RNA-seq Differential Expression (DE) Analysis Using edgeR

Jessica Randall

EMORY
UNIVERSITY

Emory Integrated
Computational Core
Emory Integrated Core Facilities

Briefly, edgeR uses a negative binomial distribution to determine the likelihood of differential expression between comparisons of two or more groups. This distribution mathematically accounts for the fact that we are assessing gene counts and we are assuming that most genes we are comparing between the groups will not be differentially expressed. The package is flexible in that it provides the user options to use empirical Bayes methods or frequentist methods which rely on the exact test, linear models, and quasi-likelihood tests.

Linked is the original 2010 paper from Robinson, McCarthy, and Smyth introducing the concepts implemented in edgeR.

There are two additional papers exploring improved functionality. One from McCarthy, Chen, and Smyth in 2012 on accounting for biological variation and one from Chen, Lun, and Smyth in 2016 on establishing an example workflow for package functions

We will be using the pasilla package for our example data.

edgeR has wide range of applications to genomic analyses. Our simplified example describes the quasi-likelihood F-test which is the method in edgeR which provides the strictest control of potential false positive genes. We strongly encourage you to reach out to EICC with questions regarding options available to you with edgeR. Check out some of the graphs from previous projects here.

Definition of terms

0.0.0.1 Quasi-likelihood F Test: Unadjusted p values vs adjusted p values/(FDR): In DE analysis, a single p-value tells you how likely it is that a single gene is differentially expressed between at least two groups (ex: a control and a treatment group) due to some actual difference between the groups as opposed to random chance. False Discovery Rate (FDR) tells you how likely it is that all genes identified as DE are false positives. A FDR of 5% means that among all genes called DE, an average of 5% of those are truly not DE.DE genes are only considered significantly so if they meet the adjusted p value, not only the unadjusted p value.

Compared to DESeq2 it has been our experience that edgeR is more stringent in its calling of genes as significantly DE. If your study is exploratory and are not sure which or how many DE genes you are expecting to find in your experiment, edgeR may not be the best package for your analysis.

0.0.0.2 Unadjusted p values vs adjusted p-values/(FDR): In DE analysis, a single p-value tells you how likely it is that a single gene is differential expressed between at least two groups (ex: a control and a treatment group) due to some actual difference between the groups as opposed to random chance. False Discovery Rate (FDR) tells you how likely it is that all genes identified as DE are false positives. A FDR of 5% means that among all genes called DE, an average of 5% of those are truly not DE. DE genes are only considered significantly so if they meet the adjusted p value, not only the unadjusted p-value. FDRs for each individual gene are called q-values or local FDRs.

Loading data

Our very first step is to load the libraries we'll need to assess the functions required for analysis and graphing. Please see Bioconductor for information about initial installation and use of Bioconductor and its packages. We also set the minimal theme in gglot2 for all graphs to have the same aesthetic features by default.

The pasilla experiment studied RNAi knockdown of Pasilla, the Drosophila melanogaster ortholog of mammalian NOVA1 and NOVA2, on the transcriptome. Data are provided by NCBI Gene Expression Omnibus under accession numbers GSM461176 to GSM461181.

Here we will demonstrate importing the count matrix and sample data from the pasilla package since we're using it as an example. Typically we will use the here package to specify the path for the counts and sample data files in a list of files to import and export from the task.

We're also going to specify that we'd like the row names of our sample data to come from the first column, called "file" since this is where we've stored which sample is which and finally we remove extra columns from our sample data which we won't be using in our analysis.

Please reach out to EICC if you would like to compare 3 or more groups as this is a simplified example. It may also be the case you will need more than 6 samples per experimental group or that you may need to remove genes with average counts greater than 5, 10, 15, or even 20 for sufficient statistical power. Please see our PROPER walk-through for an example of our of power and sample size analysis.

Preparing for Analysis

In order to preform a pairwise comparison we need to specify some information about our data. In edgeR we must create a special object called a DGElist object, here abbreviated as y.

This object takes in the countdata object and the sample data comparison variable of interest as a vector to create the DGEList object and prepare for analysis.

Since we want to know if edgeR filters out any additional genes with low counts, we create another object called y2 to perform additional filtering. We save the unfiltered DGElist object as y so we can check how many genes we are starting with and how many are filtered for low counts later on.

Since edgeR looks at data alphabetically, we also need to make sure we specify the untreated group is our reference group with the ref="untreated" statement.

```r
y <- DGEList(counts = countdata,
             group = sampledata$condition)

y2<-y

y2$samples$group <- relevel(y2$samples$group,
                            ref="untreated")
```

Design Matrix

The design matrix is a mathematical representation of our experimental design. This is the way the quasi-likelihood F test needs to see our experiment represented in order to perform the comparisons we would like.

Think of the first line like a simple linear equation: y= B0 + x * B1, y is our outcome of interest that we are expecting will vary based on the B0 and x * B1 terms. In our case y is gene expression ~ 0 is our intercept term B0 and this tells us how much additional variability we have in our experiment if being either a case or a control had no effect on gene expression. Group is our variable of interest x*B1. We would like to calculate how much being in one group or another affects the expression of each gene. The next two lines are renaming the column names of the design matrix to be the same as the levels in the samples, either "Control" or "Treatment" in our case and the row names to be the same as the column names in the cpm which are the IDs of our samples.

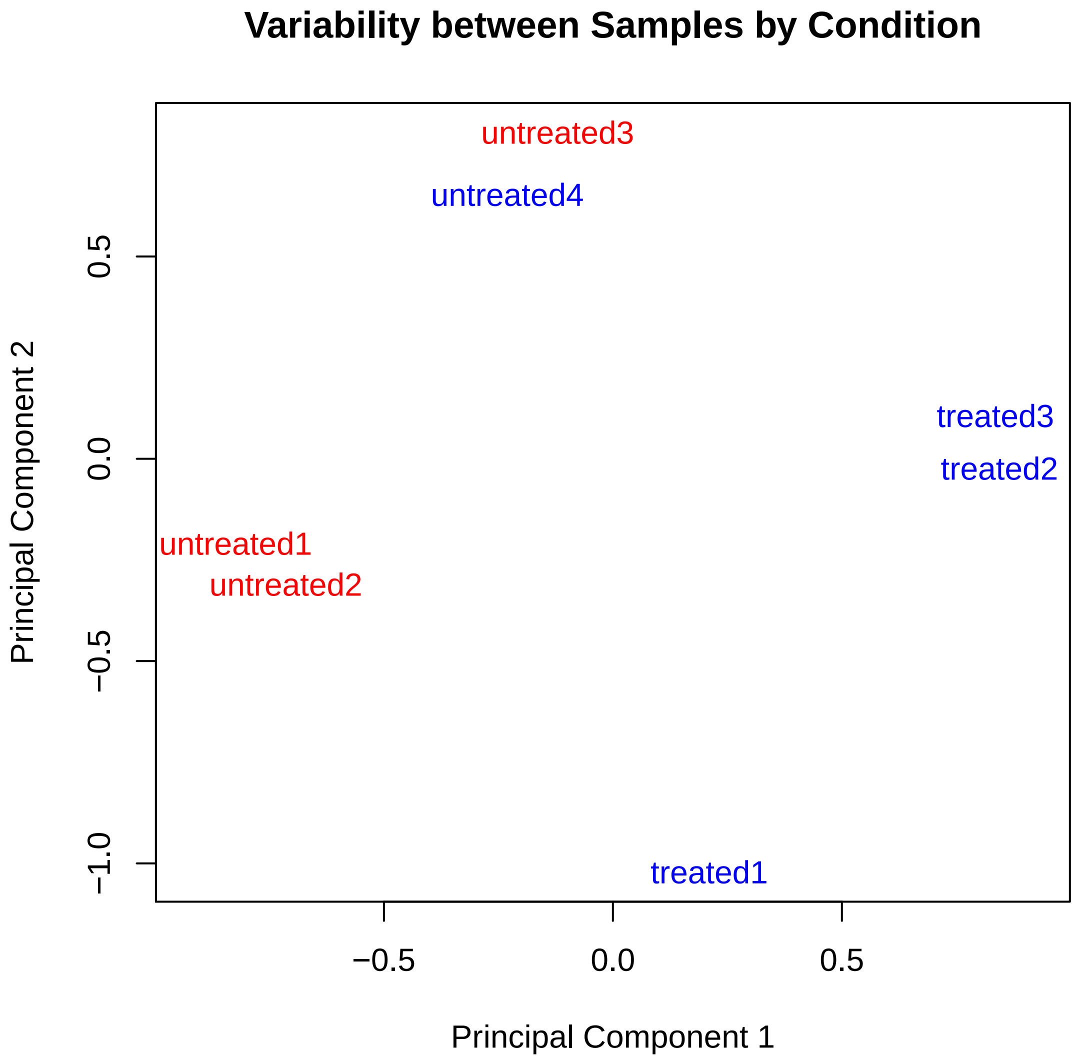
```
design <-model.matrix(~ 0 + group, data = y2$samples)

colnames(design) <- levels(y2$samples$group)
rownames(design) <- colnames(y2$counts)
```

If we had any additional data to add about the samples that we wanted to include in our analysis we would add it next but since this is a simplified example, we are only comparing treated and control samples without taking into account any additional information about them.

At this point we generate our first exploratory visualization, the principal components analysis plot. This will show us how your data cluster or how similar each sample is to others of the same group.

By default, edgeR produces a general multi-dimensional scaling plot that tells us how closely related the samples are to each other by log fold changes. We can specify that we would like to view the special case of a PCA plot which tells us how much group membership (i.e. being untreated or treated) contributes to the differences between the samples and we hope this will be high.



This is a great looking PCA plot because it shows that it is our samples are clustered by so we can say that the experimental condition is largely responsible for the variation between the samples rather than noise generated by sequencing, analysis, or some other unaccounted for variable.

I might also say that the within-group variability between samples in the treated group is probably contributing some noise to our ability to detect differences between the treated and untreated groups. The y-axis tells us how much variability between these samples is due to other factors in our model or if we have none, sources of variability we may not have accounted for like sex or ethnicity which are often leading contributors of variability between samples and should be accounted for in experimental design if you wish to control for their effects.

Prefiltering

Typically we want to ignore genes that have counts of zero across all samples since these are adding statistical noise. We may also want to be more stringent and remove genes with rows that sum to 10, 20 or even 30 or less since these could also be contributing statistical noise that would prevent us from seeing the differences between the cases and controls, especially if it is a small one. Think of it like removing the static from an analog TV. Once it's gone, the picture is more clear.

For this example, we will use the filterbyExpr function to filter on counts per million (cpm) and let edgeR choose the filtering cutoff for us based on the data. We can even check how many genes were filtered at this step. We see that 6680 genes were filtered out and 7919 were kept in for analysis.

If you choose to use EICC and edgeR specifically to do your differential expression analysis, we do have the option to customize this part of the analysis based on your data and are able to filter more or fewer lowly expressed genes as needed.

```
keep <- filterByExpr(y2, design)

table(keep)
```

```
## keep
## FALSE  TRUE
##  6680  7919
```

Recalculate library size

Since we've removed genes with low counts we need to recalculate the library size before we normalize it for analysis. DESeq2 does this sort of behind the scenes from what a typical user would see but edgeR requires the user to do this by hand.

```
y2 <- y2[keep, ,keep.lib.sizes=FALSE]
```

Normalizing data

There are a number of reasons we may want to normalizing data in edgeR. We want to account for gene length, GC content, sequencing depth, or sample specific effects. edgeR minimizes the effects of most of these concerns behind the scenes but we do have to do a little work ourselves to account for RNA composition. If you choose to do an RNA-seq project with EICC we would choose normalization strategies customized to your data.

The calcNormFactors function here and the trimmed mean of M-values (TMM) method specifically is used in most, if not all differential expression analysis because we assume that the majority of genes are not going to be DE. This function uses the original library size from the raw counts to create an effective library size which is scaled by the log-fold changes between the samples for most genes. This scaling prevents or at least minimizes the effects of very extreme log-fold changes skewing results. Normalization with TMM puts these varying library sizes on the same scale so we can compare the samples.

```
y2 <- calcNormFactors(y2, method="TMM")
```

Estimating biological variability between samples

In DE analysis we are testing the idea that gene expression varies between these groups, control and treatment, and we expect that most of the variation between the samples comes from them being either a control sample or a treatment sample. However, there will always be a certain amount of variability we cannot anticipate but that we can measure. This inherent biological variability, also called the biological coefficient of variation or BCV, also varies by the type of sample. In well controlled experiments technical replicates would have smaller BCVs than unrelated, unique individuals.

First, we estimate the dispersion which is the amount of variance in our experiment. This is critical to the analysis because we want genes that appear to be consistently high or low across samples to count more in analysis than genes that vary (which signals possible outlier genes) count less. We have specified that robust=TRUE that we may account for any outlier genes. If there are none, this does not affect estimation.

We then take the square root of the dispersion to give us the BCV. This check of the BCV serves as a measure of quality control for the analysis. Since our samples are from genetically similar samples, we would expect our BCV to be between 0.05-0.2.

```
y2 <- estimateDisp(y2, design, robust=TRUE)

bcv <-sqrt(y2$common.dispersion)

bcv
```

```
## [1] 0.1705351
```

Our biological variability is estimated to be 0.17 which makes sense for genetically similar individuals.

Testing

At this point we have prepared all of the data to be input into our quasi- likelihood F (QLF) test. Recall that the hypothesis we are testing is that there is no difference in gene expression between the treatment and control samples. We chose to use the QLF test because of all of the tests edgeR offers, it is the most up to date as of writing this and gives us the strongest control of false positive results. In this step we specify our comparison of interests, treated to untreated, we fit the QLF model, test it, and summarize our results.

Since in our example we have an idea of what we're expecting to find in terms of the number of DE genes, we want to be sure we have as few false positive DE genes as possible so that we know whichever we do see, we can confidently follow up on with a lab test.

The decision to use edgeR will vary by your experimental goals. If you use edgeR, the QLF test is always the recommended test because even if you are doing a simple pairwise comparison now, you can add in additional variables like sex, ethnicity, or batch with relative ease as compared to the other tests available. Please contact EICC if you would like to discuss your experimental goals so we can assist you in choosing the appropriate analysis tool.

```
##        -1*untreated 1*treated
## Down                        0
## NotSig                   7919
## Up                          0
```

Results

In this example we see that while controlling the adjusted p/FDR threshold at <0.05, we have no DE genes between these groups.

In our walkthorugh of DESeq2 we use the same example dataset and found that while controlling the adjusted p/FDR threshold at <0.05, we have 838 DE genes between these groups, 406 are more expressed in the treated samples or up-regulated and 432 are more expressed in the untreated samples or down-regulated. Up and down regulation refers to the group you have set as the control or reference group. In this case, we are comparing the treated samples to the untreated samples so the untreated samples are our reference group and we say genes are up or down regulated in comparison to this group.

Different analytical tools will often give you slightly different results and edgeR may be better for projects which have specified genes of interest in mind rather than exploratory projects since edgeR is much more strict with potential false positive results. While DESeq2 may allow more false positives into your significant results it also provides additional tools for evaluating their veracity before following up with a lab test.

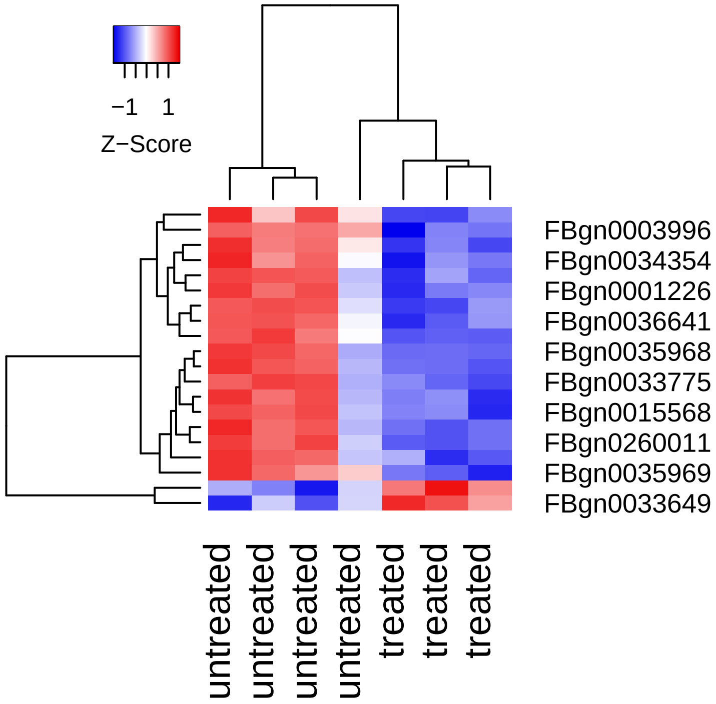
FDR thresholds can also range from 0.05-0.2 and it is much better to have the option to lower your threshold during analysis rather than have to increase it because your study was not sufficiently powered to find anything but the most extremely highly or lowly differentially expressed genes. Please reach out to us at EICC if you would like assistance in planning your experimental design for your RNAseq project and in setting appropriate FDR thresholds.

Finally, we create a data frame sorted by adjusted p-value that includes the log fold change difference between the groups, the log counts per million of each gene, the result of the F test statistics, the nominal p value, and the local adjusted p-value/q value for each gene along with that gene's Symbol. We check that the data frame was created successfully by using the informal unit test of dimension with the expected number of rows and columns and telling the program to stop if the file does not have these dimensions. After this runs successfully we typically export them as a .csv file for you.

Visualizing

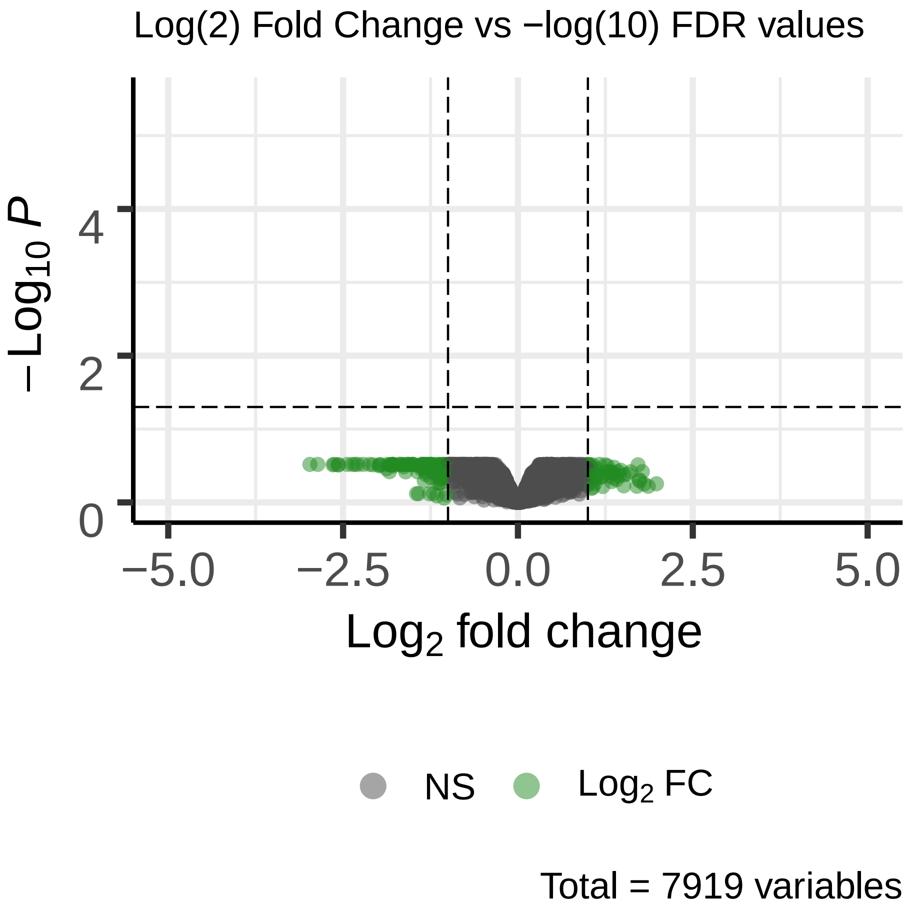
We now perform additional data visualizations. Typically we provide a PCA plot, heat maps, and volcano plots. We would be happy to work with you to customize these for publication. Please see our Data Visualization menu for more options and examples from previous projects.

Before we generate our heatmap in edgeR, we need to transform the counts in the DGEList object y) into counts per million (CPM) and then log transform them. In this example we have decided to sort the DE genes by smallest adjusted p value and only examine the top 20 DE genes. The number of DE genes you may want to visualize is customizable based on your project with EICC.



Please note that these are sorted for convenience but the gene at the top of the list is no more significant than the gene at the bottom of the list. As is the case with nominal p-values, a smaller adjusted p-value does not make a gene more statistically significant than one with a larger adjusted p-value. If the genes are below the threshold, they are all equally statistically significantly differential expressed. These are sorted for convenience but the gene at the top of the list is no more significant than the gene at the bottom of the list. We also typically provide clients with an initial volcano plot created with the EnhancedVolcano R library. Similar to the PCA plot and heat map, this is a highly customizable graph and we would like to work with you to design graphs which best tell the story of your results.

A volcano plot is technically a scatter plot where the x-axis has the log2 transformed fold changes between the compared samples and the y axis has the local adjusted p-values for each gene, also called the q-value. Here we have also labelled the genes with FDR < 0.1 as that is where we set our threshold when we generated our results. The points in red are those which meet the threshold for statistical significance with a q value less than or equal to 0.1 and a log2 fold change of 1.0 or greater. Points in green are those with only log2 fold changes >1.0 and those in blue have claques < 0.1. The points in grey are non statistically significant by any measure. All of these parameters can be adjusted based on your cutoffs and thresholds.



There are many more functions and many more specifications to functions than are used here in order to show a simplified example of one of the tools we use for differential expression analysis. Obtaining specific, actionable, and publication quality results from analysis requires a deeper understanding of your specific data set and we would love the opportunity to discuss these options with you.

While we encourage clients to reach out prior to sequencing so that we can collaborate to design the experiment to answer your specific questions, we look forward to hearing from you at any stage of your RNA-seq project. Please find our contact information available on our website and check out some of the graphs we've made for previous clients here.

Session information and References

```
## [1] "Wed Mar 04 13:07:22 2020"
```

```
## R version 3.6.2 (2019-12-12)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] EnhancedVolcano_1.4.0 ggrepel_0.8.1         ggplot2_3.2.1        
##  [4] edgeR_3.28.0          limma_3.42.0          tidyr_1.0.2          
##  [7] knitr_1.28            dplyr_0.8.4           abind_1.4-5          
## [10] readr_1.3.1           pacman_0.5.1         
## 
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.3         pillar_1.4.3       compiler_3.6.2     bitops_1.0-6      
##  [5] tools_3.6.2        statmod_1.4.34     digest_0.6.25      evaluate_0.14     
##  [9] tibble_2.1.3       lifecycle_0.1.0    gtable_0.3.0       lattice_0.20-38   
## [13] pkgconfig_2.0.3    png_0.1-7          rlang_0.4.4        yaml_2.2.1        
## [17] xfun_0.12          withr_2.1.2        stringr_1.4.0      caTools_1.18.0    
## [21] gtools_3.8.1       vctrs_0.2.3        hms_0.5.3          locfit_1.5-9.1    
## [25] grid_3.6.2         tidyselect_1.0.0   glue_1.3.1         R6_2.4.1          
## [29] rmarkdown_2.1      bookdown_0.17      gdata_2.18.0       farver_2.0.3      
## [33] purrr_0.3.3        magrittr_1.5       gplots_3.0.3       splines_3.6.2     
## [37] scales_1.1.0       htmltools_0.4.0    assertthat_0.2.1   colorspace_1.4-1  
## [41] labeling_0.3       KernSmooth_2.23-16 stringi_1.4.6      lazyeval_0.2.2    
## [45] munsell_0.5.0      crayon_1.3.4
```

```
## 
## To cite the 'bookdown' package in publications use:
## 
##   Yihui Xie (2020). bookdown: Authoring Books and Technical Documents
##   with R Markdown. R package version 0.17.
## 
##   Yihui Xie (2016). bookdown: Authoring Books and Technical Documents
##   with R Markdown. Chapman and Hall/CRC. ISBN 978-1138700109
## 
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```
## 
```

```
## To cite package 'readr' in publications use:
##
##   Hadley Wickham, Jim Hester and Romain Francois (2018). readr: Read
##   Rectangular Text Data. R package version 1.3.1.
##   https://CRAN.R-project.org/package=readr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {readr: Read Rectangular Text Data},
##     author = {Hadley Wickham and Jim Hester and Romain Francois},
##     year = {2018},
##     note = {R package version 1.3.1},
##     url = {https://CRAN.R-project.org/package=readr},
##   }
```

```
##
## To cite package 'abind' in publications use:
##
##   Tony Plate and Richard Heiberger (2016). abind: Combine
##   Multidimensional Arrays. R package version 1.4-5.
##   https://CRAN.R-project.org/package=abind
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {abind: Combine Multidimensional Arrays},
##     author = {Tony Plate and Richard Heiberger},
##     year = {2016},
##     note = {R package version 1.4-5},
##     url = {https://CRAN.R-project.org/package=abind},
##   }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```

```
##
## To cite package 'tidyr' in publications use:
##
##   Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R
##   package version 1.0.2. https://CRAN.R-project.org/package=tidyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Tidy Messy Data},
##     author = {Hadley Wickham and Lionel Henry},
##     year = {2020},
##     note = {R package version 1.0.2},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```
##
```

```
## To cite package 'dplyr' in publications use:
##
##   Hadley Wickham, Romain François, Lionel Henry and Kirill Müller
##   (2020). dplyr: A Grammar of Data Manipulation. R package version
##   0.8.4. https://CRAN.R-project.org/package=dplyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {dplyr: A Grammar of Data Manipulation},
##     author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller},
##     year = {2020},
##     note = {R package version 0.8.4},
##     url = {https://CRAN.R-project.org/package=dplyr},
##   }
```

```
##
## To cite the 'knitr' package in publications use:
##
##   Yihui Xie (2020). knitr: A General-Purpose Package for Dynamic Report
##   Generation in R. R package version 1.28.
##
##   Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition.
##   Chapman and Hall/CRC. ISBN 978-1498716963
##
##   Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible
##   Research in R. In Victoria Stodden, Friedrich Leisch and Roger D.
##   Peng, editors, Implementing Reproducible Computational Research.
##   Chapman and Hall/CRC. ISBN 978-1466561595
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```
##
## To cite package 'tidyr' in publications use:
##
##   Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R
##   package version 1.0.2. https://CRAN.R-project.org/package=tidyr
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {tidyr: Tidy Messy Data},
##     author = {Hadley Wickham and Lionel Henry},
##     year = {2020},
##     note = {R package version 1.0.2},
##     url = {https://CRAN.R-project.org/package=tidyr},
##   }
```

```
##
## To cite ggplot2 in publications, please use:
##
```

```
##   H. Wickham. ggplot2: Elegant Graphics for Data Analysis.
##   Springer-Verlag New York, 2016.
##
## A BibTeX entry for LaTeX users is
##
##   @Book{,
##     author = {Hadley Wickham},
##     title = {ggplot2: Elegant Graphics for Data Analysis},
##     publisher = {Springer-Verlag New York},
##     year = {2016},
##     isbn = {978-3-319-24277-4},
##     url = {https://ggplot2.tidyverse.org},
##   }
```

```
##
## See Section 1.2 in the User's Guide for more detail about how to cite
## the different edgeR pipelines.
##
##   Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor
##   package for differential expression analysis of digital gene
##   expression data. Bioinformatics 26, 139-140
##
##   McCarthy DJ, Chen Y and Smyth GK (2012). Differential expression
##   analysis of multifactor RNA-Seq experiments with respect to
##   biological variation. Nucleic Acids Research 40, 4288-4297
##
## To see these entries in BibTeX format, use 'print(<citation>,
## bibtex=TRUE)', 'toBibtex(.)', or set
## 'options(citation.bibtex.max=999)'.
```

```
##
## To cite package 'EnhancedVolcano' in publications use:
##
##   Kevin Blighe, Sharmila Rana and Myles Lewis (2019). EnhancedVolcano:
##   Publication-ready volcano plots with enhanced colouring and labeling.
##   R package version 1.4.0.
##   https://github.com/kevinblighe/EnhancedVolcano
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {EnhancedVolcano: Publication-ready volcano plots with enhanced colouring and
## labeling},
##     author = {Kevin Blighe and Sharmila Rana and Myles Lewis},
##     year = {2019},
##     note = {R package version 1.4.0},
##     url = {https://github.com/kevinblighe/EnhancedVolcano},
##   }
##
## ATTENTION: This citation information has been auto-generated from the
## package DESCRIPTION file and may need manual editing, see
## 'help("citation")'.
```