metaRNASeq: Differential meta-analysis of RNA-seq data

Guillemette Marot, Florence Jaffrézic, Andrea Rau

Modified: January 23, 2015. Compiled: October 2, 2020

Abstract

This vignette illustrates the use of the *metaRNASeq* package to combine data from multiple RNA-seq experiments. Based both on simulated and real publicly available data, it also explains the way the p-value data provided in the package have been obtained.

Contents

1 Introduction

High-throughput sequencing (HTS) data, such as RNA-sequencing (RNA-seq) data, are increasingly used to conduct differential analyses, in which gene-by-gene statistical tests are performed in order to identify genes whose expression levels show systematic covariation with a particular condition, such as a treatment or phenotype of interest. Due to

their large cost, however, only few biological replicates are often considered in each experiment leading to a low detection power of differentially expressed genes. For this reason, analyzing data arising from several experiments studying the same question can be a useful way to increase detection power for the identification of differentially expressed genes.

The *metaRNASeq* package implements two p-value combination techniques (inverse normal and Fisher methods); see [4] for additional details. There are two fundamental assumptions behind the use of these p-value combination procedures: first, that p-values have been obtained the same way for each experiment (i.e., using the same model and test); and second, that they follow a uniform distribution under the null hypothesis. In this vignette, we illustrate these p-value combination techniques after obtaining p-values for differential expression in each individual experiment using the *DESeq2* Bioconductor package [1]. Count data are simulated using the `sim.function` provided in the *metaRNASeq* package; see section 2 for additional detail.

2 Simulation study

To begin, we load the necessary packages and simulation parameters:

```
> library(metaRNASeq)
> data(param)
> dim(param)

[1] 26408     3

> data(dispFuncs)
```

These simulation parameters include the following information:

- `param`: Matrix of dimension (26408 $\times$ 3) containing mean expression in each of two conditions (here, labeled "condition 1" and "condition 2") and a logical vector indicating the presence or absence of differential expression for each of 26,408 genes

- `dispFuncs`: List of length 2, where each list is a vector containing the two estimated coefficients (α_0 and α_1) for the gamma-family generalized linear model (GLM) fit by *DESeq* (version 1.8.3) describing the mean-dispersion relationship for each of the two real datasets considered in [4]. These regressions represent the typical relationship between mean expression values μ and dispersions α in each dataset, where the coefficients α_0 and α_1 are found to parameterize the fit as $\alpha = \alpha_0 + \alpha_1/\mu$.

These parameters were calculated on real data sets from two human melanoma cell lines [5], corresponding to two different studies performed for the same cell line comparison, with two biological replicates per cell line in the first and three per cell line in the

second. These data are presented in greater detail in [5] and [2], and are freely available in the Supplementary Materials of the latter.

Once parameters are loaded, we simulate data. We use the `set.seed` function to obtain reproducible results.

```
> set.seed(123)
> matsim <- sim.function(param = param, dispFuncs = dispFuncs)
> sim.conds <- colnames(matsim)
> rownames(matsim) <- paste("tag", 1:dim(matsim)[1],sep="")
> dim(matsim)

[1] 26408    16
```

The simulated matrix data contains 26,408 genes and 4 replicates per condition per study. It is possible to change the number of replicates in each study using either the `nrep` argument or the `classes` argument. Using `nrep` simulates the same number of replicates per condition per study. In order to simulate an unbalanced design, the `classes` argument may be used. For example, setting

```
classes = list(c(1,2,1,1,2,1,1,2),c(1,1,1,2,2,2,2))
```

leads to 5 and 3 replicates in each condition for the first study, and 3 and 4 replicates in each condition in the second.

3 Individual analyses of the two simulated data sets

Before performing a combination of p-values from each study, it is necessary to perform a differential analysis of the individual studies (using the same method). In the following example, we make use of the *DESeq2* package to obtain p-values for differential analyses of each study independently; however, we note that other differential analysis methods (e.g., *edgeR* or *baySeq*) could be used prior to the meta analysis.

3.1 Differential analysis of each individual study with DESeq2

Inputs to DEseq2 methods can be extracted with `extractfromsim` for each individual study whose name appears in the column names of `matsim`, see the following example for study1 and study2.

```
> colnames(matsim)

 [1] "study1cond1" "study1cond1" "study1cond1" "study1cond1"
 [5] "study1cond2" "study1cond2" "study1cond2" "study1cond2"
 [9] "study2cond1" "study2cond1" "study2cond1" "study2cond1"
[13] "study2cond2" "study2cond2" "study2cond2" "study2cond2"
```

```
> simstudy1 <- extractfromsim(matsim,"study1")
> head(simstudy1$study)
```

```
     rep1 rep2 rep3 rep4 rep5 rep6 rep7 rep8
tag1  338  401  428  565  476  545  407  367
tag2  919  849 1397 1541  917 1268 1596 1020
tag3  127  166  235  276  133  206  238  127
tag4  224  353  426  252  881  717  889  808
tag5    4    4    8    6    9    5   10    9
tag6  108   61   39   22  158   97   16  107
```

```
> simstudy1$pheno
```

```
      study condition
rep1 study1 untreated
rep2 study1 untreated
rep3 study1 untreated
rep4 study1 untreated
rep5 study1   treated
rep6 study1   treated
rep7 study1   treated
rep8 study1   treated
```

```
> simstudy2 <- extractfromsim(matsim,"study2")
```

Differential analyses for each study are then easily performed using the `DESeqDataSetFromMatrix` method.

```
>  if (requireNamespace("DESeq2", quietly = TRUE)) {
+     dds1 <- DESeq2::DESeqDataSetFromMatrix(countData = simstudy1$study,
+       colData = simstudy1$pheno,design = ~ condition)
+     res1 <- DESeq2::results(DESeq2::DESeq(dds1))
+     dds2 <- DESeq2::DESeqDataSetFromMatrix(countData = simstudy2$study,
+       colData = simstudy2$pheno,design = ~ condition)
+     res2 <- DESeq2::results(DESeq2::DESeq(dds2))
+   }
```

We recommand to store both p-value and Fold Change results in lists in order to perform meta-analysis and keep track of the potential conflicts (see section 5)

```
> rawpval <- list("pval1"=res1[["pvalue"]],"pval2"=res2[["pvalue"]])
> FC <- list("FC1"=res1[["log2FoldChange"]],"FC2"=res2[["log2FoldChange"]])
```

Differentially expressed genes in each individual study can also be marked in a matrix `DE`:

```
> adjpval <- list("adjpval1"=res1[["padj"]],"adjpval2"=res2[["padj"]])
> studies <- c("study1", "study2")
> DE <- mapply(adjpval, FUN=function(x) ifelse(x <= 0.05, 1, 0))
> colnames(DE)=paste("DE",studies,sep=".")
```

DE returns a matrix with 1 for genes identified as differentially expressed and 0 otherwise (one column per study)

Since the proposed p-value combination techniques rely on the assumption that p-values follow a uniform distribution under the null hypothesis, it is necesary to check that the histograms of raw-pvalues reflect that assumption:

```
> par(mfrow = c(1,2))
> hist(rawpval[[1]], breaks=100, col="grey", main="Study 1", xlab="Raw p-values")
> hist(rawpval[[2]], breaks=100, col="grey", main="Study 2", xlab="Raw p-values")
```

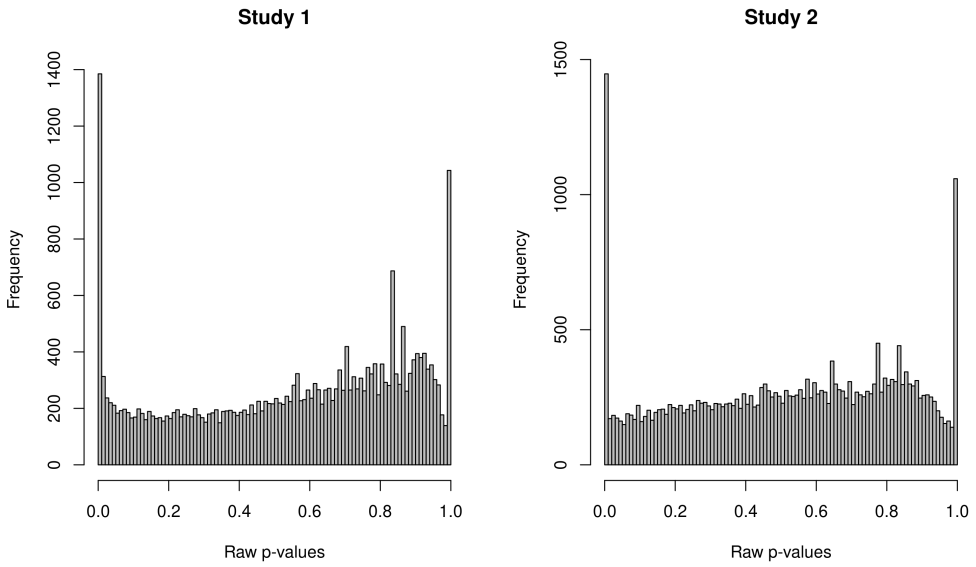

Figure 1: Histograms of raw p-values for each of the individual differential analyses performed using the *DESeq2* package.

The peak near 0 corresponds to differentially expressed genes, no other peak should appear. Sometimes another peak may appear due to genes with very low values of expression which often lead to an enrichment of p-values close to 1 as they take on discrete values. As such genes are unlikely to display evidence for differential expression, it is recommended to perform an independent filtering. The application of such a filter typically removes those genes contributing to a peak of p-values close to 1, leading to a distribution of p-values under the null hypothesis more closely following a uniform

distribution. As the proposed p-value combination techniques rely on this assumption, it is sometimes necessary to independently filter genes with very low read counts.

In this example the `results` function of DESeq2 performs an automatic independent filtering. If a row is filtered by independent filtering, then only the adjusted p-value will be set to NA, and the graphic of raw p-values does not change. In order to have a distribution of raw p-values under the null hypothesis following a uniform distribution, we must manually set the corresponding raw p-values to NA.

```
> filtered <- lapply(adjpval, FUN=function(pval) which(is.na(pval)))
> rawpval[[1]][filtered[[1]]]=NA
> rawpval[[2]][filtered[[2]]]=NA
```

To confirm that the raw p-values under the null hypothesis are roughly uniformly distributed, we may also inspect histograms of the raw p-values from each of the individual differential analyses (see Figure 2):

```
> par(mfrow = c(1,2))
> hist(rawpval[[1]], breaks=100, col="grey", main="Study 1",
+   xlab="Raw p-values")
> hist(rawpval[[2]], breaks=100, col="grey", main="Study 2",
+   xlab="Raw p-values")
```

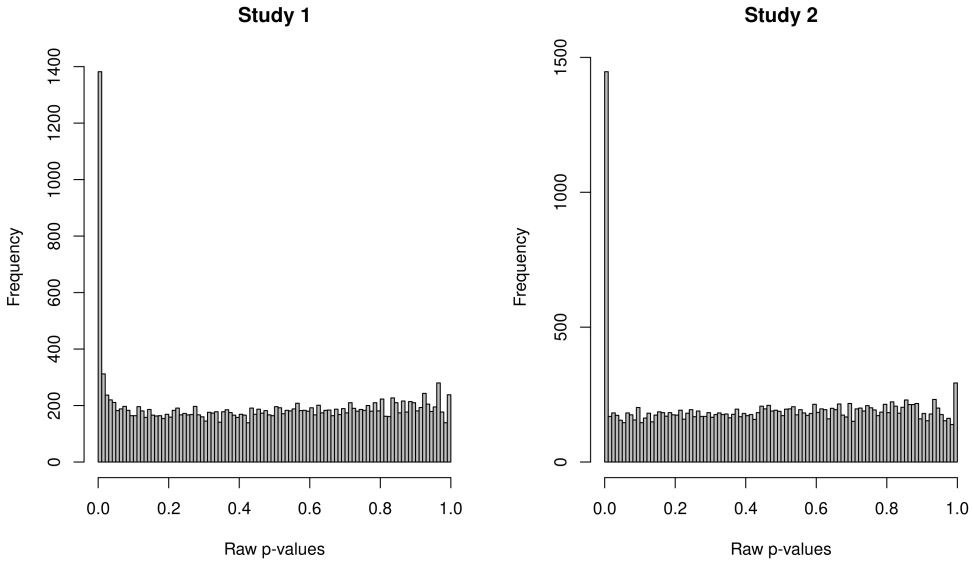

Figure 2: Histograms of raw p-values for each of the individual differential analyses performed using the independent filtering from *DESeq2* package.

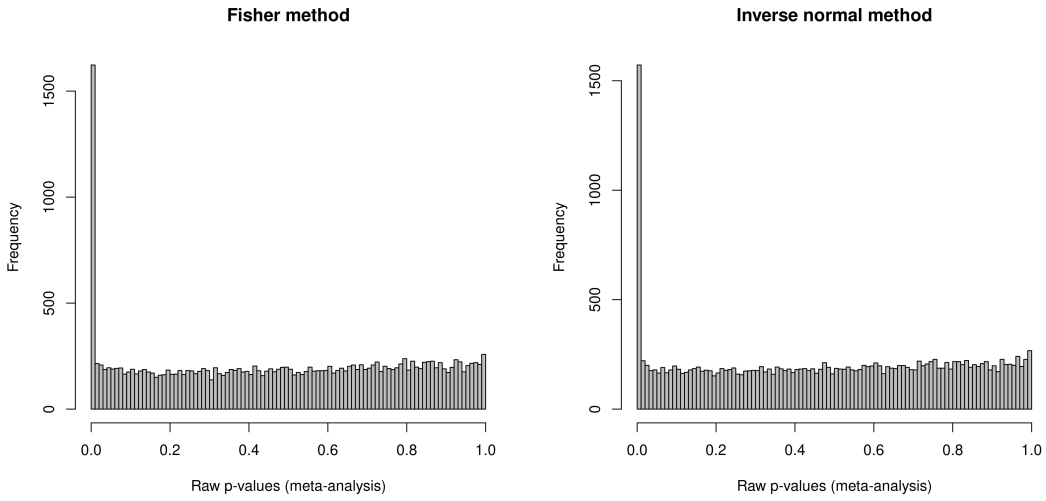

Figure 3: (Left) Histogram of raw p-values obtained after a meta-analysis of all studies, with p-value combination performed using the Fisher method. (Right) Histogram of raw p-values obtained after a meta-analysis of all studies, with p-value combination performed using the inverse normal method.

4 Use of p-value combination techniques

The code in this section may be used independently from the previous section if p-values from each study have been obtained using the same differential analysis test between the different studies. Vectors of p-values must have the same length; `rawpval` is a list (or data.frame) containing the vectors of raw p-values obtained from the individual differential analyses of each study.

The p-value combination using the Fisher method may be performed with the `fishercomb` function, and the subsequent p-values obtained from the meta-analysis may be examined (Figure 3, left):

```
> fishcomb <- fishercomb(rawpval, BHth = 0.05)
> hist(fishcomb$rawpval, breaks=100, col="grey", main="Fisher method",
+   xlab = "Raw p-values (meta-analysis)")
```

The use of the inverse normal combination technique requires the choice of a weight for each study. In this example, we choose `nrep=8`, since 8 replicates had been simulated in each study. As before, we may examine a histogram of the subsequent p-values obtained from the meta-analysis (Figure 3, right).

```
> invnormcomb <- invnorm(rawpval,nrep=c(8,8), BHth = 0.05)
> hist(invnormcomb$rawpval, breaks=100, col="grey",
```

```
+   main="Inverse normal method",
+   xlab = "Raw p-values (meta-analysis)")
```

Finally, we suggest summarizing the results of the individual differential analyses as well as the differential meta-analysis (using the Fisher and inverse normal methods) in a data.frame:

```
> DEresults <- data.frame(DE,
+   "DE.fishercomb"=ifelse(fishcomb$adjpval<=0.05,1,0),
+   "DE.invnorm"=ifelse(invnormcomb$adjpval<=0.05,1,0))
> head(DEresults)
```

```
  DE.study1 DE.study2 DE.fishercomb DE.invnorm
1         0         0             0          0
2         0         0             0          0
3         0         0             0          0
4         1         1             1          1
5         0        NA             0          0
6         0         0             0          0
```

5 Treatment of conflicts in differential expression

As pointed out in [4], it is not possible to directly avoid conflicts between over- and under- expressed genes in separate studies that appear in differential meta-analyses of RNA-seq data. We thus advise checking that individual studies identify differential expression in the same direction (i.e., if in one study, a gene is identified as differentially over-expressed in condition 1 as compared to condition 2, it should not be identified as under-expressed in condition 1 as compared to condition 2 in a second study). Genes displaying contradictory differential expression in separate studies should be removed from the list of genes identified as differentially expressed via meta-analysis.

We build a matrix `signsFC` gathering all signs of fold changes from individual studies.

```
> signsFC <- mapply(FC, FUN=function(x) sign(x))
> sumsigns <- apply(signsFC,1,sum)
> commonsgnFC <- ifelse(abs(sumsigns)==dim(signsFC)[2], sign(sumsigns),0)
```

The vector `commonsgnFC` will return a value of 1 if the gene has a positive $\log_2$ fold change in all studies, -1 if the gene has a negative $\log_2$ fold change in all studies, and 0 if contradictory $\log_2$ fold changes are observed across studies (i.e., positive in one and negative in the other). By examining the elements of `commonsgnFC`, it is thus possible to identify genes displaying contradictory differential expression among studies.

```
> unionDE <- unique(c(fishcomb$DEindices,invnormcomb$DEindices))
> FC.selecDE <- data.frame(DEresults[unionDE,],do.call(cbind,FC)[unionDE,],
+   signFC=commonsgnFC[unionDE], DE=param$DE[unionDE])
> keepDE <- FC.selecDE[which(abs(FC.selecDE$signFC)==1),]
> conflictDE <- FC.selecDE[which(FC.selecDE$signFC == 0),]
> dim(FC.selecDE)

[1] 1468    8

> dim(keepDE)

[1] 1252    8

> dim(conflictDE)

[1] 216   8

> head(keepDE)

   DE.study1 DE.study2 DE.fishercomb DE.invnorm        FC1
4          1         1             1          1  1.3953545
11         1         1             1          1 -0.9995552
22         1         1             1          1 -1.1846747
36         1         1             1          1 -3.0703646
55         0         1             1          1 -0.4229671
59         1         1             1          1  1.1465211
          FC2 signFC   DE
4   2.0827031      1 TRUE
11 -0.5666576     -1 TRUE
22 -0.9829686     -1 TRUE
36 -2.8604490     -1 TRUE
55 -1.0557672     -1 TRUE
59  1.3520264      1 TRUE
```

Note that out of all the conflicts, 150 represented genes were simulated to be truly differentially expressed.

```
> table(conflictDE$DE)

FALSE  TRUE
   66   150
```

6 IDD, IRR and Venn Diagram

Different indicators can be used to evaluate the performance of the meta-analysis, some of them are described in [3] and returned by the function `IDD.IRR`. DE corresponds to the number of differentially expressed genes. IDD (Integration Driven discoveries) returns the number of genes that are declared DE in the meta-analysis that were not identified in any of the individual studies alone, Loss the number of genes that are identified DE in individual studies but not in meta-analysis. The Integration-driven Discovery Rate (IDR) and Integration-driven Revision Rate (IRR) are the corresponding proportions of IDD and Loss.

```
> fishcomb_de <- rownames(keepDE)[which(keepDE[,"DE.fishercomb"]==1)]
> invnorm_de <- rownames(keepDE)[which(keepDE[,"DE.invnorm"]==1)]
> indstudy_de <- list(rownames(keepDE)[which(keepDE[,"DE.study1"]==1)],
+                     rownames(keepDE)[which(keepDE[,"DE.study2"]==1)])
> IDD.IRR(fishcomb_de,indstudy_de)

     DE     IDD    Loss     IDR     IRR
1248.00   18.00    0.00    1.44    0.00

> IDD.IRR(invnorm_de ,indstudy_de)

     DE     IDD    Loss     IDR     IRR
1217.00   22.00   35.00    1.81    2.85
```

In this example, the p-value combination technique with Fisher's method gives 18 (1.44%) new genes and 0 (0%), are sidetracked. The inverse normal combination technique gives 22 (1.81%) new genes and 35 (2.85%), are sidetracked

To compare visually the number of differentially expressed genes in individual studies or in meta-analysis, it is also possible to draw a Venn diagram, for example with the *VennDiagram* package.

```
>  if (require("VennDiagram", quietly = TRUE)) {
+   venn.plot<-venn.diagram(x = list(study1=which(keepDE[,"DE.study1"]==1),
+                                    study2=which(keepDE[,"DE.study2"]==1),
+                                    fisher=which(keepDE[,"DE.fishercomb"]==1),
+                                    invnorm=which(keepDE[,"DE.invnorm"]==1)),
+                           filename = NULL, col = "black",
+                           fill = c("blue", "red", "purple","green"),
+                           margin=0.05, alpha = 0.6)
+   jpeg("venn_jpeg.jpg");
+   grid.draw(venn.plot);
+   dev.off();
+ }
```

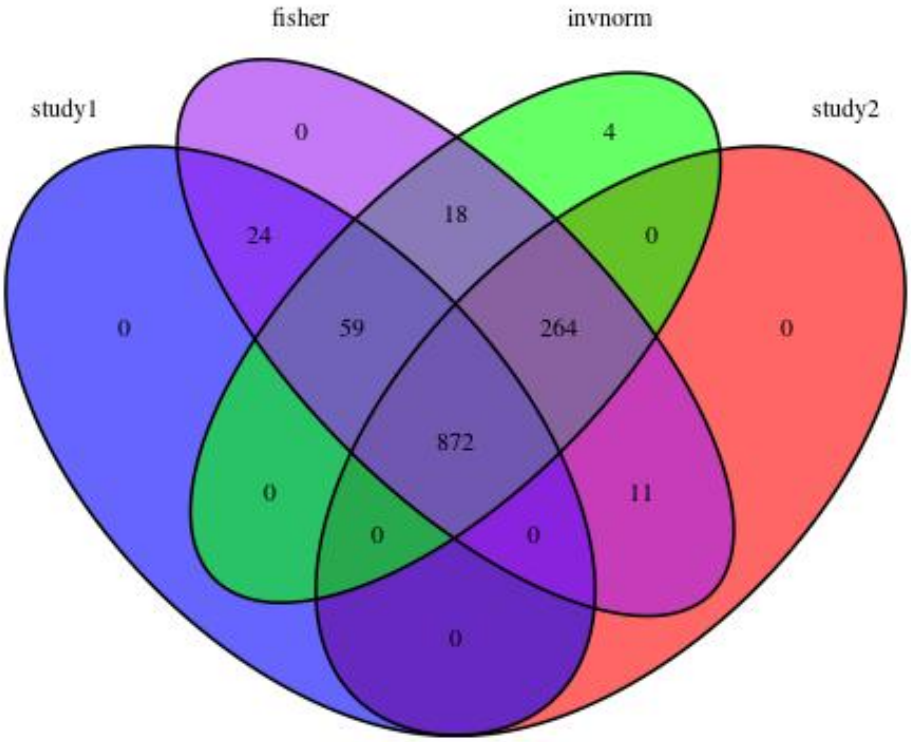

Figure 4: Venn Diagram comparing the list of DE genes at a 5% BH threshold obtained by each individual study and p-value combination techniques

7 Session Info

```
> sessionInfo()

R version 3.6.3 (2020-02-29)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Debian GNU/Linux bullseye/sid

Matrix products: default
BLAS:   /srv/R/R-patched/build.20-04-27/lib/libRblas.so
LAPACK: /srv/R/R-patched/build.20-04-27/lib/libRlapack.so
```

```
locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] grid      stats     graphics  grDevices utils
[6] datasets  methods   base

other attached packages:
[1] VennDiagram_1.6.20  futile.logger_1.4.3
[3] metaRNASeq_1.0.3

loaded via a namespace (and not attached):
 [1] SummarizedExperiment_1.18.2 genefilter_1.70.0
 [3] tidyselect_1.1.0            locfit_1.5-9.4
 [5] purrr_0.3.4                 splines_3.6.3
 [7] lattice_0.20-41             colorspace_1.4-2
 [9] vctrs_0.3.4                 generics_0.0.2
[11] stats4_3.6.3                blob_1.2.1
[13] survival_3.2-7              XML_3.99-0.3
[15] rlang_0.4.7                 pillar_1.4.6
[17] glue_1.4.2                  DBI_1.1.0
[19] BiocParallel_1.22.0         BiocGenerics_0.34.0
[21] bit64_4.0.5                 RColorBrewer_1.1-2
[23] lambda.r_1.2.4              matrixStats_0.57.0
[25] GenomeInfoDbData_1.2.3      lifecycle_0.2.0
[27] zlibbioc_1.34.0             munsell_0.5.0
[29] gtable_0.3.0                DESeq2_1.28.1
[31] memoise_1.1.0               Biobase_2.48.0
[33] geneplotter_1.66.0          IRanges_2.22.2
[35] GenomeInfoDb_1.24.2         parallel_3.6.3
[37] AnnotationDbi_1.50.3        Rcpp_1.0.5
[39] xtable_1.8-6                formatR_1.7
[41] scales_1.1.1                DelayedArray_0.14.1
[43] S4Vectors_0.26.1            annotate_1.66.0
[45] XVector_0.28.0              bit_4.0.4
[47] ggplot2_3.3.2               digest_0.6.25
```

```
[49] dplyr_1.0.2                 GenomicRanges_1.40.0
[51] tools_3.6.3                 bitops_1.0-6
[53] magrittr_1.5                RCurl_1.98-1.2
[55] RSQLite_2.2.1               tibble_3.0.3
[57] futile.options_1.0.1        crayon_1.3.4
[59] pkgconfig_2.0.3             ellipsis_0.3.1
[61] Matrix_1.3-0                R6_2.4.1
[63] compiler_3.6.3
```

References

[1] S. Anders and W. Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(R106):1–28, 2010.

[2] M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, G. Guernec, B. Jagla, L. Jouneau, D. Laloë, C. Le Gall, B. Schaëffer, S. Le Crom, M. Guedj, and F. Jaffrézic. A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Briefings in Bioinformatics*, 14(6):671–683, 2013.

[3] G. Marot, J.-L. Foulley, C.-D. Mayer, and F. Jaffrézic. Moderated effect size and P-value combinations for microarray meta-analyses. *Bioinformatics*, 25(20):2692–2699, 2009.

[4] A. Rau, G. Marot, and F. Jaffrézic. Differential meta-analysis of RNA-seq data from multiple studies. *BMC Bioinformatics*, 15(1):91, 2014.

[5] T. Strub, S. Giuliano, T. Ye, C. Bonet, C. Keime, D. Kobi, S. Le Gras, M. Cormont, R. Ballotti, C. Bertolotto, and I. Davidson. Essential role of microphthalmia transcription factor for DNA replication, mitosis and genomic stability in melanoma. *Oncogene*, 30:2319–2332, 2011.