

# recount (DER analyses)

*Kai Kammers and Shannon Ellis*

*July 22, 2016*

## Contents

<b>1 Download study data</b>	<b>2</b>
<b>2 Obtain count data</b>	<b>2</b>
<b>3 DE analysis</b>	<b>4</b>
<b>4 Annotate regions with genomic information</b>	<b>7</b>
<b>5 Gene level analysis</b>	<b>10</b>
<b>6 QC data</b>	<b>10</b>
<b>7 DE analysis : gene-level</b>	<b>11</b>
<b>8 Compare DER findings to gene-level analysis</b>	<b>14</b>
<b>9 Compare DERs to known genomic annotation</b>	<b>15</b>
<b>10 Compare DERs to known genomic annotation</b>	<b>18</b>
<b>11 Reproducibility</b>	<b>18</b>

Here is an example of how to download and analyze a `RangedSummarizedExperiment` object. While differential expression analysis of RNA-Seq data is often done at the gene and/or exon level, there is utility in looking for differential expression using an annotation-independent approach. Here, without the aid of any annotation, expressed regions are summarized, filtered, and differential expression analyses are carried out. These results are compared to those from annotation-dependent analyses carried out on the same data to assess whether additional information can be garnered from this approach. Data herein come from triple negative and HER2-positive breast cancer samples from SRA study id SRP032798.

We first load the required packages.

```
## load libraries
library('recount')
library('SummarizedExperiment')
library('limma')
library('edgeR')
library('qvalue')
library('topGO')
library('matrixStats')
library('RSkittleBrewer')
library('derfinder')
library('derfinderPlot')
library('BiocParallel')
library('GenomicRanges')
library('bumphunter')
library('downloader')
library('GenomicFeatures')
```

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

## 1 Download study data

Expressed region data are downloaded for each chromosome and summarized into a count matrix (`covMat`) where each region is a row and each sample a column.

```
chrs <- paste0('chr', c(1:22, 'X', 'Y'))
bp <- SerialParam() ## Change if you have access to more cores

## Download region level data
if(!file.exists('regions_SRP032789.Rdata')) {
  regions_list <- bplapply(chrs, function(chr) {
    regs <- expressed_regions('SRP032789', chr, cutoff = 5L,
      maxClusterGap = 3000L, verbose = FALSE)
    return(regs)
  }, BPPARAM = bp)
  names(regions_list) <- chrs
  regions <- unlist(GRangesList(regions_list))

  ## Save the regions
  save(regions, regions_list, file = 'regions_SRP032789.Rdata')
} else {
  load('regions_SRP032789.Rdata')
}

## Compute coverage matrix for study SRP032789
if(!file.exists('covMat_SRP032789.Rdata')) {
  covMat <- bplapply(chrs, function(chr) {
    coverageMatrix <- coverage_matrix('SRP032789', chr,
      regions_list[[chr]], verboseLoad = FALSE)
    return(coverageMatrix)
  }, BPPARAM = bp)
  covMat <- do.call(rbind, covMat)

  ## Round the coverage matrix to integers
  covMat <- round(covMat, 0)
  save(covMat, file = 'covMat_SRP032789.Rdata')
} else {
  load('covMat_SRP032789.Rdata')
}
```

## 2 Obtain count data

After summarizing data across expressed regions, data are filtered to only include samples of interest (TNBC and HER2+ tumor samples, (`covMat_filt`) and regions with greater than 5 mean read counts across samples (`counts`).

```

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

## Check ordering of samples
pheno <- pheno[pheno$Run_s %in% colnames(covMat), ]
identical(pheno$Run_s, colnames(covMat))

## [1] FALSE

## Obtain correct order for pheno data
pheno <- pheno[match(colnames(covMat), pheno$Run_s), ]
identical(pheno$Run_s, colnames(covMat))

## [1] TRUE
head(cbind(pheno$Run_s, colnames(covMat)))

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

## Find tumor type information
group <- pheno$tumor_type_s
table(group)

## 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 type
covMat_filt <- covMat[, group %in% c('HER2 Positive Breast Tumor', 'TNBC Breast Tumor')]
group <- group[group %in% c('HER2 Positive Breast Tumor', 'TNBC Breast Tumor')]
dim(covMat_filt)

## [1] 328481      11
rownames(covMat_filt) <- rownames(regions)

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

## [1] 291247      11
## Obtain chromosome and position information for regions included after filtering
regions_counts <- regions[filter, ]

```

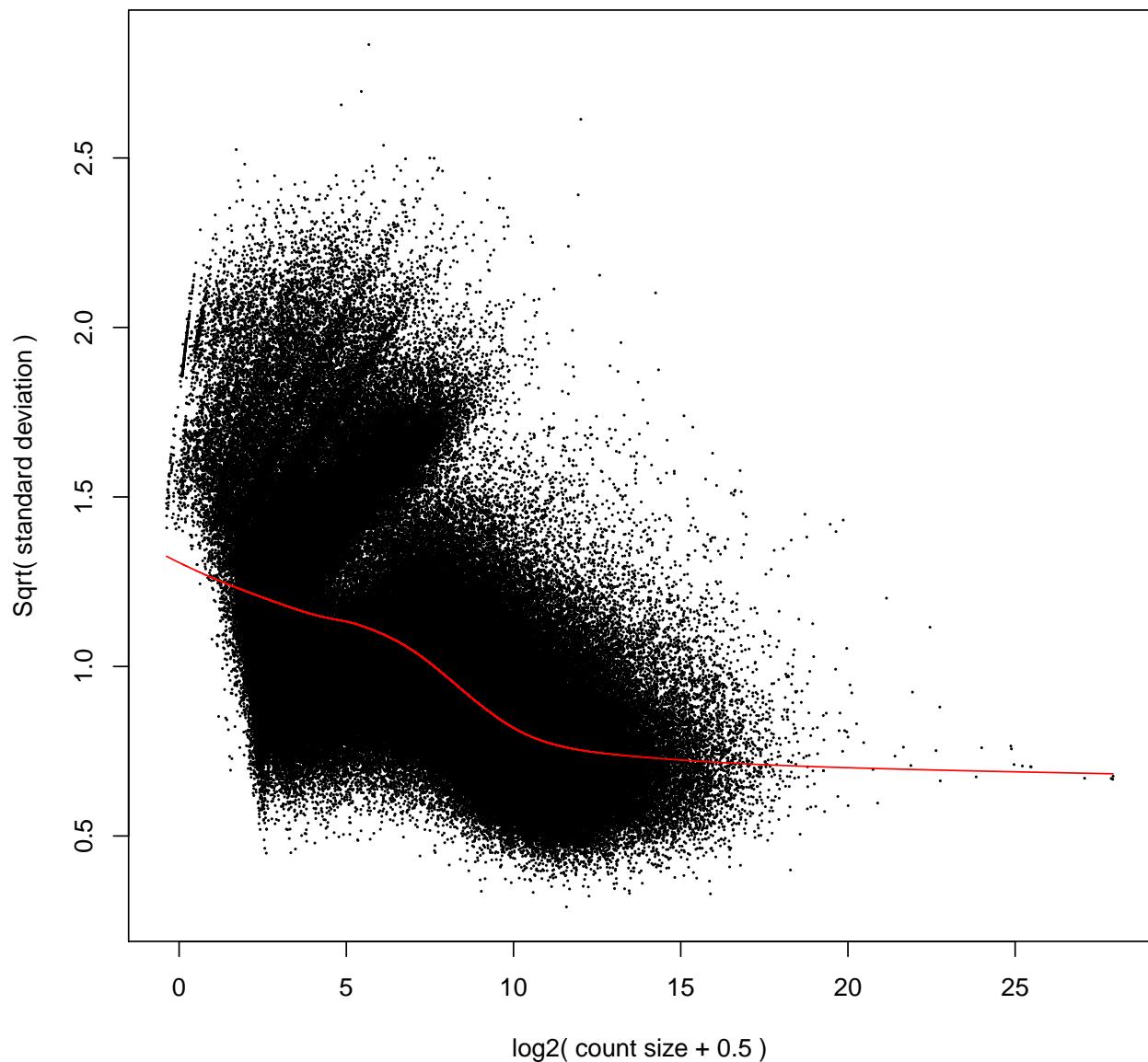
### 3 DE analysis

Differential expression analysis is carried out at the expressed-region level using `voom` and `limma`. Results for each region are plotted using a volcano plot to compare the effect size of the differential expression [ as measured by the  $\log_2(fold - change)$  in expression ] and its significance [  $-\log_{10}(p - value)$  ].

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

```
##      (Intercept) groupTNBC Breast Tumor
## 1              1          1     1
## 2              1          1     1
## 3              1          1     1
## 4              1          1     1
## 5              1          1     1
## 6              1          0     0
## 7              1          0     0
## 8              1          0     0
## 9              1          0     0
## 10             1          0     0
## 11             1          1     1
## attr(),"assign")
## [1] 0 1
## attr(),"contrasts")
## attr(),"contrasts")$group
## [1] "contr.treatment"
dge <- DGEList(counts = counts)
dge <- calcNormFactors(dge)
v <- voom(dge, design, plot = TRUE)
```

### voom: Mean–variance trend



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

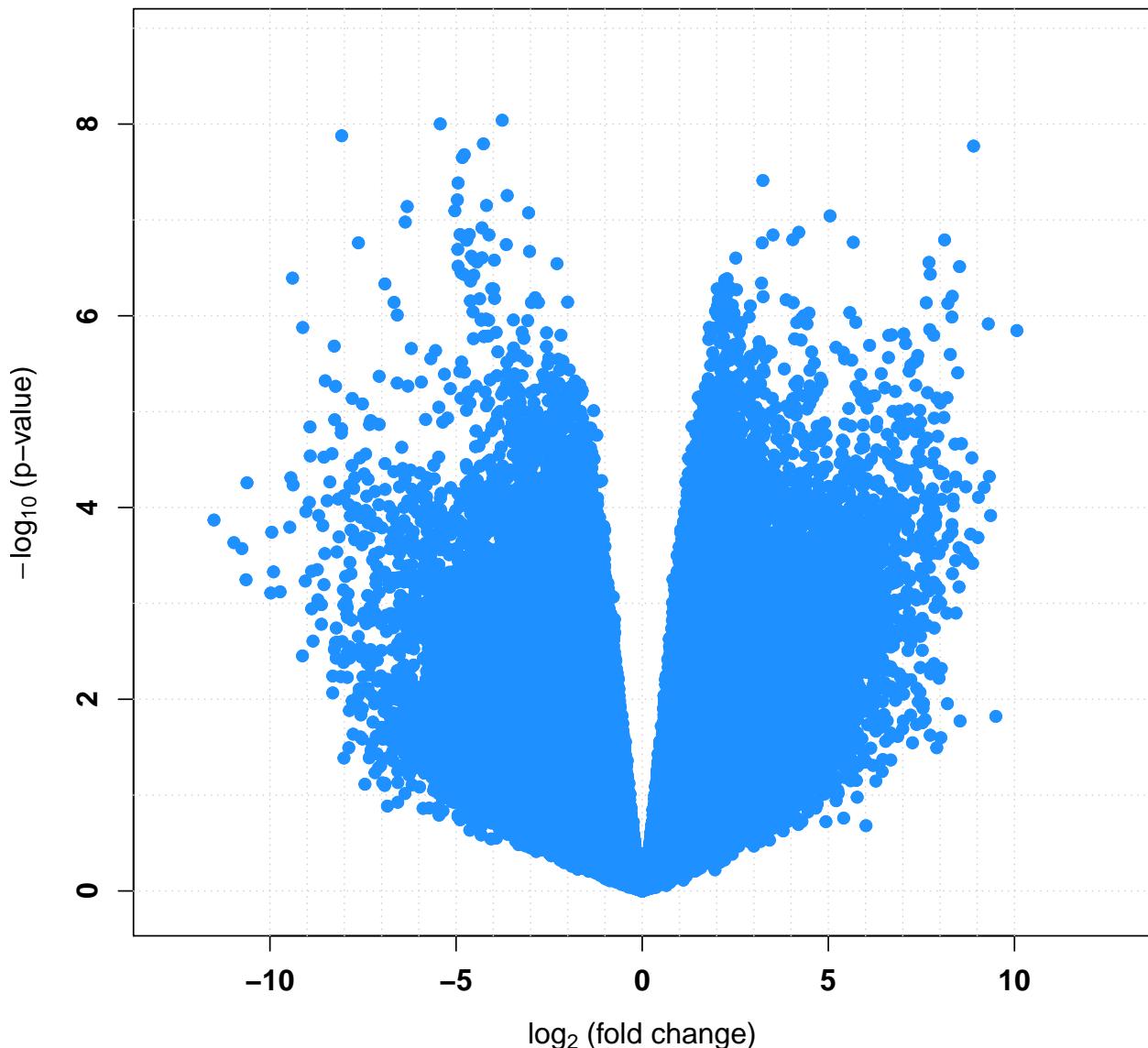
## [1] 35809
## Add differential expression information to GRanges object containing information
## about where the expressed regions are located in the genome (regions_counts)
regions_counts$log2FC <- res.regions$log2FC
regions_counts$p.mod <- res.regions$p.mod
regions_counts$q.mod <- res.regions$q.mod
```

```

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

```

**Volcano plot: TNBC vs. HER2+ in SRP032789 (er level)**



## 4 Annotate regions with genomic information

After detecting regions differentially expressed between TNBC and HER2+ samples, we can annotate where in the genome these regions are found and compare them back to the results from the gene- and exon-level differential expression analyses to determine what new information is gained by utilziing expressed region level data.

```
## Obtain significant (q<0.05) regions
regions_counts_sig <- regions_counts[regions_counts$q.mod < 0.05, ]

## Identify the top regions by highest total coverage
# top <- regions_counts_sig[order(regions_counts_sig$area, decreasing = TRUE)[1:100], ]

## Annotate significant regions
txdb <- makeTxDbFromGFF('ftp://ftp.sanger.ac.uk/pub/gencode/Gencode_human/release_25/gencode.v25.annotation.gff3', organism = 'Homo sapiens')

## Import genomic features from the file as a GRanges object ...
## OK

## Prepare the 'metadata' data frame ...
## OK

## Make the TxDb object ...

## Warning: RSQLite:::dbGetPreparedStatement() is deprecated, please switch to
## DBI:::dbGetQuery(params = bind.data).

## Warning: Named parameters not used in query: internal_chrom_id, chrom,
## length, is_circular

## Warning: Named parameters not used in query: internal_id, name, type,
## chrom, strand, start, end

## Warning: Named parameters not used in query: internal_id, name, chrom,
## strand, start, end

## Warning: Named parameters not used in query: internal_id, name, chrom,
## strand, start, end

## Warning: Named parameters not used in query: internal_tx_id, exon_rank,
## internal_exon_id, internal_cds_id

## Warning: Named parameters not used in query: gene_id, internal_tx_id
## OK

genes <- annotateTranscripts(txdb)

## No annotationPackage supplied. Trying org.Hs.eg.db.

## Loading required package: org.Hs.eg.db

##
## Getting TSS and TSE.
## Getting CSS and CSE.
## Getting exons.
## Annotating genes.
```

```

## Match detected regions with known gene annotation
## Warning: This can take some time
## (dependent upon how many regions are included for annotation)
# annotation_der <- matchGenes(regions_counts_sig, genes)

## Merge DER information with annotation information
# regions_annotated <- cbind(regions_counts_sig, annotation_der)

```

There are 35809 statistically significant regions detected ( $q < 0.05$ ). We can use a venn diagram to visually represent where they are found in the genome.

```

## Get Annotation
if(!file.exists('genomicState.Hsapiens.BioMart.ENSEMBLMARTENSEMBL.GRCh38.p5.Rdata')) {
  ## Genomic state created by https://github.com/nellore/runs/blob/master/gtex/DER_analysis/coverageMatrix
  download('https://github.com/nellore/runs/blob/master/gtex/DER_analysis/coverageMatrix/genomicState.Rdata')

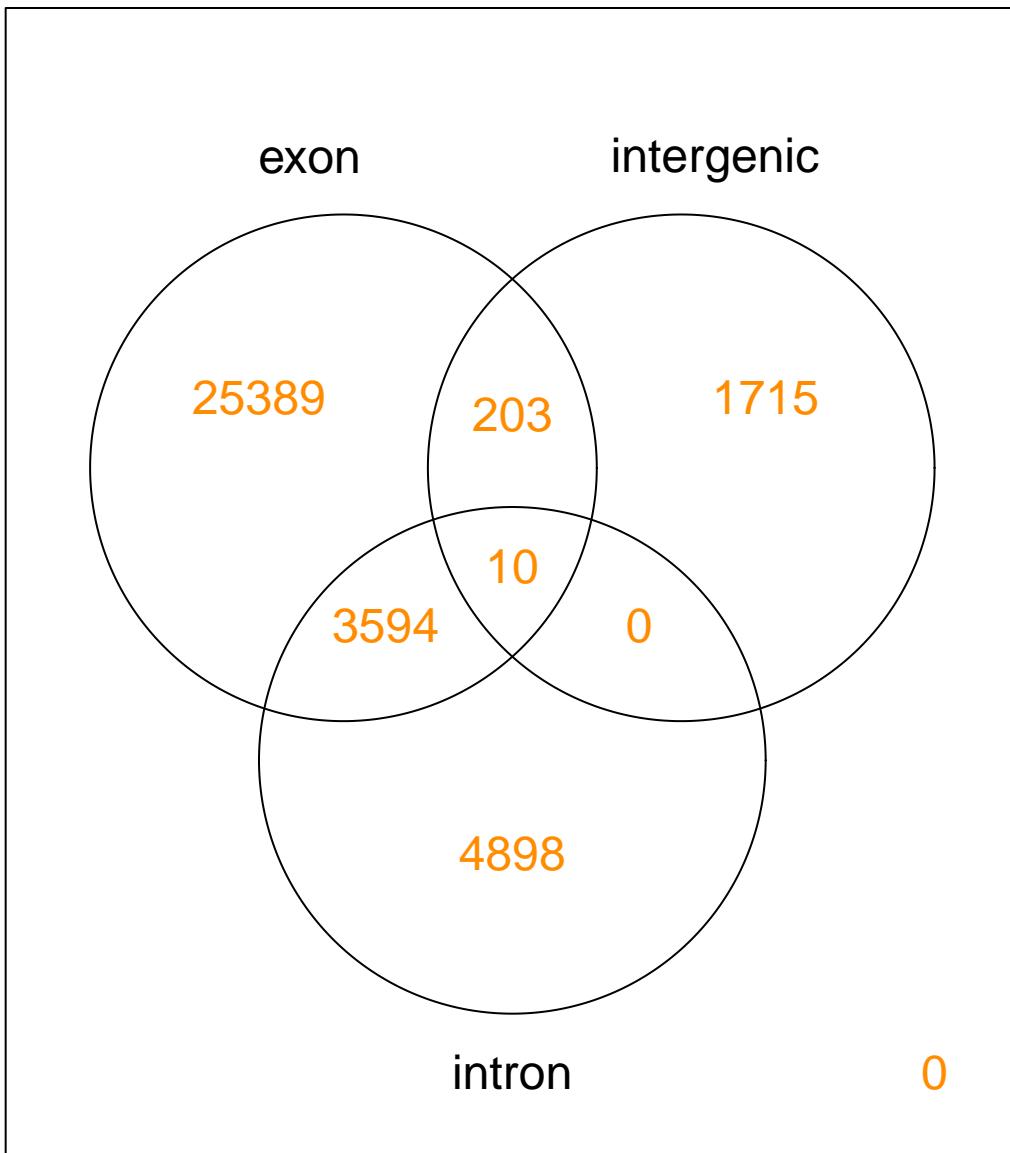
  load('genomicState.Hsapiens.BioMart.ENSEMBLMARTENSEMBL.GRCh38.p5.Rdata')
  gs_raw <- genomicState.Hsapiens.BioMart.ENSEMBLMARTENSEMBL.GRCh38.p5$fullGenome
  gs <- renameSeqlevels(gs_raw, paste0('chr', seqlevels(gs_raw)))
  gs_exons <- gs[gs$theRegion == "exon"]

  ## Make venn diagram
  ## Venn Diagram for significant DERs (q<0.05)
  annoRegs_sigDER <- annotateRegions(regions_counts_sig, gs, minoverlap = 1)

  ## 2017-02-07 14:13:36 annotateRegions: counting
  ## 2017-02-07 14:13:37 annotateRegions: annotating
  vennRegions(annoRegs_sigDER, main = 'Significant DERs (q<0.05)', counts.col = trop[1])
}

```

## Significant DERs ( $q < 0.05$ )



```
## exon intergenic intron Counts
## 1 0 0 0 0
## 2 0 0 1 4898
## 3 0 1 0 1715
## 4 0 1 1 0
## 5 1 0 0 25389
## 6 1 0 1 3594
## 7 1 1 0 203
## 8 1 1 1 10
## attr(),"class")
## [1] "VennCounts"
```

## 5 Gene level analysis

In order to compare these differentially expressed regions with gene-level results, we must carry out differential expression in these same data at the gene level.

We first download the project of interest (SRP032798), obtaining expression data for the study of interest.

```
## Find the project of interest (SRP032798), e.g. with parts of the abstract
project_info <- 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_info$project)
}

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

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

## 6 QC data

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

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

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

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

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

##      [,1]      [,2]
```

```

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

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

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

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

## class: RangedSummarizedExperiment
## dim: 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
counts <- assays(rse)$counts

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

## [1] 26742     11
counts_genes <- counts

```

## 7 DE analysis : gene-level

Using limma and voom, differentially expressed genes are detected at the gene level.

```

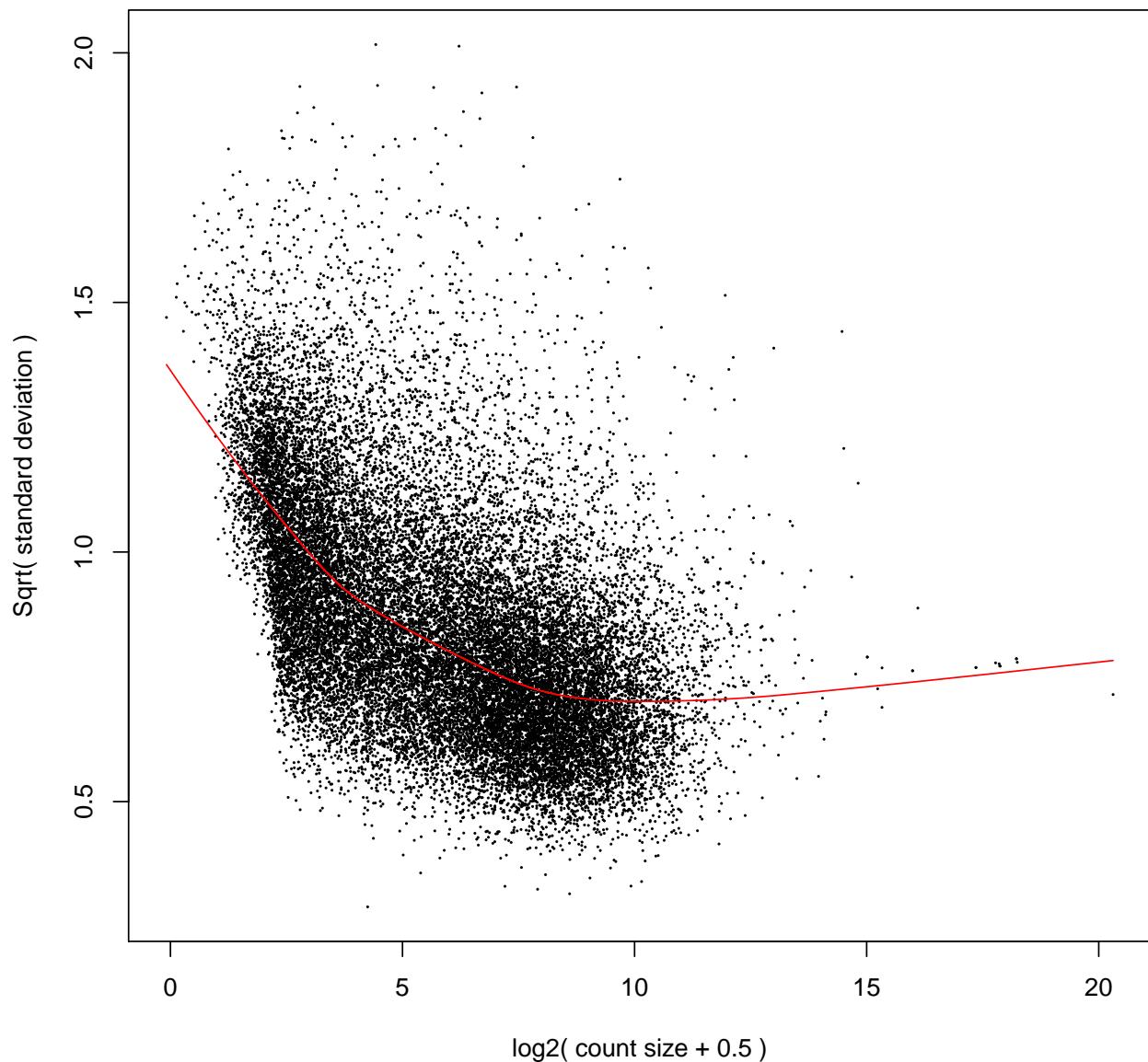
## Perform differential expression analysis with limma-voom
design <- model.matrix(~ rse$group)
design

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

```

```
## 4          1          1
## 5          1          1
## 6          1          0
## 7          1          0
## 8          1          0
## 9          1          0
## 10         1          0
## 11         1          1
## attr(),"assign")
## [1] 0 1
## attr(),"contrasts")
## attr(),"contrasts")$`rse$group`
## [1] "contr.treatment"
dge <- DGEList(counts = counts)
dge <- calcNormFactors(dge)
v <- voom(dge, design, plot = TRUE)
```

### voom: Mean–variance trend



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

## [1] 2974
```

## 8 Compare DER findings to gene-level analysis

By avoiding using gene or exon-level annotation for quantification, regions usually excluded from analysis can be included. Here, the same data set is utilized for differential expression; however, we determine the number of significantly differentially expressed regions (DERs,  $q < 0.05$ ) that do not overlap any gene. These regions are places within the genome that would be completely missed by the common gene-level analyses.

```
## Determine DERs not detected by gene level analysis
## Get DEGs into GRanges object format
genes_GRanges <- SummarizedExperiment::rowRanges(rse_gene$filter, [])
genes_GRanges$log2FC <- res.genes$log2FC
genes_GRanges$p.mod <- res.genes$p.mod
genes_GRanges$q.mod <- res.genes$q.mod

## For each region, determine if region overlaps with a gene included in the gene-level analysis
overlap <- countOverlaps(regions_counts_sig, genes_GRanges, type="any")
overlap <- countOverlaps(regions_counts, genes_GRanges, type="any")

regions_counts$overlap <- overlap
regions_counts_ingroup <- regions_counts[overlap>0]
regions_counts_notingroup <- regions_counts[overlap==0]

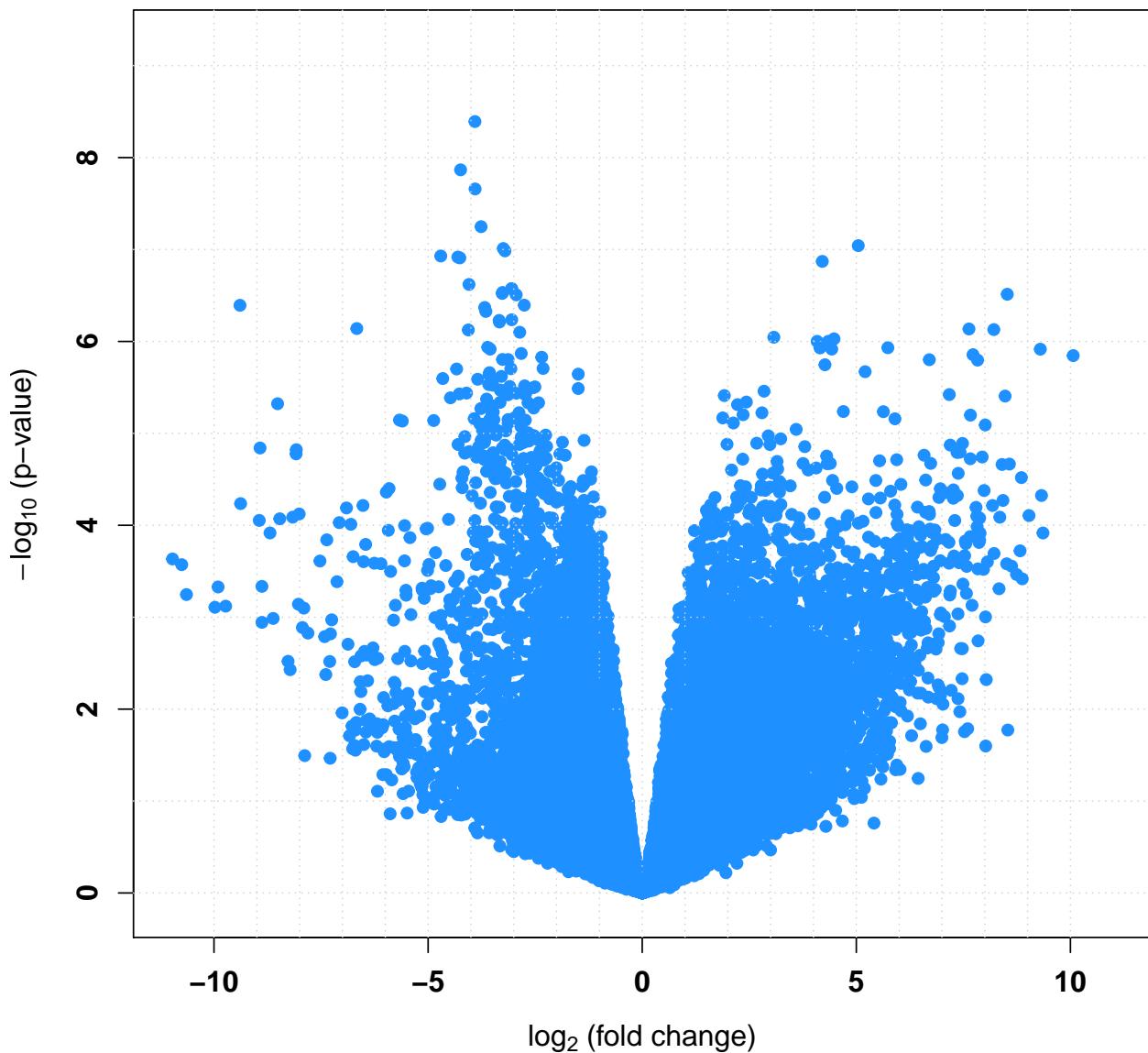
## For significant DERs ( $q < 0.05$ ), how many sites do not fall within a gene
## (and thus would not be detected by gene-level analyses

## number of DERs that do not overlap with genes in gene-level analysis
length(regions_counts_notingroup[regions_counts_notingroup$q.mod<0.05])

## [1] 1822

## Volcano plot of DERs (that do not overlap with genes)
par(font.lab = 2, cex.lab = 1.2, font.axis = 2, cex.axis = 1.2)
rx2 <- c(-11, 11)
ry2 <- c(-0.1, max(-log10(p.mod))) * 1.1
plot(log2FC, -log10(p.mod),
      pch = 19, xlim = rx2, ylim = ry2, col = trop[2],
      xlab = bquote(paste(log[2], ' (fold change)')),
      ylab = bquote(paste(-log[10], ' (p-value)')))
abline(v = seq(-10, 10, 1), col = 'lightgray', lty = 'dotted')
abline(h = seq(0, 23, 1), col = 'lightgray', lty = 'dotted')
points(regions_counts_notingroup$log2FC, -log10(regions_counts_notingroup$p.mod), pch = 19, col = trop[2])
title('Volcano plot: TNBC vs. HER2+ in SRP032789 (DERs not in genes)')
```

### Volcano plot: TNBC vs. HER2+ in SRP032789 (DERs not in genes)



There are 1822 DERs that do not overlap with genes included in the gene-level analysis.

These range in size from 1 bps to 5596 bps in length, with a median length of 29 bps.

## 9 Compare DERs to known genomic annotation

Beyond an interest in those regions not detected in the gene-level analysis, we are also interested in those regions that do not overlap any annotated gene. First, we quantify where DERs are found within the genome that do not overlap any known exons. Then, we highlight single locus coverage plots for a few of the longest DERs.

```
## For each region, determine if region overlaps with a gene included
overlapANY <- countOverlaps(regions_counts,gs_exons,type="any")
```

```

regions_counts$overlapANY <- overlapANY
regions_counts_ingroupANY <- regions_counts[regions_counts$overlapANY>0]
regions_counts_notinANYgene <- regions_counts[regions_counts$overlapANY==0]

## Number of DERs that do not overlap with any genes
length(regions_counts_notinANYgene[regions_counts_notinANYgene$q.mod<0.05])

## [1] 6613

## Get top 1% longest regions outside of genes to visualize
regions = subset(regions_counts_notinANYgene, width(regions_counts_notinANYgene) > quantile(width(regions_counts_notinANYgene), 0.99))
#order so most significant are plotted first
regions <- regions[order(regions$p.mod), ]

## make region plots
tIndexes = split(seq_len(length(rse$group)), rse$group)

## Get required information for the plots
## Annotate regions
annoRegs <- annotateRegions(regions, gs, minoverlap = 1)

## 2017-02-07 14:13:54 annotateRegions: counting
## 2017-02-07 14:13:55 annotateRegions: annotating

## Load full coverage
SRP032789_urls <- subset(recount_url, project == 'SRP032789' & file_name != 'mean_SRP032789.bw')
files <- SRP032789_urls$url[grep1('bw$', SRP032789_urls$url)]
names(files) <- gsub('.bw', '', SRP032789_urls$file_name[grep1('bw$', SRP032789_urls$url)]) 

## Only include TNBC and HER2+ samples
files <- subset(files, names(files) %in% colnames(covMat_filt))

## Obtain fullCoverage matrix for all chromosomes
fullCov <- fullCoverage(files = files, chrs = paste0('chr', c(1:22, 'X', 'Y')), 
    which = regions, verbose = FALSE)

## Find nearest annotation with bumphunter::matchGenes()
nearestAnnotation <- matchGenes(x = regions, subject = genes)

## .....

## Get the region coverage
geneRegionCov <- getRegionCoverage(fullCov=fullCov, regions=regions,
    targetSize = 4e+07, totalMapped=colData(rse)$reads_aligned, verbose = FALSE)
geneRegionCovMeans = lapply(geneRegionCov, function(x) {
    sapply(tIndexes, function(ii) rowMeans(x[,ii]))
})

## Plot region coverage for the longest DERs
pdf('DERs_notingene.pdf', h = 5, w = 7)
plotRegionCoverage(regions=regions,
    regionCoverage=geneRegionCovMeans,
    groupInfo= factor(unique(rse$group)), colors = trop,
    nearestAnnotation=nearestAnnotation,
    annotatedRegions=annoRegs,

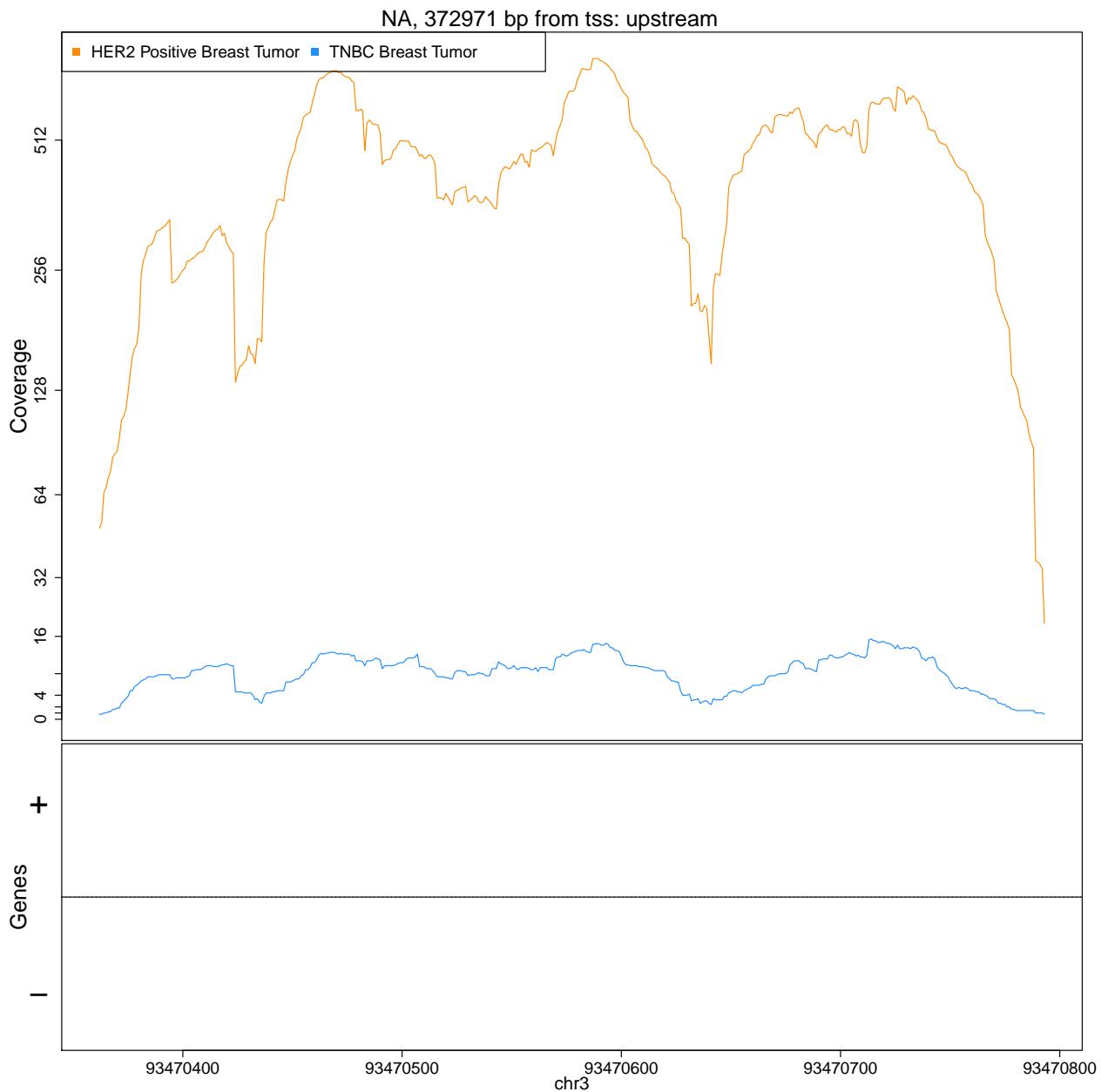
```

```

ask=FALSE, whichRegions=1:10, verbose=FALSE,
txdb = txdb)
dev.off()

## pdf
## 2
## Example of DER outside of any exonic region
plotRegionCoverage(regions=regions,
regionCoverage=geneRegionCovMeans,
groupInfo= factor(unique(rse$group)), colors = trop,
nearestAnnotation=nearestAnnotation,
annotatedRegions=annoRegs,
ask=FALSE, whichRegions=6, verbose=FALSE,
txdb = txdb)

```



There are 6613 DERs that do not overlap with any known genes.

These range in size from 1 bps to 15879 bps in length, with a median length of 22 bps.

## 10 Compare DERs to known genomic annotation

Finally, to compare the DER analysis to gene-level results, we must assign each DER to its nearest gene, creating the output file: “AnnotatedDERs.Rdata”.

```
## Assign regions to annotate with their genomic position
regions = regions_counts

## Obtain annotation information
ensemblAnno = annotateRegions(regions,gs)

## 2017-02-07 14:17:40 annotateRegions: counting
## 2017-02-07 14:17:41 annotateRegions: annotating
countTable = ensemblAnno$countTable

## Assign overlap between regions and genomic annotation
dA = distanceToNearest(regions_counts, genes)
regions$nearestSymbol = genes$Entrez[subjectHits(dA)]
regions$distToGene = mcols(dA)$distance
mcols(regions) = cbind(mcols(regions), countTable)

## Only include regions that are within genes to compare to gene-level findings
regions_exons <- regions[regions$exon==1]
regions_exons_distToGene <- regions_exons[regions_exons$distToGene==0]

## Extract p-value and nearest gene information to be used in comparison to gene-level analysis
annotatedDERs <- regions_exons_distToGene$p.mod
names(annotatedDERs) <- regions_exons_distToGene$nearestSymbol

## Save file for use in comparison analysis
save(annotatedDERs, file="AnnotatedDERs.Rdata")
```

## 11 Reproducibility

This analysis report was made possible thanks to:

- R (R Core Team, 2016)
- *BiocParallel* (Morgan, Obenchain, Lang, and Thompson, 2016)
- *BiocStyle* (Oleś, Morgan, and Huber, 2016)
- *derfinder* (Collado-Torres, Nellore, Frazee, Wilks, et al., 2016)
- *devtools* (Wickham and Chang, 2016)
- *downloader* (Chang, 2015)
- *edgeR* (Robinson, McCarthy, and Smyth, 2010)
- *GenomicFeatures*
- *GenomicRanges* (Lawrence, Huber, Pagès, Aboyoun, et al., 2013)
- *knitcitations* (Boettiger, 2015)
- *matrixStats* (Bengtsson, 2016)

- *recount* (Collado-Torres, Nellore, Kammers, Ellis, et al., 2016)
- *rmarkdown* (Allaire, Cheng, Xie, McPherson, et al., 2016)
- *RSkittleBrewer* (Frazee, 2017)
- *SummarizedExperiment* (Morgan, Obenchain, Hester, and Pagès, 2016)
- *topGO* (Alexa and Rahnenfuhrer, 2016)
- *limma* (Law, Chen, Shi, and Smyth, 2014)

## Bibliography file

- [1] A. Alexa and J. Rahnenfuhrer. topGO: Enrichment Analysis for Gene Ontology. R package version 2.26.0. 2016.
- [2] J. Allaire, J. Cheng, Y. Xie, J. McPherson, et al. rmarkdown: Dynamic Documents for R. R package version 1.3. 2016. URL: <https://CRAN.R-project.org/package=rmarkdown>.
- [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] W. Chang. downloader: Download Files over HTTP and HTTPS. R package version 0.4. 2015. URL: <https://CRAN.R-project.org/package=downloader>.
- [6] L. Collado-Torres, A. Nellore, A. C. Frazee, C. Wilks, et al. “Flexible expressed region analysis for RNA-seq with derfinder”. In: Nucl. Acids Res. (2016). DOI: 10.1093/nar/gkw852. URL: <http://nar.oxfordjournals.org/content/early/2016/09/29/nar.gkw852>.
- [7] L. Collado-Torres, A. Nellore, K. Kammers, S. E. Ellis, et al. “recount: A large-scale resource of analysis-ready RNA-seq expression data”. In: bioRxiv (2016). DOI: 10.1101/068478. URL: <http://biorkxiv.org/content/early/2016/08/08/068478>.
- [8] A. Frazee. RSkittleBrewer: Fun with R Colors. R package version 1.1. 2017. URL: <https://github.com/alyssafrazee/RSkittleBrewer>.
- [9] C. Law, Y. Chen, W. Shi and G. Smyth. “Voom: precision weights unlock linear model analysis tools for RNA-seq read counts”. In: Genome Biology 15 (2014), p. R29.
- [10] M. Lawrence, W. Huber, H. Pagès, P. Aboyoum, et al. “Software for Computing and Annotating Genomic Ranges”. In: PLoS Computational Biology 9 (8 2013). DOI: 10.1371/journal.pcbi.1003118. URL: <http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1003118>.
- [11] M. Morgan, V. Obenchain, J. Hester and H. Pagès. SummarizedExperiment: SummarizedExperiment container. R package version 1.4.0. 2016.
- [12] M. Morgan, V. Obenchain, M. Lang and R. Thompson. BiocParallel: Bioconductor facilities for parallel evaluation. R package version 1.8.1. 2016. URL: <https://github.com/Bioconductor/BiocParallel>.
- [13] A. Oleś, M. Morgan and W. Huber. BiocStyle: Standard styles for vignettes and other Bioconductor documents. R package version 2.2.1. 2016. URL: <https://github.com/Bioconductor/BiocStyle>.
- [14] 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/>.
- [15] 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.
- [16] 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 9.31154 mins
```

```

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

## 2017-02-07 14:17:48
## Reproducibility info
options(width = 120)
devtools::session_info()

## Session info -----
## setting value
## version R version 3.3.2 RC (2016-10-26 r71594)
## system x86_64, darwin13.4.0
## ui X11
## language (EN)
## collate en_US.UTF-8
## tz America/New_York
## date 2017-02-07

## Packages -----
## package * version date source
## acepack 1.4.1 2016-10-29 CRAN (R 3.3.0)
## AnnotationDbi * 1.36.2 2017-01-30 Bioconductor
## AnnotationHub 2.6.4 2016-11-24 Bioconductor
## assertthat 0.1 2013-12-06 cran (@0.1)
## backports 1.0.5 2017-01-18 CRAN (R 3.3.2)
## base64enc 0.1-3 2015-07-28 cran (@0.1-3)
## bibtex 0.4.0 2014-12-31 CRAN (R 3.3.0)
## Biobase * 2.34.0 2016-10-18 Bioconductor
## BiocGenerics * 0.20.0 2016-10-18 Bioconductor
## BiocInstaller 1.24.0 2016-10-18 Bioconductor
## BiocParallel * 1.8.1 2016-10-30 Bioconductor
## BiocStyle * 2.2.1 2016-11-24 Bioconductor
## biomaRt 2.30.0 2016-10-18 Bioconductor
## Biostrings 2.42.1 2016-12-01 Bioconductor
## biovizBase 1.22.0 2016-10-18 Bioconductor
## bitops 1.0-6 2013-08-17 cran (@1.0-6)
## BSgenome 1.42.0 2016-10-18 Bioconductor
## bumphunter * 1.14.0 2016-10-18 Bioconductor
## checkmate 1.8.2 2016-11-02 CRAN (R 3.3.1)
## cluster 2.0.5 2016-10-08 CRAN (R 3.3.2)
## codetools 0.2-15 2016-10-05 CRAN (R 3.3.2)
## colorout * 1.1-2 2016-10-19 Github (jalvesaq/colorout@6d84420)
## colorspace 1.3-2 2016-12-14 CRAN (R 3.3.2)
## data.table 1.10.4 2017-02-01 CRAN (R 3.3.2)
## DBI 0.5-1 2016-09-10 cran (@0.5-1)
## derfinder * 1.8.0 2016-10-18 Bioconductor
## derfinderHelper 1.8.0 2016-10-18 Bioconductor
## derfinderPlot * 1.8.0 2016-10-18 Bioconductor
## devtools 1.12.0 2016-06-24 CRAN (R 3.3.0)
## dichromat 2.0-0 2013-01-24 cran (@2.0-0)
## digest 0.6.12 2017-01-27 CRAN (R 3.3.2)
## doRNG 1.6 2014-03-07 CRAN (R 3.3.0)
## downloader * 0.4 2015-07-09 CRAN (R 3.3.0)

```

```

## edgeR * 3.16.5 2016-12-15 Bioconductor
## ensemblDb 1.6.2 2016-11-17 Bioconductor
## evaluate 0.10 2016-10-11 cran (@0.10)
## foreach * 1.4.3 2015-10-13 CRAN (R 3.3.0)
## foreign 0.8-67 2016-09-13 CRAN (R 3.3.2)
## Formula 1.2-1 2015-04-07 CRAN (R 3.3.0)
## GenomeInfoDb * 1.10.2 2016-12-29 Bioconductor
## GenomicAlignments 1.10.0 2016-10-18 Bioconductor
## GenomicFeatures * 1.26.2 2016-12-17 Bioconductor
## GenomicFiles 1.10.3 2016-10-21 Bioconductor
## GenomicRanges * 1.26.2 2017-01-02 Bioconductor
## GEOquery 2.40.0 2016-10-18 Bioconductor
## GGally 1.3.0 2016-11-13 CRAN (R 3.3.2)
## ggbio 1.22.3 2016-12-23 Bioconductor
## ggplot2 2.2.1 2016-12-30 CRAN (R 3.3.2)
## GO.db * 3.4.0 2016-10-19 Bioconductor
## graph * 1.52.0 2016-10-18 Bioconductor
## gridExtra 2.2.1 2016-02-29 CRAN (R 3.3.0)
## gtable 0.2.0 2016-02-26 CRAN (R 3.3.0)
## Hmisc 4.0-2 2016-12-31 CRAN (R 3.3.2)
## htmlTable 1.9 2017-01-26 CRAN (R 3.3.2)
## htmltools 0.3.5 2016-03-21 cran (@0.3.5)
## htmlwidgets 0.8 2016-11-09 CRAN (R 3.3.2)
## httpuv 1.3.3 2015-08-04 CRAN (R 3.3.0)
## httr 1.2.1 2016-07-03 CRAN (R 3.3.0)
## interactiveDisplayBase 1.12.0 2016-10-18 Bioconductor
## IRanges * 2.8.1 2016-11-08 Bioconductor
## iterators * 1.0.8 2015-10-13 CRAN (R 3.3.0)
## jsonlite 1.2 2016-12-31 CRAN (R 3.3.2)
## knitr * 1.0.7 2015-10-28 CRAN (R 3.3.0)
## knitr 1.15.1 2016-11-22 CRAN (R 3.3.2)
## lattice 0.20-34 2016-09-06 CRAN (R 3.3.2)
## latticeExtra 0.6-28 2016-02-09 CRAN (R 3.3.0)
## lazyeval 0.2.0 2016-06-12 cran (@0.2.0)
## limma * 3.30.10 2017-02-03 Bioconductor
## locfit * 1.5-9.1 2013-04-20 CRAN (R 3.3.0)
## lubridate 1.6.0 2016-09-13 CRAN (R 3.3.0)
## magrittr 1.5 2014-11-22 cran (@1.5)
## Matrix 1.2-8 2017-01-20 CRAN (R 3.3.2)
## matrixStats * 0.51.0 2016-10-09 CRAN (R 3.3.0)
## memoise 1.0.0 2016-01-29 CRAN (R 3.3.0)
## mime 0.5 2016-07-07 CRAN (R 3.3.0)
## munsell 0.4.3 2016-02-13 cran (@0.4.3)
## nnet 7.3-12 2016-02-02 CRAN (R 3.3.2)
## org.Hs.eg.db * 3.4.0 2016-10-19 Bioconductor
## OrganismDbi 1.16.0 2016-10-18 Bioconductor
## pkgmaker 0.22 2014-05-14 CRAN (R 3.3.0)
## plyr 1.8.4 2016-06-08 cran (@1.8.4)
## qvalue * 2.6.0 2016-10-18 Bioconductor
## R6 2.2.0 2016-10-05 CRAN (R 3.3.0)
## RBGL 1.50.0 2016-10-18 Bioconductor
## RColorBrewer 1.1-2 2014-12-07 cran (@1.1-2)
## Rcpp 0.12.9 2017-01-14 CRAN (R 3.3.2)
## RCurl 1.95-4.8 2016-03-01 cran (@1.95-4.)

```

```

##  recount          * 1.0.9    2017-02-07 Bioconductor
##  RefManageR       0.13.1   2016-11-13 CRAN (R 3.3.2)
##  registry         0.3      2015-07-08 CRAN (R 3.3.0)
##  rentrez          1.0.4    2016-10-26 CRAN (R 3.3.0)
##  reshape           0.8.6    2016-10-21 CRAN (R 3.3.0)
##  reshape2          1.4.2    2016-10-22 CRAN (R 3.3.0)
##  RJSONIO           1.3-0    2014-07-28 cran (@1.3-0)
##  rmarkdown          * 1.3     2016-12-21 CRAN (R 3.3.2)
##  rngtools          1.2.4    2014-03-06 CRAN (R 3.3.0)
##  rpart             4.1-10   2015-06-29 CRAN (R 3.3.2)
##  rprojroot          1.2     2017-01-16 CRAN (R 3.3.2)
##  Rsamtools          1.26.1   2016-10-22 Bioconductor
##  RSkittleBrewer    * 1.1     2016-10-19 Github (alyssaafrazee/RSkittleBrewer@0088112)
##  RSQLite            1.1-2    2017-01-08 CRAN (R 3.3.2)
##  rtracklayer        1.34.1   2016-11-02 Bioconductor
##  S4Vectors          * 0.12.1  2016-12-01 Bioconductor
##  scales             0.4.1    2016-11-09 CRAN (R 3.3.2)
##  shiny              1.0.0    2017-01-12 cran (@1.0.0)
##  SparseM            * 1.74    2016-11-10 CRAN (R 3.3.2)
##  stringi             1.1.2    2016-10-01 cran (@1.1.2)
##  stringr             1.1.0    2016-08-19 cran (@1.1.0)
##  SummarizedExperiment * 1.4.0    2016-10-18 Bioconductor
##  survival           2.40-1   2016-10-30 CRAN (R 3.3.0)
##  tibble              1.2     2016-08-26 cran (@1.2)
##  topGO              * 2.26.0   2016-10-18 Bioconductor
##  VariantAnnotation   1.20.2   2016-12-01 Bioconductor
##  withr               1.0.2    2016-06-20 CRAN (R 3.3.0)
##  XML                 3.98-1.5 2016-11-10 CRAN (R 3.3.2)
##  xtable              1.8-2    2016-02-05 CRAN (R 3.3.0)
##  XVector             0.14.0   2016-10-18 Bioconductor
##  yaml                2.1.14   2016-11-12 CRAN (R 3.3.2)
##  zlibbioc            1.20.0   2016-10-18 Bioconductor

```