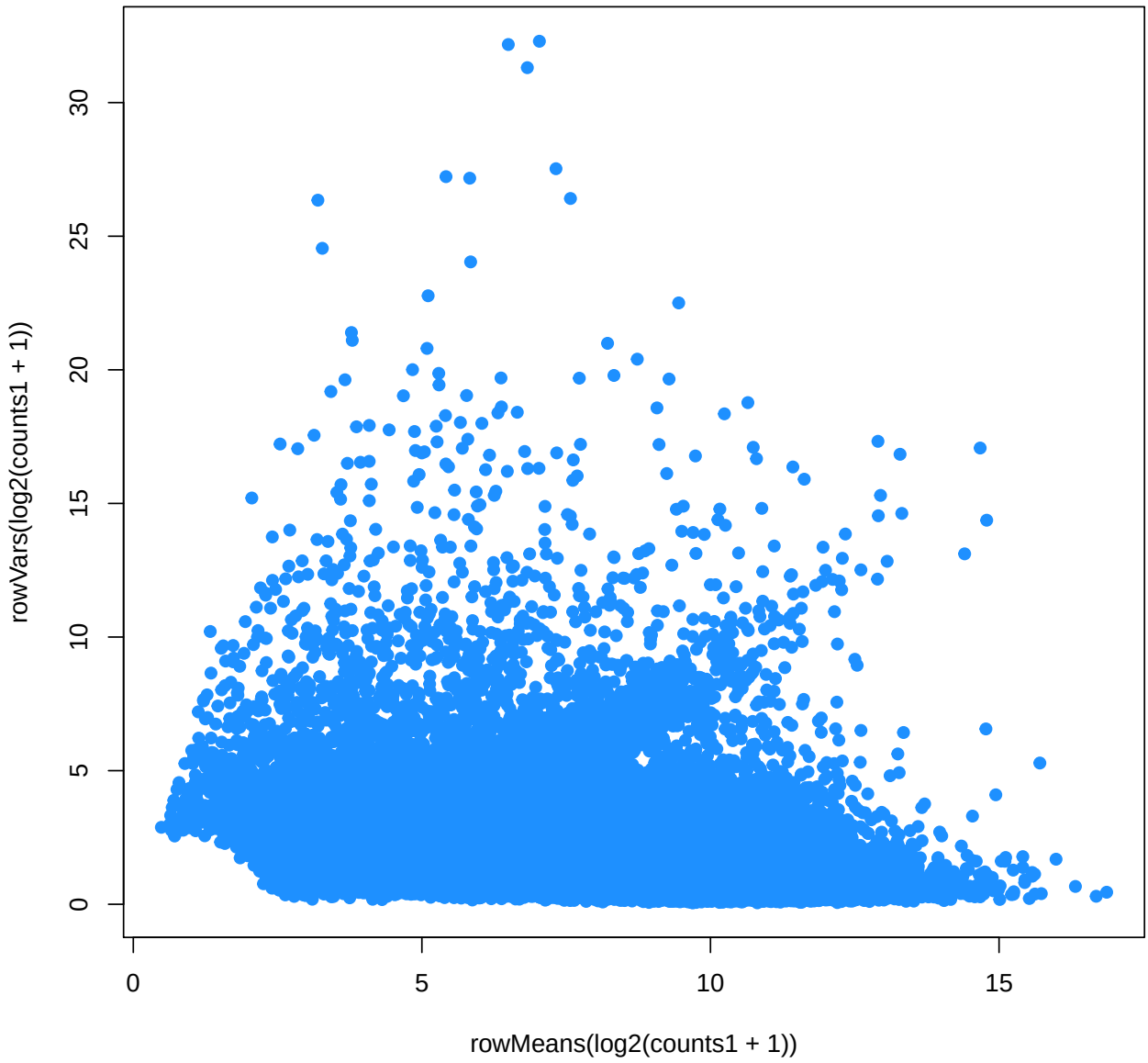
```
## obtain count matrix
counts1 <- assays(rse1)$counts

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

```
## [1] 18333    16
```

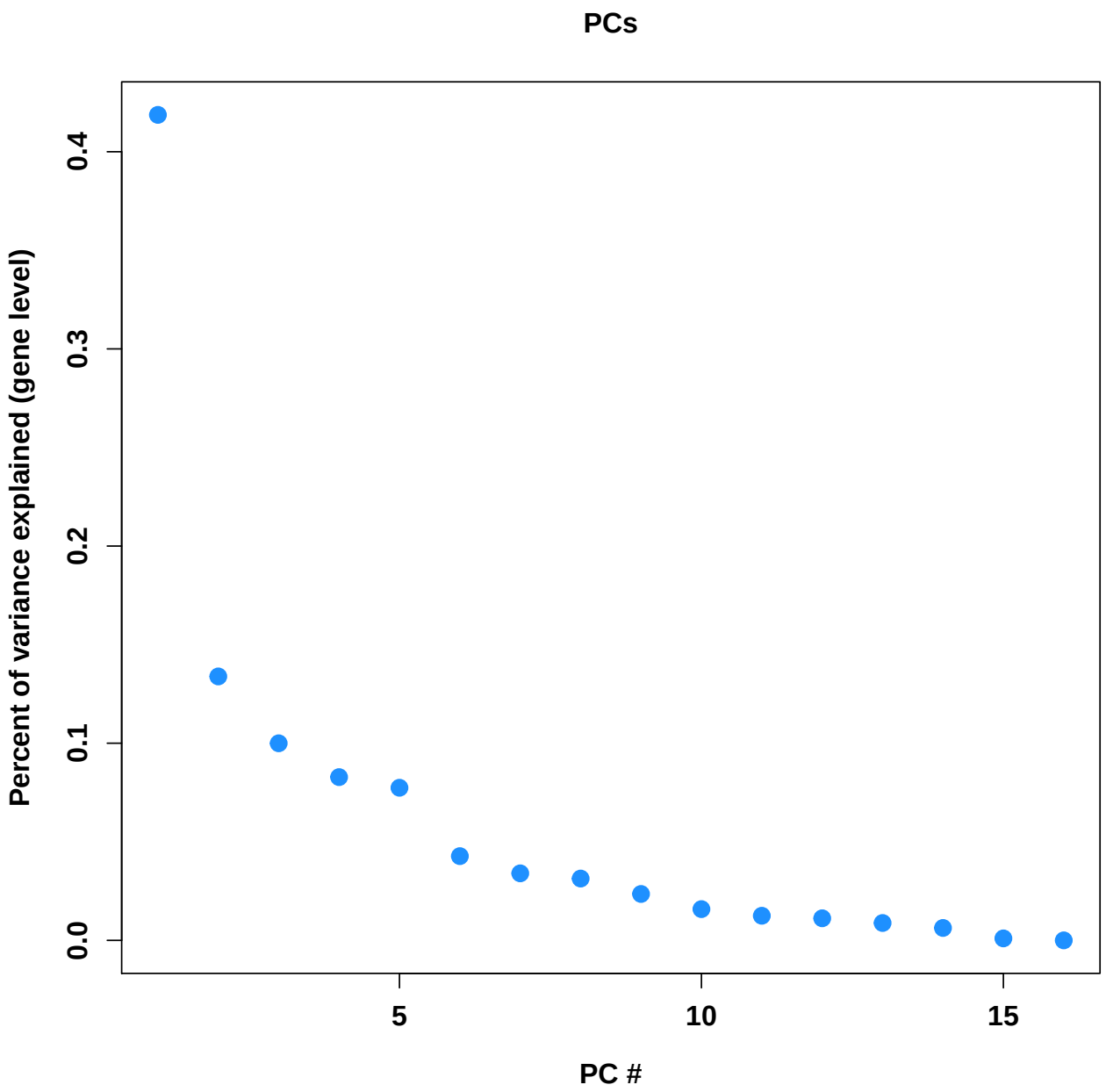
```
## set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse1$group))

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

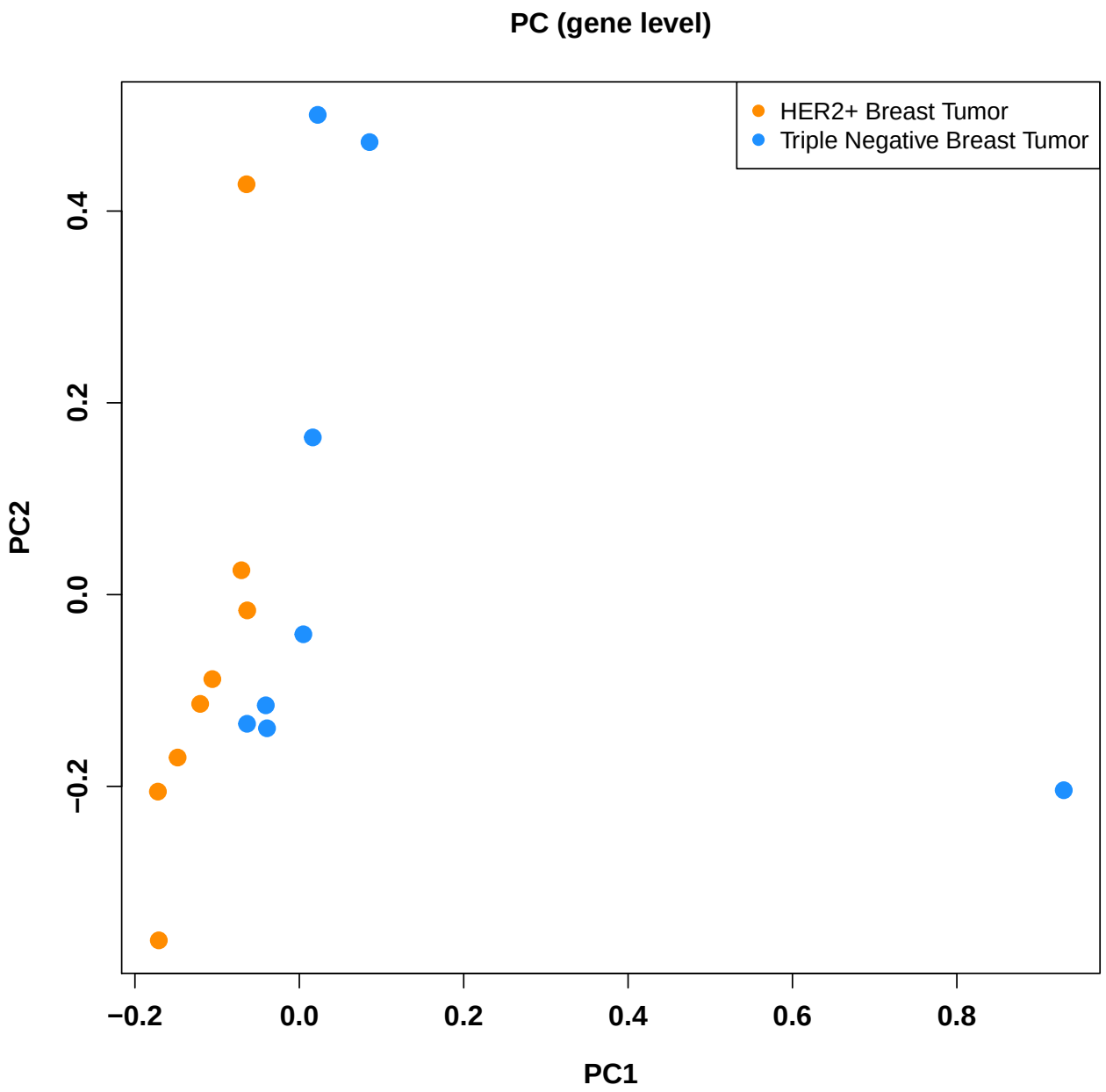



```
## calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))

## 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))))
```




```
## 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
```

```
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
```

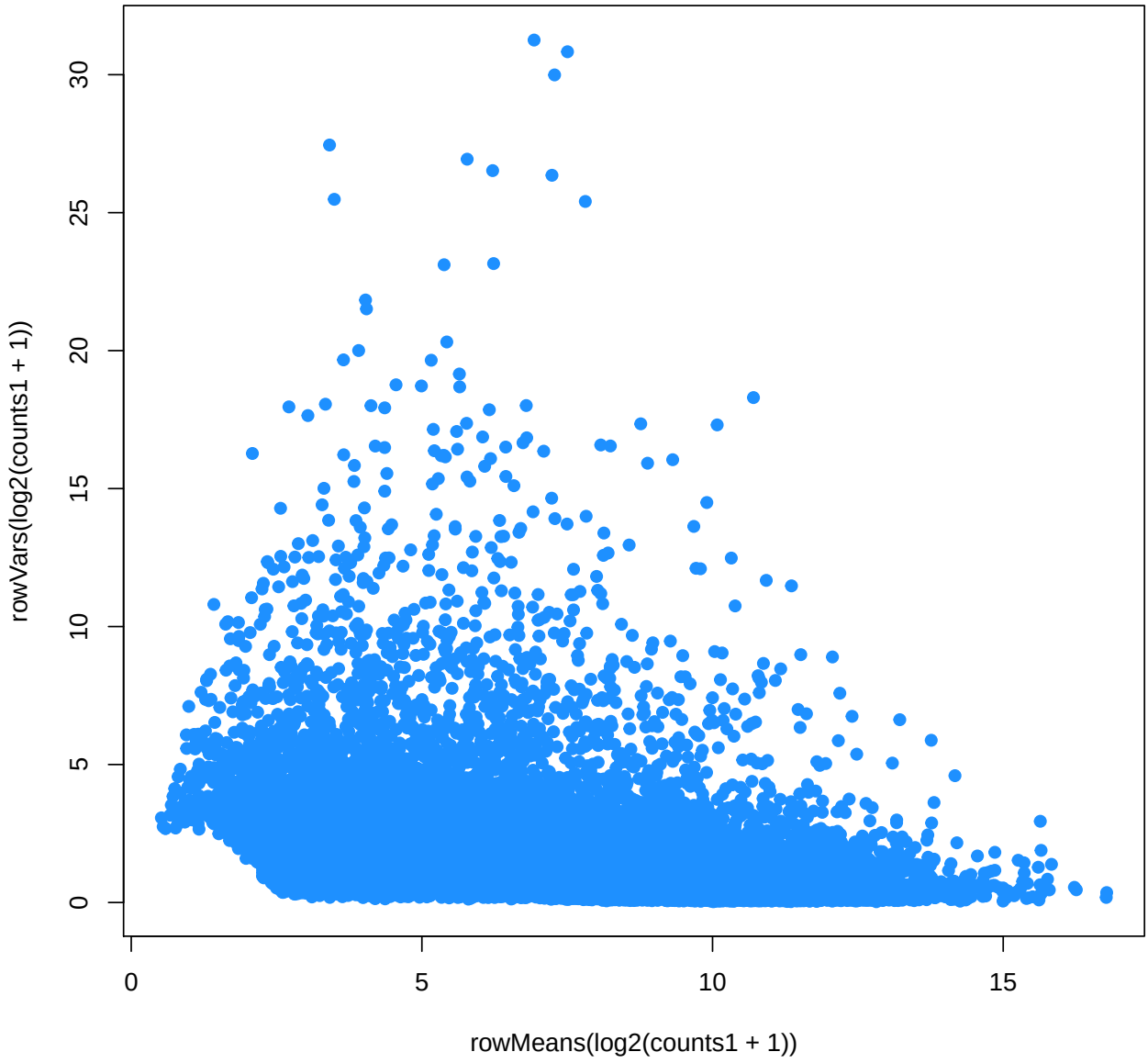
```
## obtain count matrix
counts1 <- assays(rse1)$counts

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

```
## [1] 18370    15
```

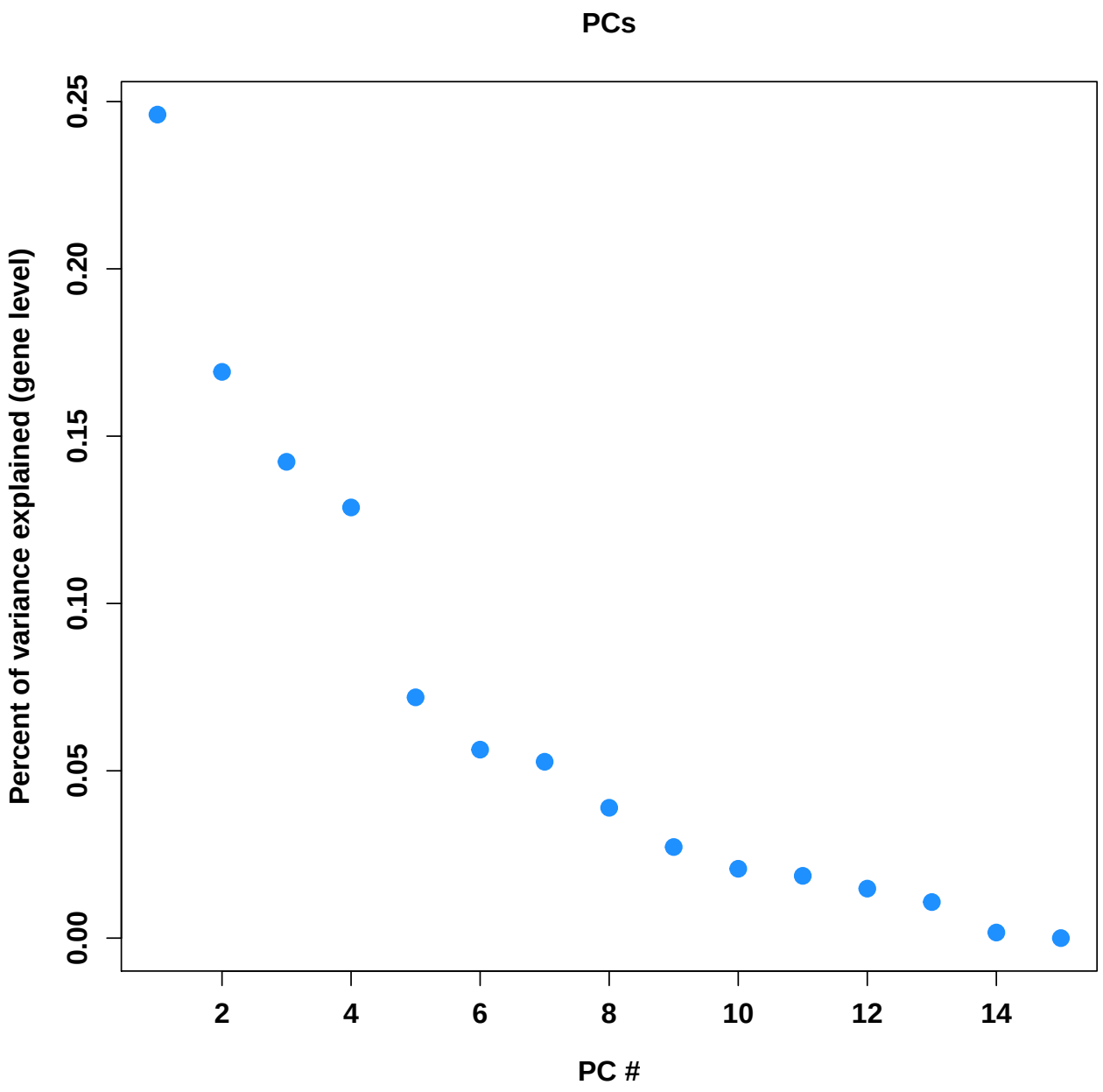
```
## set colors
trop <- RSkittleBrewer('tropical')[c(1, 2)]
cols <- as.numeric(as.factor(rse1$group))
```

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

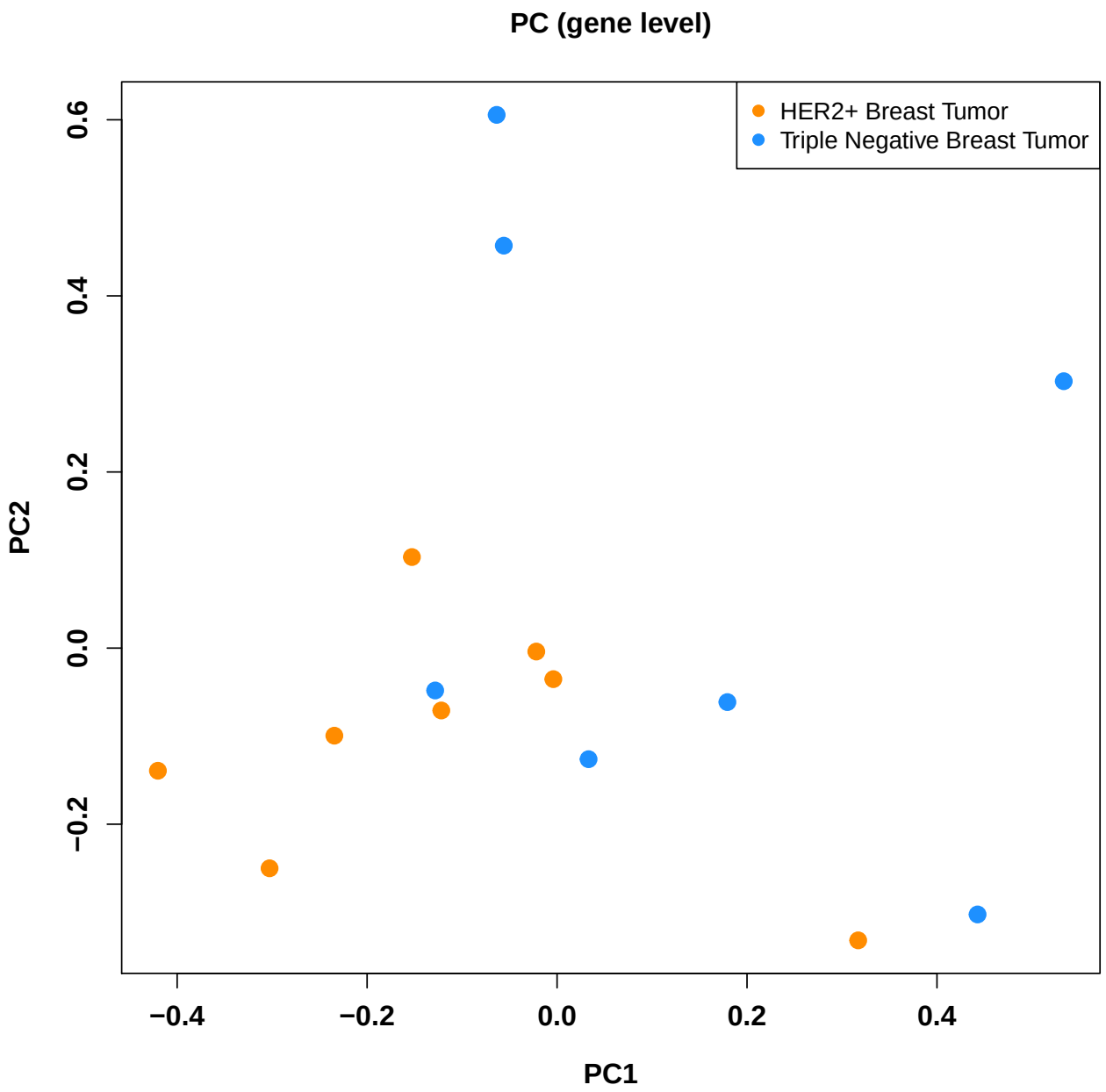



```
## calculate PCs with svd function
expr.pca <- svd(counts1 - rowMeans(counts1))

## 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))))
```

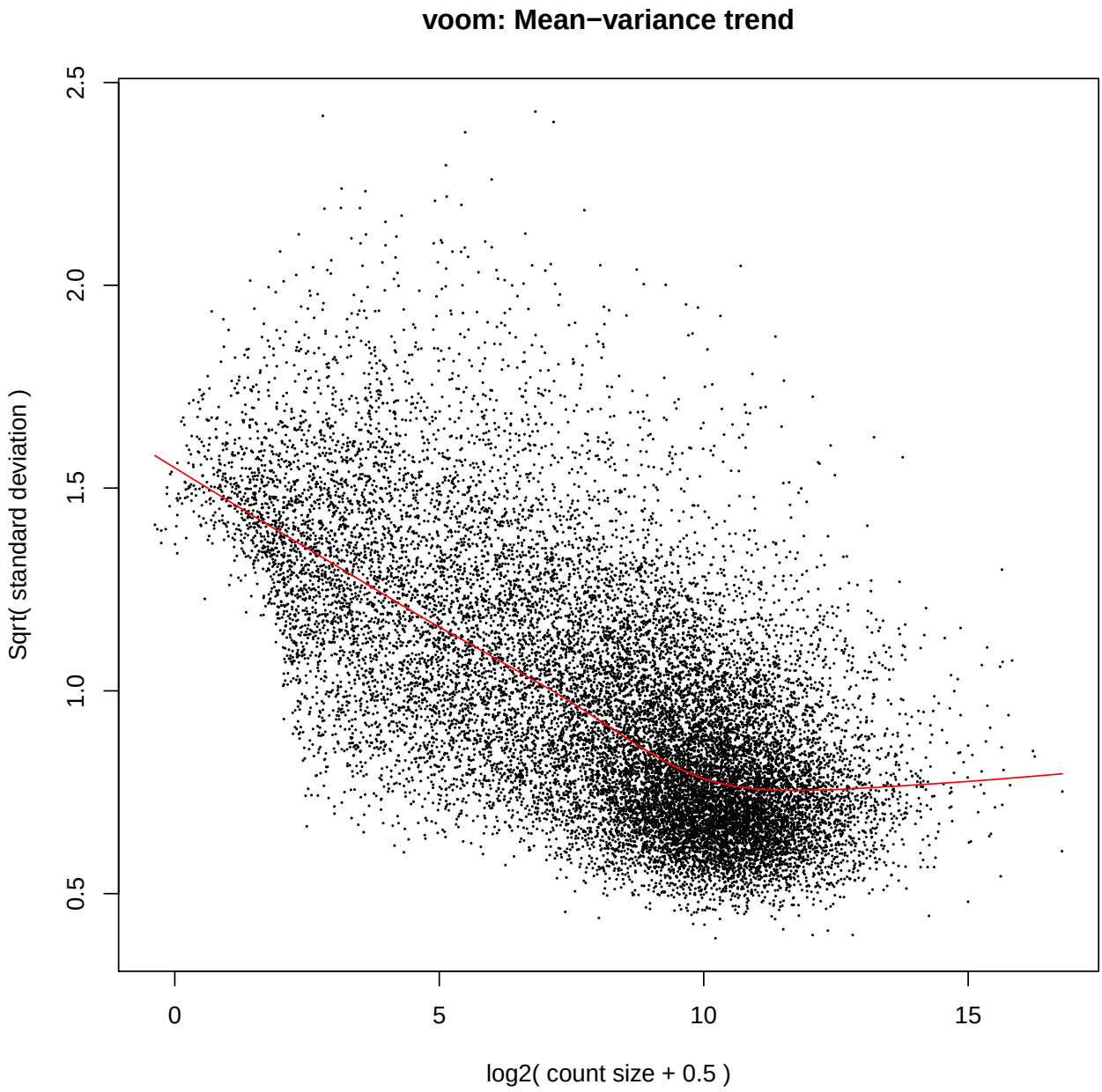



```
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse1$group)
design
```

```
##    (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
```

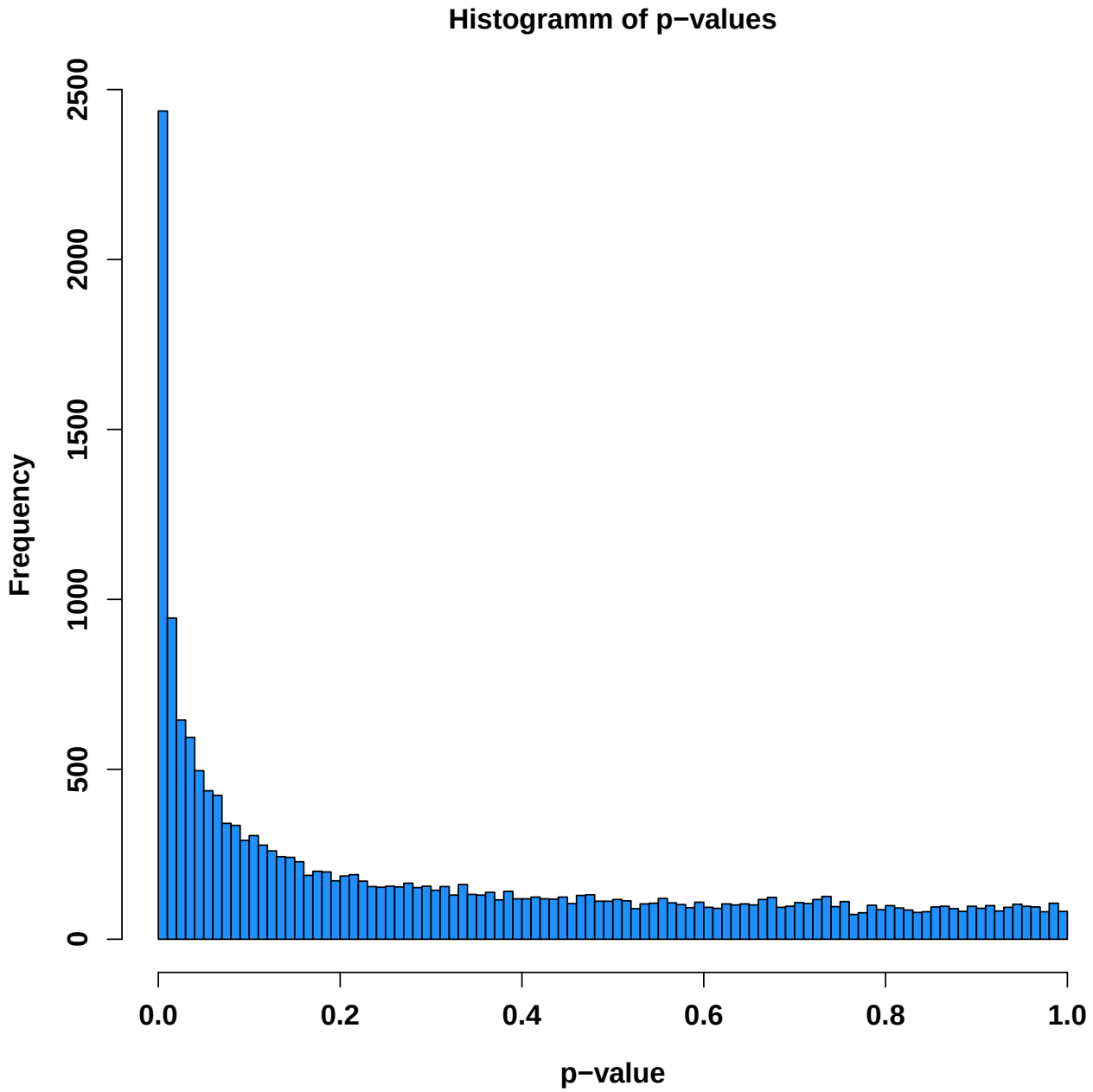
```
## 11           1                                    0
## 12           1                                    0
## 13           1                                    1
## 14           1                                    1
## 15           1                                    1
## attr(,"assign")
## [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)
```



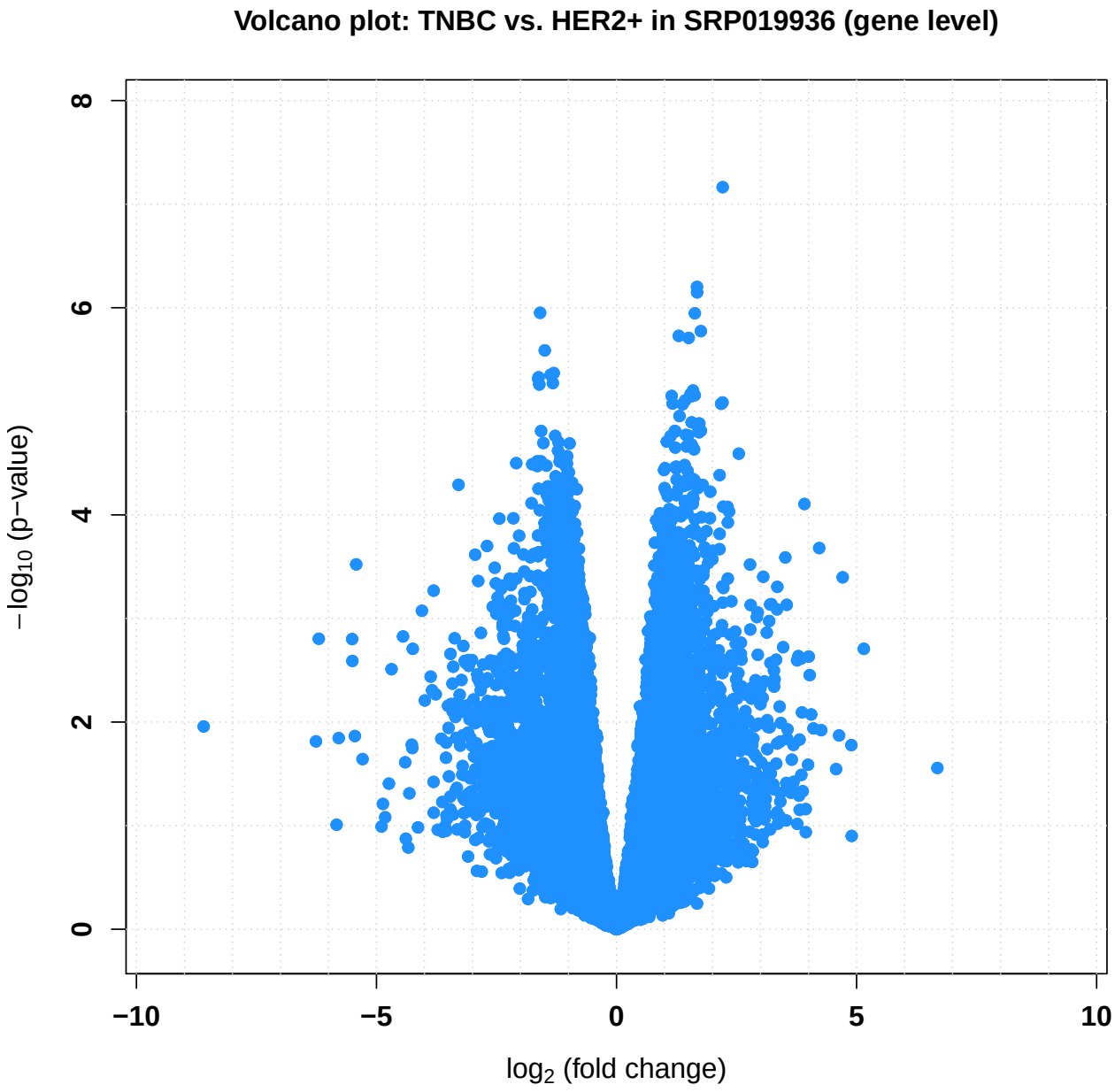
```
fit <- lmFit(v, design)
fit <- eBayes(fit)
log2FC1 <- fit$coefficients[, 2]
t.mod1 <- fit$t[, 2]
p.mod1 <- fit$p.value[, 2]
q.mod1 <- qvalue(p.mod1)$q

## Histogram
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)')
```



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
```

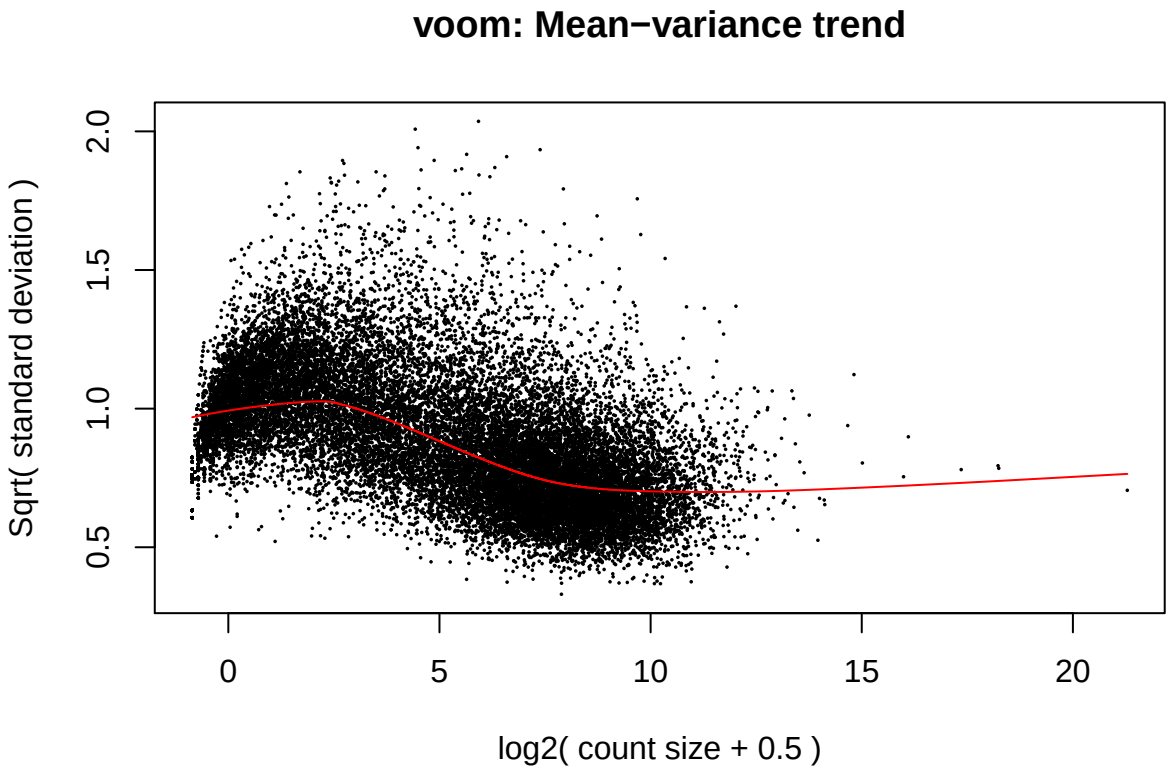
```
## obtain count matrix without filtering
counts2 <- assays(rse2)$counts
dim(counts2)
```

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

```
## 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


## some summary statistics
sum(p.mod1 <= 0.05)
```

```
## [1] 5117
```

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

```
## [1] 3197
```

```
sum(p.mod2 <= 0.05)
```

```
## [1] 5856
```

```
sum(q.mod2 <= 0.05)
```

```
## [1] 2941
```

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

## use p-values from study 2 as weights for study 1
ihw_res <- ihw(p.mod1 ~ t.mod2, alpha = 0.05)
rejections(ihw_res)
```

```
## [1] 1673
```

```
head(ihw_res@df)
```

```
##                pvalue adj_pvalue    weight weighted_pvalue group  covariate
## 1          0.53356404 0.90904485 0.6864193      0.77731499    11  1.8505302
## 10         0.46942017 0.68819184 0.9667694      0.48555550     3 -1.3749147
## 100        0.58806782 0.80697332 0.9251430      0.63565073     8  0.7023154
## 1000       0.87146306 1.00000000 0.9337000      0.93334373     9  0.8411674
## 100008589 0.01187256 0.08334221 0.9843144      0.01206175     5 -0.7600378
## 100009676 0.22427859 0.51582173 0.7515267      0.29843064    10  1.3856221
##            fold
## 1             2
## 10            1
## 100           3
## 1000          2
## 100008589     3
## 100009676     3
```

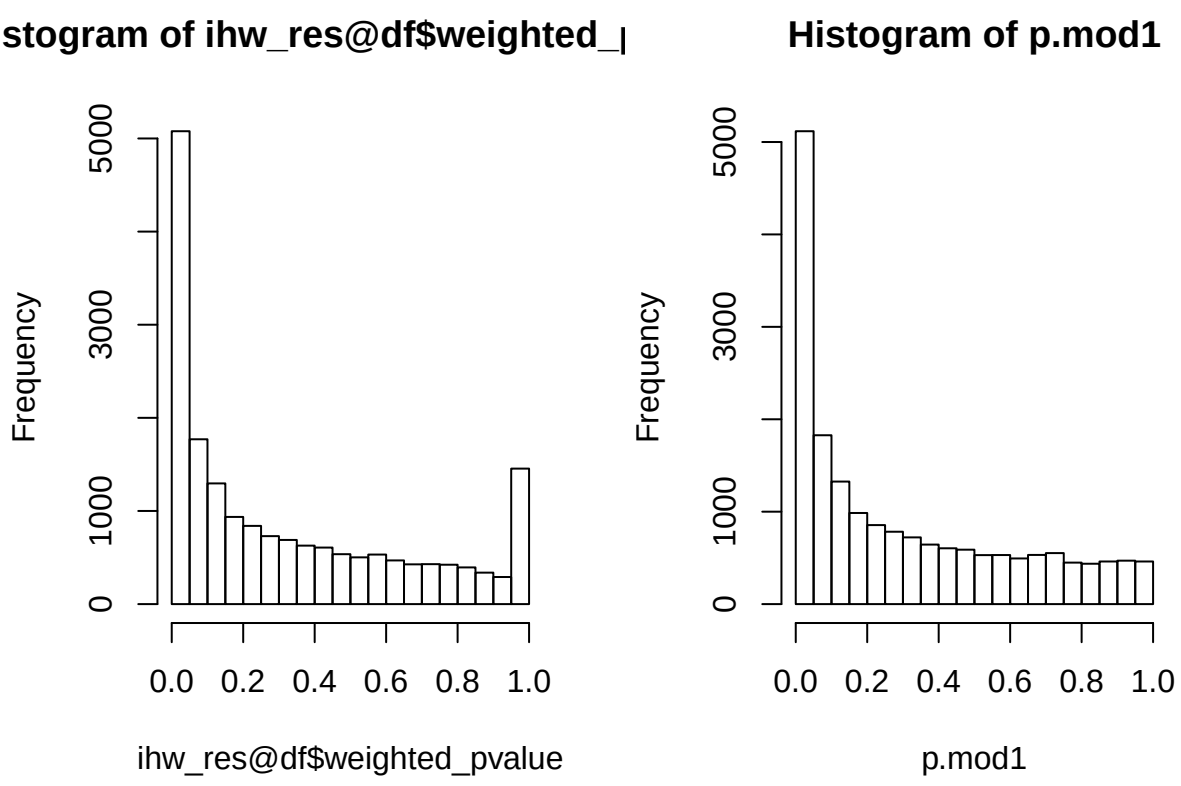
```
padj_bh_ihw <- p.adjust(ihw_res@df$weighted_pvalue, method = 'BH')
padj_bh <- p.adjust(p.mod1, method = 'BH')
sum(padj_bh_ihw <= 0.05, na.rm = TRUE)
```

```
## [1] 1673
```

```
sum(padj_bh <= 0.05, na.rm = TRUE)
```

```
## [1] 1649
```

```
par(mfrow = c(1, 2))
hist(ihw_res@df$weighted_pvalue)
hist(p.mod1)
```



```
## use counts as weights for study 1
counts1.mean <- apply(counts1, 1, mean)
ihw_res <- ihw(p.mod1 ~ counts1.mean, alpha = 0.05)
rejections(ihw_res)
```

```
## [1] 1917
```

```
head(ihw_res@df)
```

```
##               pvalue adj_pvalue    weight weighted_pvalue group
## 1         0.53356404 1.00000000 0.4541001      1.00000000     5
## 10        0.46942017 1.00000000 0.0000000      1.00000000     1
## 100       0.58806782 1.00000000 0.5418122      1.00000000     5
## 1000      0.87146306 1.00000000 0.1514423      1.00000000     4
## 100008589 0.01187256 0.06718263 1.3343024      0.00889795    12
## 100009676 0.22427859 0.98906344 0.3275580      0.68469895     4
##              covariate fold
## 1           395.133333    2
## 10            7.866667    1
## 100         412.200000    3
## 1000        118.333333    2
## 100008589 48625.800000    3
## 100009676   180.400000    3
```

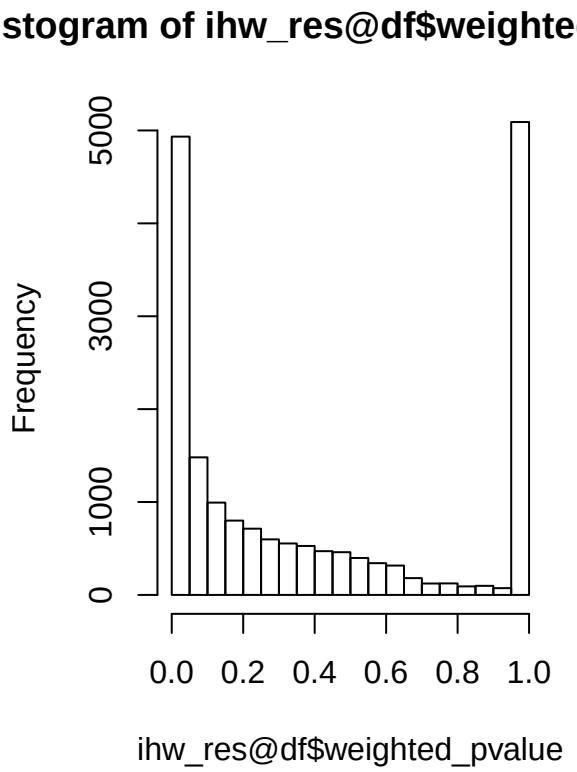
```
padj_bh_ihw <- p.adjust(ihw_res@df$weighted_pvalue, method = 'BH')
padj_bh <- p.adjust(p.mod1, method = 'BH')
sum(padj_bh_ihw <= 0.05, na.rm = TRUE)
```

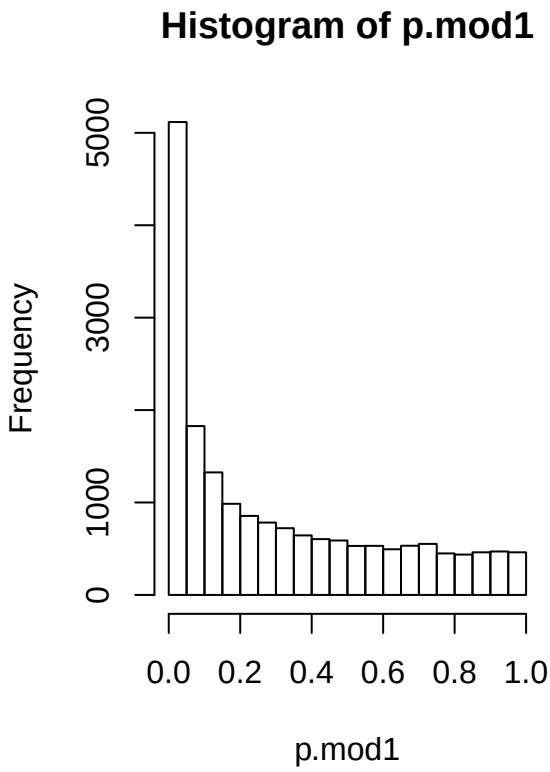
```
## [1] 1917
```

```
sum(padj_bh <= 0.05, na.rm = TRUE)
```

```
## [1] 1649
```

```
par(mfrow = c(1, 2))
hist(ihw_res@df$weighted_pvalue)
hist(p.mod1)
```





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
```

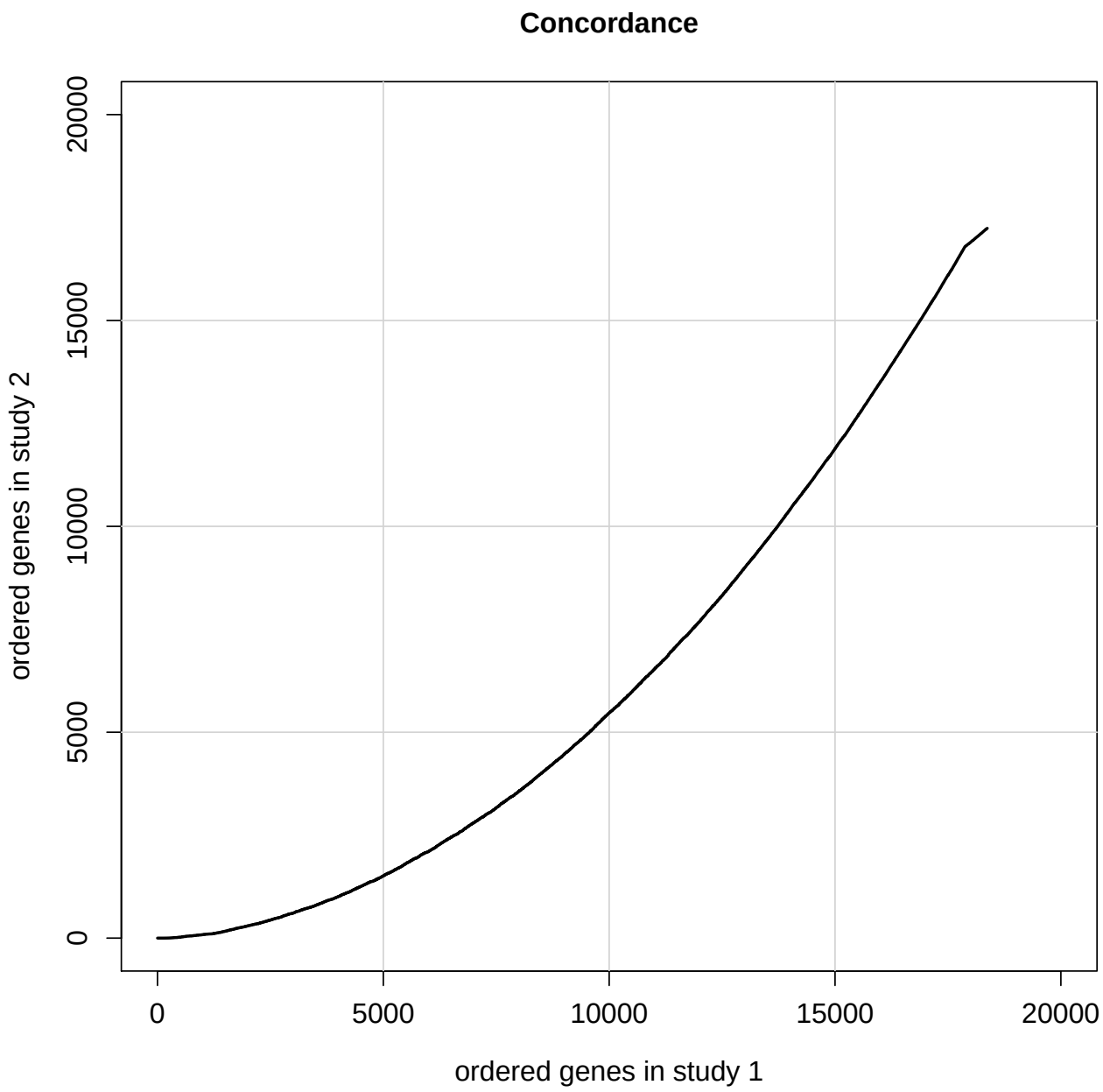
```
table(names(p.mod2.sort) %in% names(p.mod1.sort))
```

```
##
```

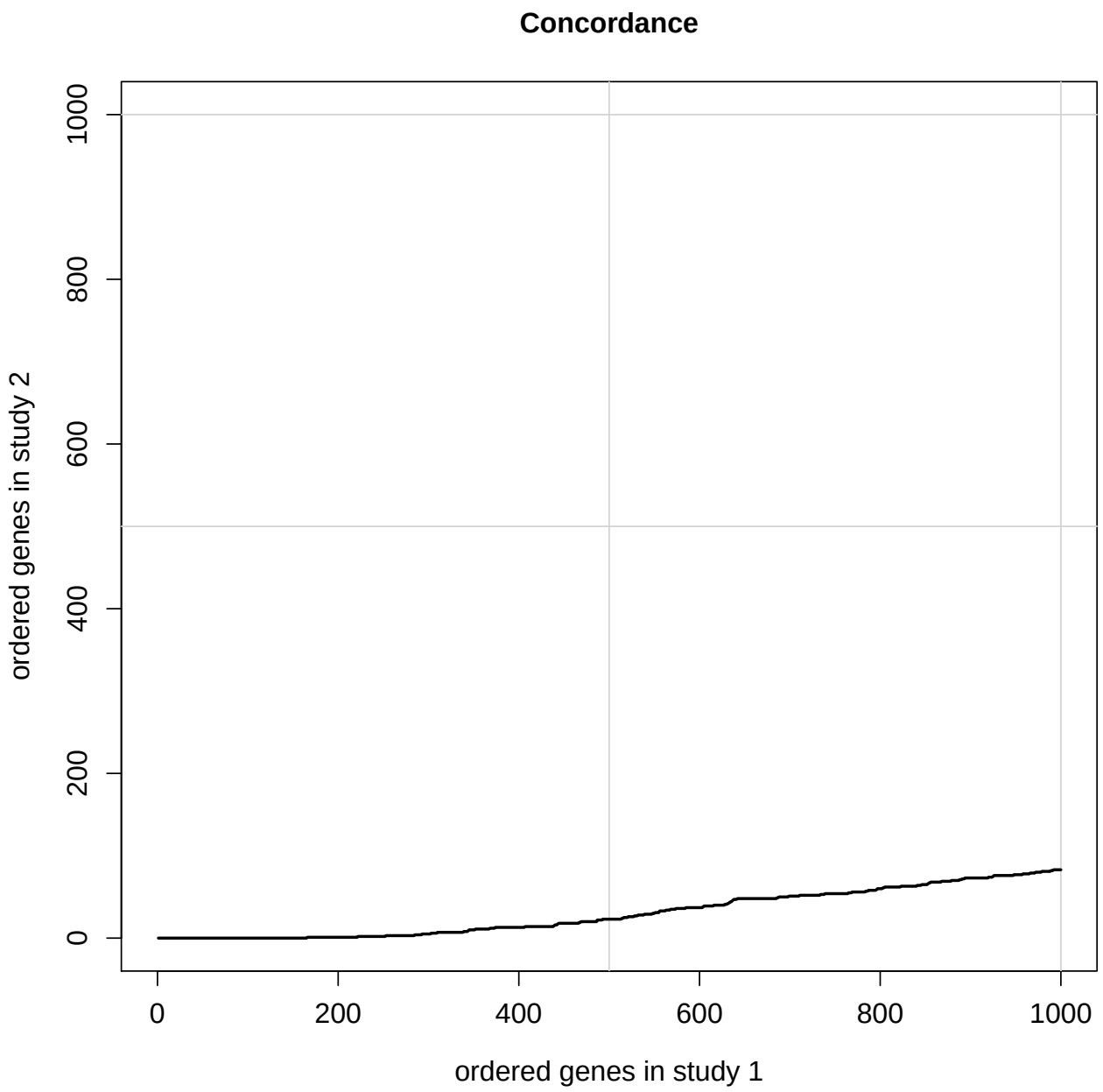
```
## FALSE  TRUE
##   636 17238
```

```
conc <- NULL
for(i in 1:length(p.mod1.sort)){
    conc[i] <- sum(names(p.mod1.sort)[1:i] %in% names(p.mod2.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.mod1.sort)), conc,
     type = 'l', las = 0,
     xlim = c(0, 20000),
     ylim = c(0, 20000),
     xlab = 'ordered genes in study 1',
     ylab = 'ordered genes in study 2',
     main = 'Concordance')
for(k in 1:3){
    abline(v = k * 5000, cex = 0.5, col = 'lightgrey')
    abline(h = k * 5000, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod1.sort)), conc, col = 'black', lwd = 2)
```




```
## 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 study 1',
     ylab = 'ordered genes in study 2',
     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 and raw p-values

```
## use t-values from study 2 as weights for study 1
ihw_res <- ihw(p.mod1 ~ t.mod2, alpha = 0.05)

## sort p-values
p.mod1.sort <- p.mod1[order(p.mod1)]
p.mod2 <- ihw_res@df$weighted_pvalue
names(p.mod2) <- rownames(ihw_res@df)
p.mod2.sort <- p.mod2[order(p.mod2)]

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

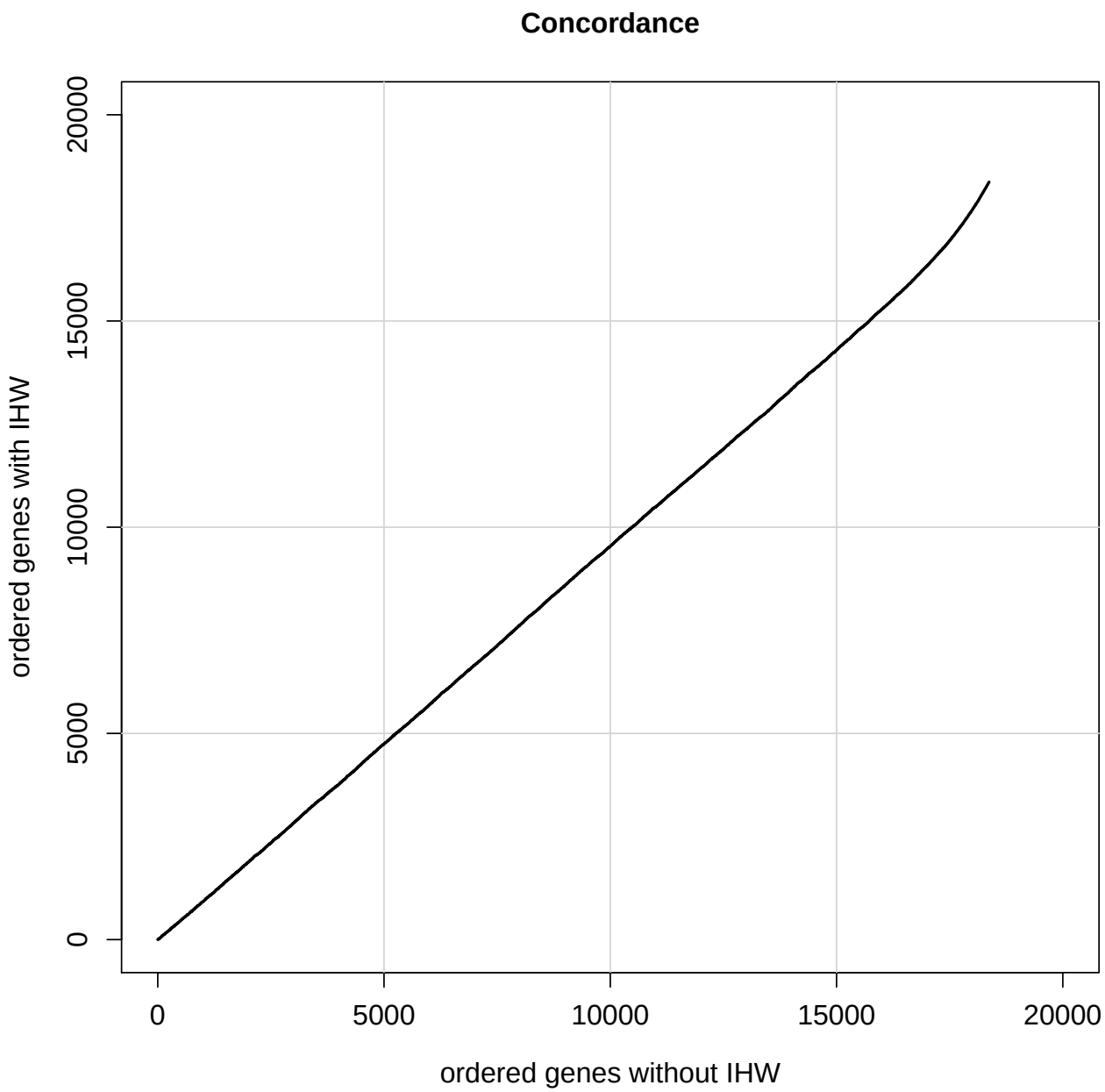
```
##
##  TRUE
## 18370
```

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

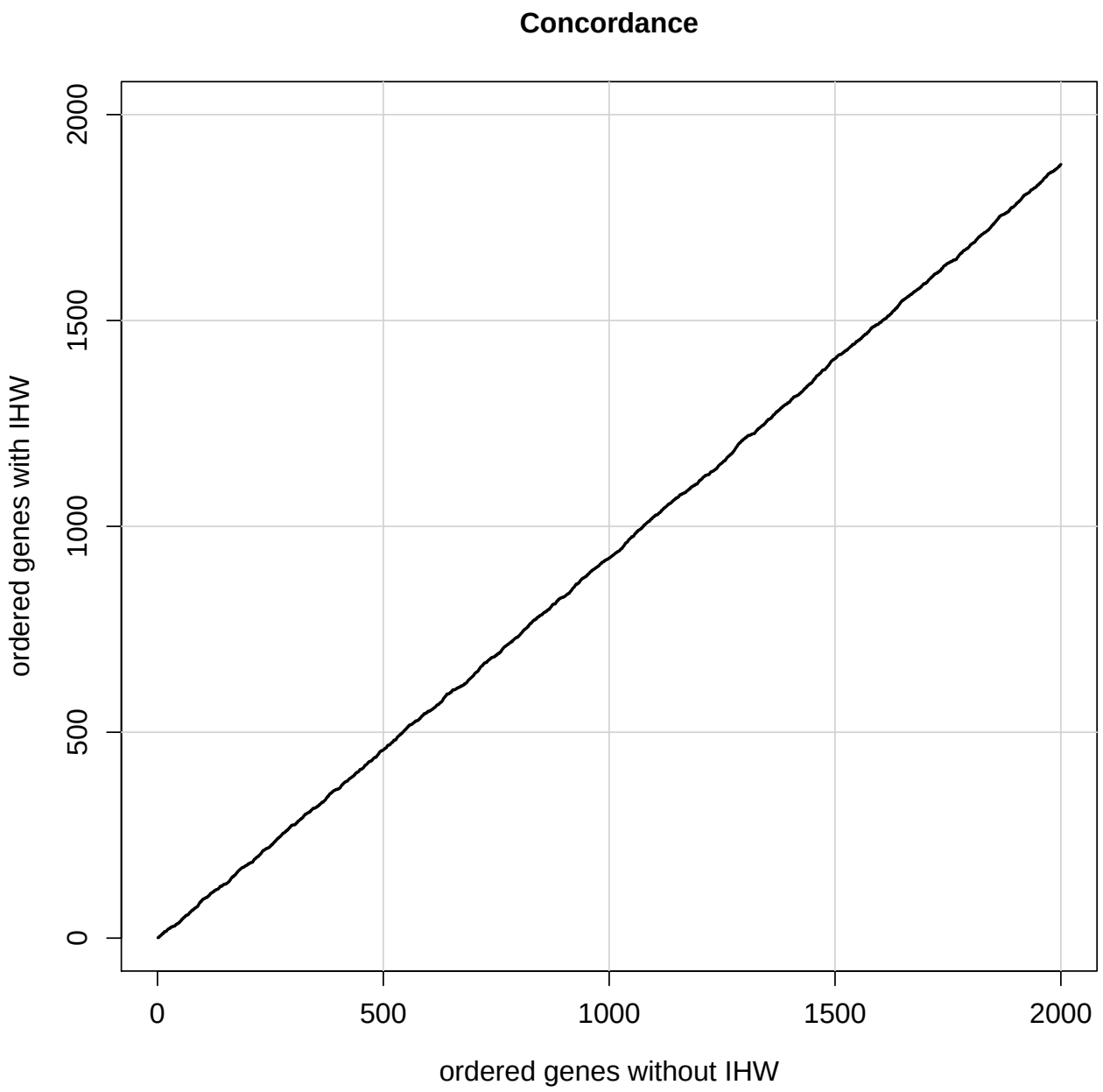
```
##
##  TRUE
## 18370
```

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

## 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.mod1.sort)), conc,
     type = 'l', las = 0,
     xlim = c(0, 20000),
     ylim = c(0, 20000),
     xlab = 'ordered genes without IHW',
     ylab = 'ordered genes with IHW',
     main = 'Concordance')
for(k in 1:3){
    abline(v = k * 5000, cex = 0.5, col = 'lightgrey')
    abline(h = k * 5000, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:length(p.mod1.sort)), conc, col = 'black', lwd = 2)
```




```
## top 2000 genes
par(mfrow = c(1, 1), font.lab = 1.5, cex.lab = 1.2, font.axis = 1.5, cex.axis = 1.2)
plot(seq(1:2000), conc[1:2000],
     type = 'l', las = 0,
     xlim = c(0, 2000),
     ylim = c(0, 2000),
     xlab = 'ordered genes without IHW',
     ylab = 'ordered genes with IHW',
     main = 'Concordance')
for(k in 1:4){
    abline(v = k * 500, cex = 0.5, col = 'lightgrey')
    abline(h = k * 500, cex = 0.5, col = 'lightgrey')
}
lines(seq(1:2000), conc[1:2000], col = 'black', lwd = 2)
```



Reproducibility

This analysis report was made possible thanks to:

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

- *recount* (Collado-Torres and Leek, 2016)
- *rmarkdown* (Allaire, Cheng, Xie, McPherson, et al., 2016)
- *RSkittleBrewer* (Frazee, 2016)
- *SummarizedExperiment* (Morgan, Obenchain, Hester, and Pagès, 2016)
- *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 0.9.6. 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. 2015. URL: https://CRAN.R-project.org/package=knitcitations.

[5] L. Collado-Torres and J. T. Leek. recount: Explore and download data from the recount project. R package version 0.99.10. 2016. URL: https://github.com/leekgroup/recount.

[6] 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://biorxiv.org/content/early/2016/05/07/015370.

[7] A. Frazee. RSkittleBrewer: Fun with R Colors. R package version 1.1. 2016. URL: https://github.com/alyssafrazee/RSkittlel

[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 1.3.4. 2016.

[11] A. Oleś, M. Morgan and W. Huber. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.1.6. 2016. URL: https://github.com/Bioconductor/BiocStyle.

[12] R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: https://www.R-project.org/.

[13] M. D. Robinson, D. J. McCarthy and G. K. Smyth. "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data". In: Bioinformatics 26 (2010), pp. -1.

[14] H. Wickham and W. Chang. devtools: Tools to Make Developing R Packages Easier. R package version 1.11.1. 2016. URL: https://CRAN.R-project.org/package=devtools.


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

```
## Time difference of 1.867766 mins
```

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

```
## 2016-06-13 16:41:36
```

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

```
## Session info -------------------------------------------------------------------------------------
##  setting  value
##  version  R version 3.3.0 RC (2016-05-01 r70572)
##  system   x86_64, darwin13.4.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/New_York
##  date     2016-06-13
## Packages -----------------------------------------------------------------------------------------
##  package             * version  date       source
##  bibtex                0.4.0    2014-12-31 CRAN (R 3.3.0)
##  Biobase             * 2.33.0   2016-05-05 Bioconductor
##  BiocGenerics        * 0.19.1   2016-06-11 Bioconductor
##  BiocStyle           * 2.1.6    2016-06-11 Bioconductor
##  bitops                1.0-6    2013-08-17 CRAN (R 3.3.0)
##  colorout            * 1.1-2    2016-05-05 Github (jalvesaq/colorout@6538970)
##  colorspace            1.2-6    2015-03-11 CRAN (R 3.3.0)
##  devtools              1.11.1   2016-04-21 CRAN (R 3.3.0)
##  digest                0.6.9    2016-01-08 CRAN (R 3.3.0)
##  edgeR               * 3.15.0   2016-05-27 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        * 1.9.1    2016-05-13 Bioconductor
##  GenomicRanges       * 1.25.4   2016-06-10 Bioconductor
##  ggplot2               2.1.0    2016-03-01 CRAN (R 3.3.0)
##  gtable                0.2.0    2016-02-26 CRAN (R 3.3.0)
##  htmltools             0.3.5    2016-03-21 CRAN (R 3.3.0)
##  httr                  1.1.0    2016-01-28 CRAN (R 3.3.0)
##  IHW                 * 1.1.3    2016-06-13 Bioconductor
##  IRanges             * 2.7.6    2016-06-10 Bioconductor
##  knitcitations       * 1.0.7    2015-10-28 CRAN (R 3.3.0)
##  knitr                 1.13     2016-05-09 CRAN (R 3.3.0)
##  limma               * 3.29.7   2016-06-13 Bioconductor
##  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)
##  matrixStats         * 0.50.2   2016-04-24 CRAN (R 3.3.0)
##  memoise               1.0.0    2016-01-29 CRAN (R 3.3.0)
##  munsell               0.4.3    2016-02-13 CRAN (R 3.3.0)
##  plyr                  1.8.3    2015-06-12 CRAN (R 3.3.0)
##  qvalue              * 2.5.2    2016-05-20 Bioconductor
##  R6                    2.1.2    2016-01-26 CRAN (R 3.3.0)
##  Rcpp                  0.12.5   2016-05-14 CRAN (R 3.3.0)
##  RCurl                 1.95-4.8 2016-03-01 CRAN (R 3.3.0)
##  recount             * 0.99.10  2016-06-12 Github (leekgroup/recount@7a7ea73)
##  RefManageR            0.10.13  2016-04-04 CRAN (R 3.3.0)
##  reshape2              1.4.1    2014-12-06 CRAN (R 3.3.0)
##  RJSONIO               1.3-0    2014-07-28 CRAN (R 3.3.0)
##  rmarkdown           * 0.9.6    2016-05-01 CRAN (R 3.3.0)
##  RSkittleBrewer      * 1.1      2016-06-13 Github (alyssafrazee/RSkittleBrewer@230d1d0)
```

```
##  rstudioapi             0.5      2016-01-24 CRAN (R 3.3.0)
##  S4Vectors            * 0.11.4   2016-06-11 Bioconductor
##  scales                 0.4.0    2016-02-26 CRAN (R 3.3.0)
##  slam                   0.1-34   2016-05-04 CRAN (R 3.3.0)
##  stringi                1.0-1    2015-10-22 CRAN (R 3.3.0)
##  stringr                1.0.0    2015-04-30 CRAN (R 3.3.0)
##  SummarizedExperiment * 1.3.4    2016-06-10 Bioconductor
##  withr                  1.0.1    2016-02-04 CRAN (R 3.3.0)
##  XML                    3.98-1.4 2016-03-01 CRAN (R 3.3.0)
##  XVector                0.13.0   2016-05-05 Bioconductor
##  yaml                   2.1.13   2014-06-12 CRAN (R 3.3.0)
##  zlibbioc               1.19.0   2016-05-05 Bioconductor
```