RNAseq turtorial for DEG analysis

Amarinder Singh Thind

14 sep, 2020

Contents

```
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Install and load packages

```
# if (!requireNamespace('BiocManager', quietly = TRUE))
# install.packages('BiocManager')

# BiocManager::install('DESeq2')
# BiocManager::install('edgeR')
# BiocManager::install('biomaRt')
# BiocManager::install('PCAtools')

library(edgeR)
library(DESeq2)
library("biomaRt")
```

load the raw count matrix

```
setwd("/Users/athind/Desktop/RNA-seq-tutorial-for-gene-differential-expression-analysis-master/")

rawcount <- read.table("RawGeneCounts.tsv", header = TRUE, sep = "\t",
    row.names = 1)
```

Filter for coding genes

```
mart <- useMart(biomart = "ensembl", dataset = "hsapiens_gene_ensembl")
all_coding_genes <- getBM(attributes = c("hgnc_symbol"), filters = c("biotype"),
    values = list(biotype = "protein_coding"), mart = mart)
rawcount <- rawcount[row.names(rawcount) %in% all_coding_genes$hgnc_symbol,
    ]
```

Meta data/ Data annotation

```
anno <- read.table("Annotation_of_samples.csv", header = TRUE,
    sep = ",")  ##In this case Two coulmns (a) sample (b) Condition
rownames(anno) <- anno$sample
```

Define conditions (for contrast) that you want to compare if you have more than one #control #case

This is pair-wise comparison, so only consider one pair at one time

```
firstC <- "case2"  #case1 #case2 #case3 etc
SecondC <- "Control"
p.threshold <- 0.05  ##define threshold for filtering
```

subset raw and conditional data for defined pairs

```
anno <- anno[(anno$Condition == firstC | anno$Condition == SecondC),
    ]
rawcount <- rawcount[, names(rawcount) %in% anno$sample]
```

DESeq2

create Desq2 datasets

```
dds <- DESeqDataSetFromMatrix(countData = rawcount, colData = anno,
    design = ~Condition)
```

```
## factor levels were dropped which had no samples
```

```
##   it appears that the last variable in the design formula, 'Condition',
##   has a factor level, 'Control', which is not the reference level. we recommend
##   to use factor(...,levels=...) or relevel() to set this as the reference level
##   before proceeding. for more information, please see the 'Note on factor levels'
##   in vignette('DESeq2').
```

Run DESEQ2

```
dds <- DESeq(dds)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing

## -- replacing outliers and refitting for 77 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

contrast based comparison

```
# In case of multiple comparisons ## we need to change the
# contrast for every comparision

contrast <- c("Condition", firstC, SecondC)
res <- results(dds, contrast = contrast)
res$threshold <- as.logical(res$padj < p.threshold)  #Threshold defined earlier

nam <- paste("down_in", firstC, sep = "_")
# res$nam <- as.logical(res$log2FoldChange < 0)
res[, nam] <- as.logical(res$log2FoldChange < 0)

genes.deseq <- row.names(res)[which(res$threshold)]

genes_deseq2_sig <- res[which(res$threshold), ]

file <- paste("Deseq2_", firstC, "_v_", SecondC, "_results_significant_padj0.05.csv",
    sep = "")
all_results <- paste("Deseq2_", firstC, "_v_", SecondC, "_all_results.csv",
    sep = "")

write.table(genes_deseq2_sig, file, sep = ",")
write.table(res, all_results, sep = ",")
```

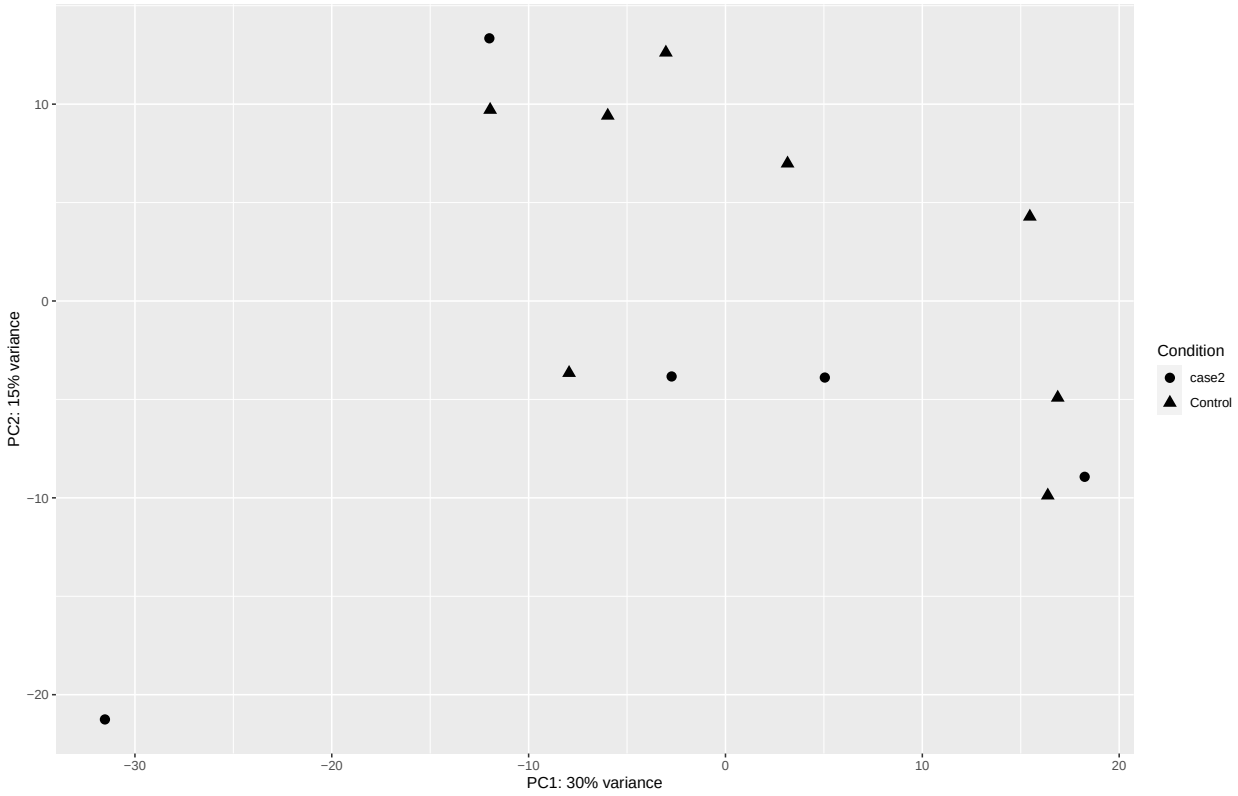
PCA and Heat-MAp Plots

Varinace transformation vst or rlog

```
vsd <- vst(dds, blind = FALSE)  #Variance type (a) Vst or (b) rlog
# rld <- rlog(dds, blind=FALSE)
```

PCA (Deseq2) with design consideration (Consider top 500 highest variable genes)

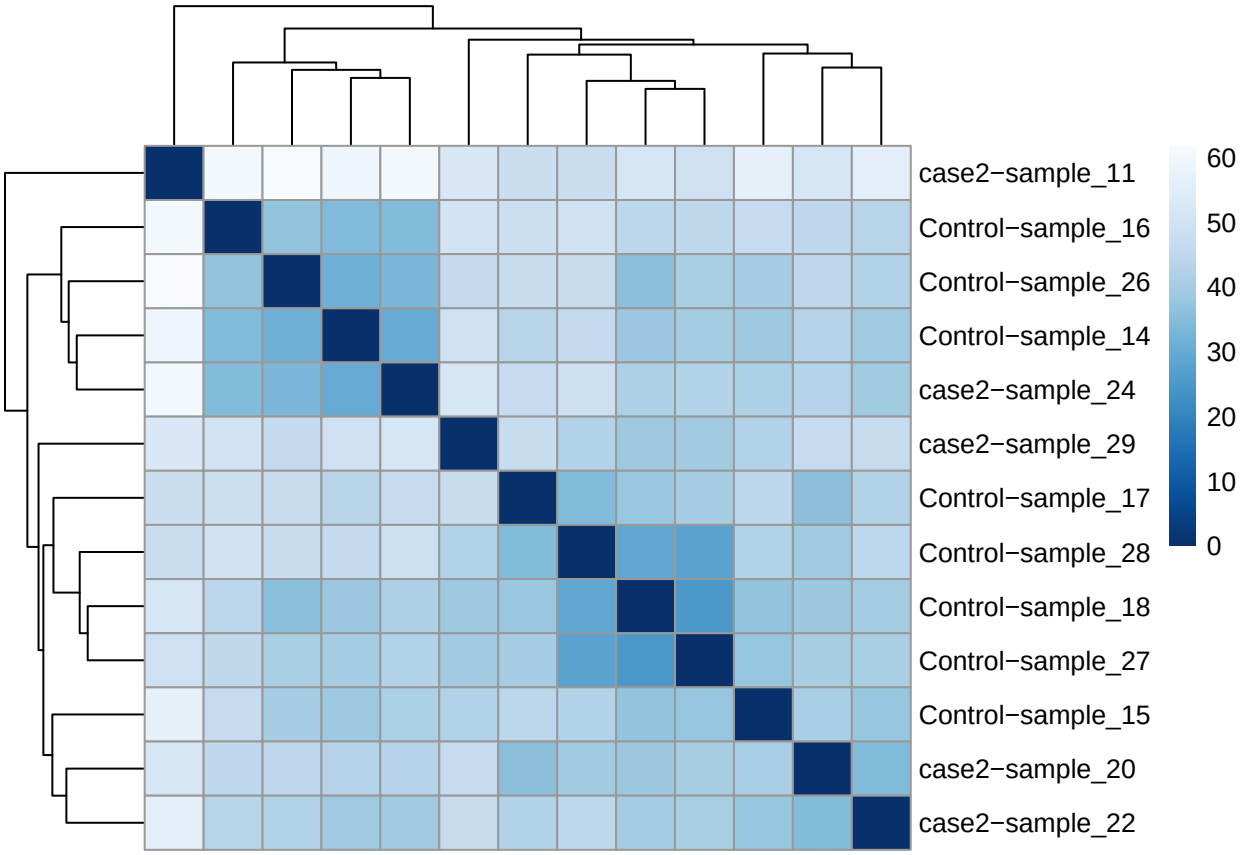
```
library(ggplot2)
pcaData <- plotPCA(vsd, intgroup=c("Condition", "sample"), returnData=TRUE)
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(PC1, PC2,  shape=Condition)) + #color=sample,
  geom_point(size=3) +
  xlab(paste0("PC1: ",percentVar[1],"% variance")) +
  ylab(paste0("PC2: ",percentVar[2],"% variance")) +
  coord_fixed()
```



heatmap

```
sampleDists <- dist(t(assay(vsd)))
library("RColorBrewer")
library("pheatmap")
sampleDistMatrix <- as.matrix(sampleDists)

rownames(sampleDistMatrix) <- paste(vsd$Condition, vsd$sample,
```

```
    sep = "-")

colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
pheatmap(sampleDistMatrix, clustering_distance_rows = sampleDists,
    clustering_distance_cols = sampleDists, col = colors)
```



edgeR

build edgeR data

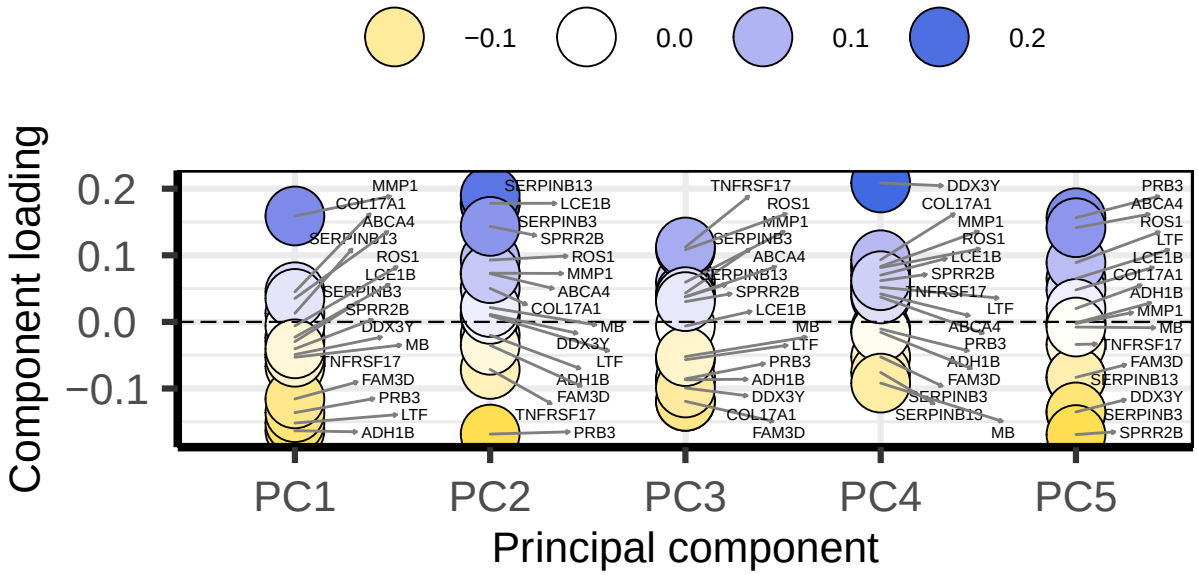
```
dge <- DGEList(counts = rawcount, group = anno$Condition)
```

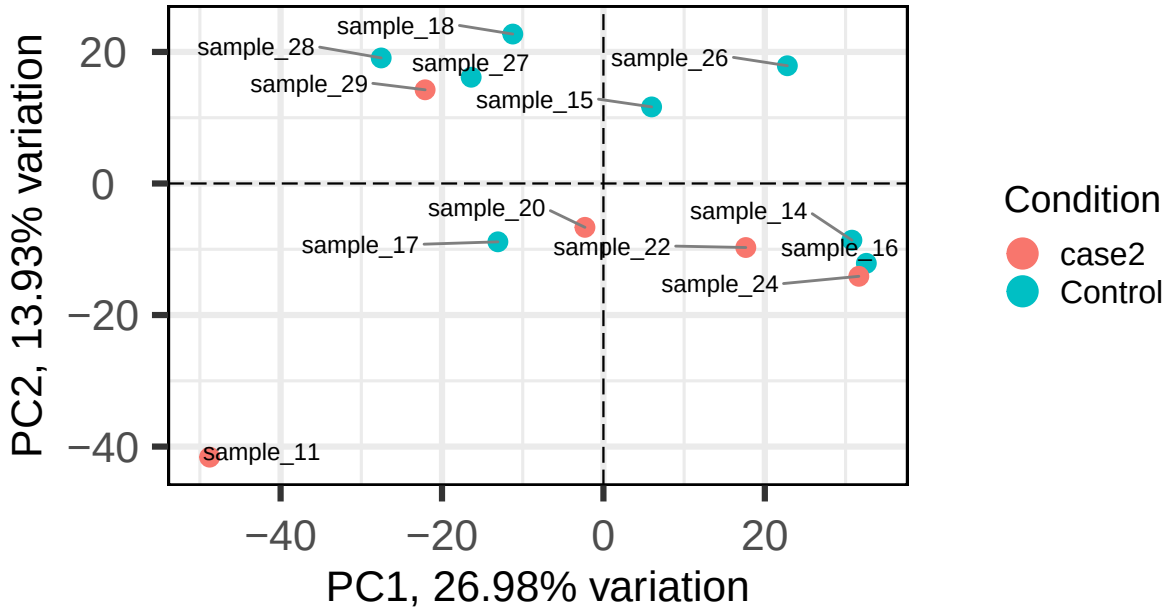
Normalization and PCA plot

PCA ## for more details, please visit following link https://bioconductor.org/packages/release/bioc/vignettes/PCAtools/inst/doc/PCAtools.html

```
## Loading required package: ggrepel
```

```
##
## Attaching package: 'PCAtools'

## The following objects are masked from 'package:stats':
##
##     biplot, screeplot

## -- removing the lower 20% of variables based on variance

## -- variables retained:

## MMP1, ADH1B, LTF, SERPINB3, LCE1B, SERPINB13, PRB3, ROS1, TNFRSF17, FAM3D, COL17A1, DDX3Y, MB, ABCA4
```





filter out lowly expressed genes

```
keep <- filterByExpr(dge)
dge <- dge[keep, , keep.lib.sizes = FALSE]
# It is recommended to recalculate the library sizes of the
# DGEList object after the filtering, although the downstream
# analysis is robust to whether this is done or not.
```

Create the contrast matrix

```
##    case2 Control
## 1      1       0
## 2      0       1
## 3      0       1
## 4      0       1
## 5      0       1
## 6      0       1
## 7      1       0
## 8      1       0
## 9      1       0
## 10     0       1
## 11     0       1
## 12     0       1
## 13     1       0
## attr(,"assign")
```

```
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$`dge$samples$group`
## [1] "contr.treatment"
```

Estimate dispersion parameter for GLM

```
dge <- estimateGLMCommonDisp(dge, design.mat)
dge <- estimateGLMTrendedDisp(dge, design.mat)
dge <- estimateGLMTagwiseDisp(dge, design.mat)
# Plot mean-variance plotBCV(dge)
```

Model fitting

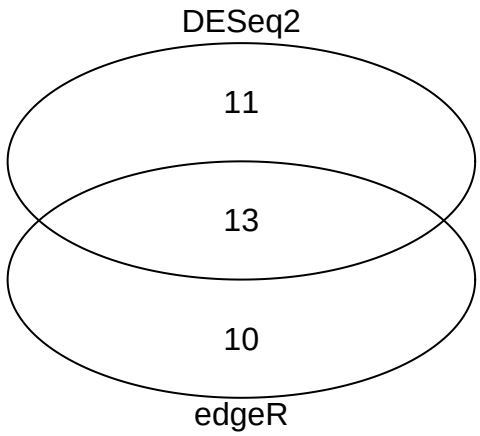
```
fit.edgeR <- glmFit(dge, design.mat)

# Differential expression

contrasts.edgeR <- makeContrasts(case2 - Control, levels = design.mat)  ##FirstC-SecondC ##Define


lrt.edgeR <- glmLRT(fit.edgeR, contrast = contrasts.edgeR)

##### DGE at padjust 0.05

# Access results tables
edgeR_results <- lrt.edgeR$table
sig.edgeR <- decideTestsDGE(lrt.edgeR, adjust.method = "BH",
    p.value = p.threshold)
# View(sig.edgeR)
significant_table <- edgeR_results[which(sig.edgeR != 0), ]
significant_table$gene <- row.names(significant_table)
genes.edgeR <- row.names(edgeR_results)[which(sig.edgeR != 0)]

edgeR_results$genes <- row.names(edgeR_results)


file_sigTab <- paste("edgeR_", firstC, "_v_", SecondC, "_results_significant_padj0.05.csv",
    sep = "")
file_allRes <- paste("edgeR_", firstC, "_v_", SecondC, "_all_results.csv",
    sep = "")

write.table(significant_table, file_sigTab, sep = ",")
write.table(edgeR_results, file_allRes, sep = ",")
```

Overlapped genes between deseq2 and edgeR

```
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space

## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

```
venn(list(edgeR = genes.edgeR, DESeq2 = genes.deseq))
```



```r
overlapped_genes <- intersect(genes.deseq, genes.edgeR)


file_common <- paste("Common_DEG_deseq2_edgeR_", firstC, "_v_",
    SecondC, ".csv", sep = "")
write.table(overlapped_genes, file_common, sep = ",", row.names = F)
```

Quick enrichment analysis

```r
# BiocManager::install('ReactomePA')
library(ReactomePA)
```

```
##
```

```
## Registered S3 method overwritten by 'enrichplot':
##   method               from
##   fortify.enrichResult DOSE
```

```
## ReactomePA v1.30.0  For help: https://guangchuangyu.github.io/ReactomePA
##
## If you use ReactomePA in published research, please cite:
## Guangchuang Yu, Qing-Yu He. ReactomePA: an R/Bioconductor package for reactome pathway analysis and
```

```r
all <- overlapped_genes  ## retreive EntrezGene id's

genes = getBM(attributes = c("hgnc_symbol", "entrezgene_id"),
    filters = "hgnc_symbol", values = all, bmHeader = T, mart = mart)
```

```
## Cache found
```

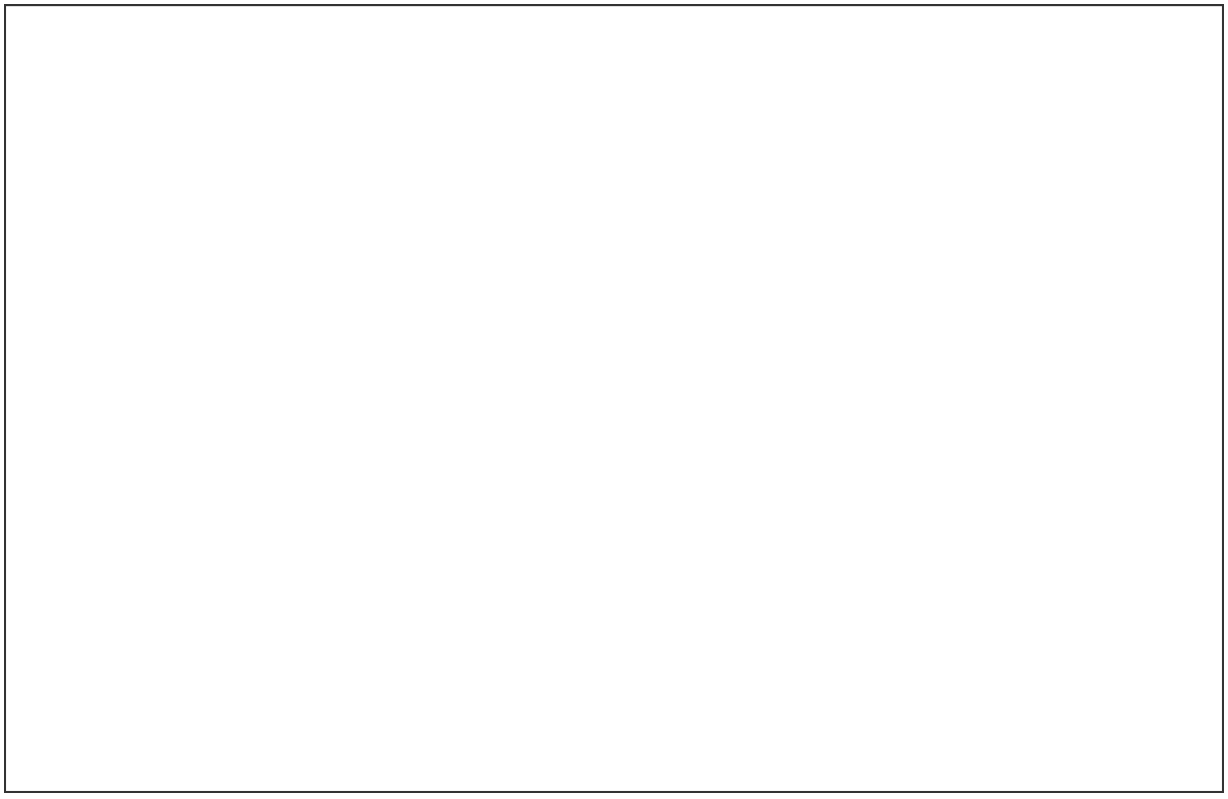
```r
genes1 <- genes$`NCBI gene (formerly Entrezgene) ID`



# ?enrichPathway #pvalueCutoff=0.02, #pAdjustMethod = 'BH',
# qvalueCutoff = 0.01,
x <- enrichPathway(gene = genes1, pvalueCutoff = 0.05, readable = T)
```

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

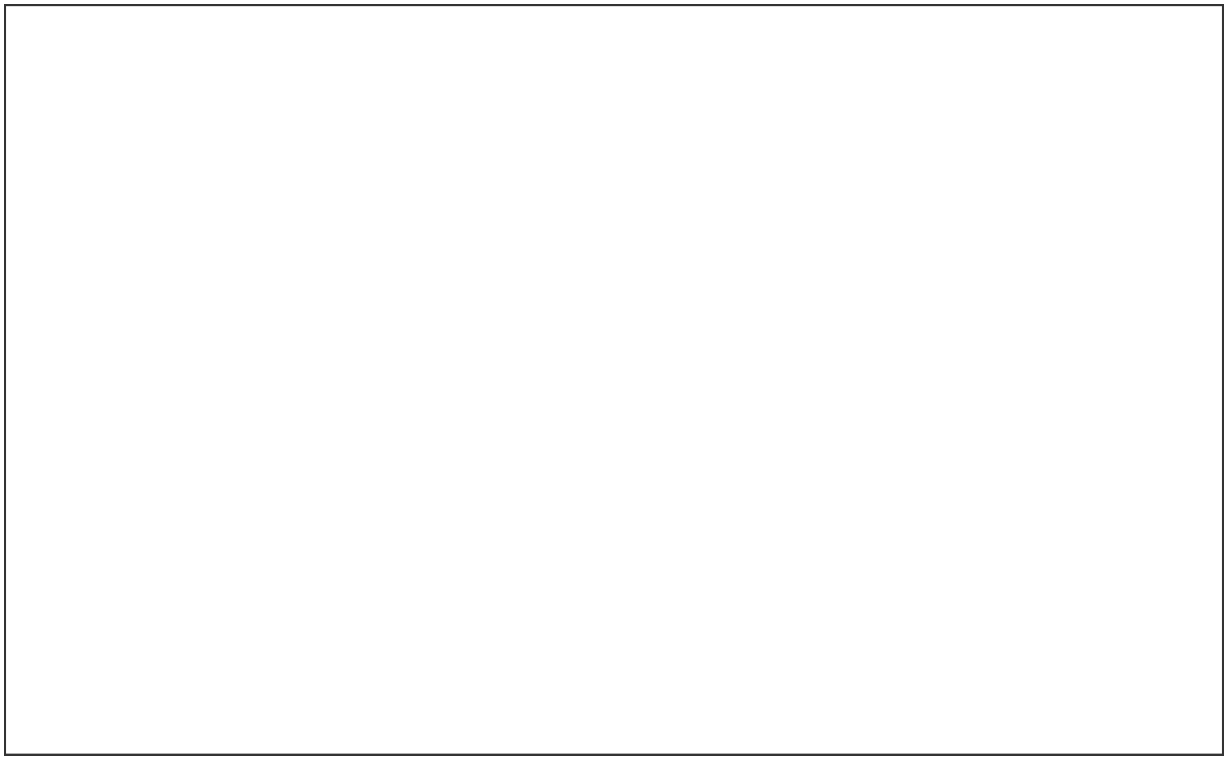
```
## Loading required package: AnnotationDbi
```

```
##
```

```r
# head(as.data.frame(x))
barplot(x, showCategory = 10)
```



```
dotplot(x, showCategory = 10)
```




```
# emapplot(x) cnetplot(x, categorySize='pvalue',
# foldChange=genes1) emapplot(x, color='pvalue')
# viewPathway('Extracellular matrix organization',
# readable=TRUE, foldChange=genes1) ## it's an example
```

Save session info

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Australia.1252  LC_CTYPE=English_Australia.1252
## [3] LC_MONETARY=English_Australia.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Australia.1252
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
```

```
## [8] methods   base
##
## other attached packages:
##  [1] org.Hs.eg.db_3.10.0         AnnotationDbi_1.48.0
##  [3] ReactomePA_1.30.0           gplots_3.0.4
##  [5] PCAtools_2.1.22             ggrepel_0.8.2
##  [7] pheatmap_1.0.12             RColorBrewer_1.1-2
##  [9] ggplot2_3.3.2               biomaRt_2.42.1
## [11] DESeq2_1.26.0               SummarizedExperiment_1.16.1
## [13] DelayedArray_0.12.3         BiocParallel_1.20.1
## [15] matrixStats_0.56.0          Biobase_2.46.0
## [17] GenomicRanges_1.38.0        GenomeInfoDb_1.22.1
## [19] IRanges_2.20.2              S4Vectors_0.24.4
## [21] BiocGenerics_0.32.0         edgeR_3.28.1
## [23] limma_3.42.2                knitr_1.29
##
## loaded via a namespace (and not attached):
##   [1] backports_1.1.7          Hmisc_4.4-1              fastmatch_1.1-0
##   [4] BiocFileCache_1.10.2     plyr_1.8.6               igraph_1.2.5
##   [7] splines_3.6.3            urltools_1.7.3           digest_0.6.25
##  [10] htmltools_0.5.0          GOSemSim_2.12.1          viridis_0.5.1
##  [13] GO.db_3.10.0             gdata_2.18.0             magrittr_1.5
##  [16] checkmate_2.0.0          memoise_1.1.0            cluster_2.1.0
##  [19] annotate_1.64.0          graphlayouts_0.7.0       askpass_1.1
##  [22] enrichplot_1.6.1         prettyunits_1.1.1        jpeg_0.1-8.1
##  [25] colorspace_1.4-1         blob_1.2.1               rappdirs_0.3.1
##  [28] xfun_0.16                dplyr_1.0.2              jsonlite_1.7.0
##  [31] crayon_1.3.4             RCurl_1.98-1.2           graph_1.64.0
##  [34] genefilter_1.68.0        survival_3.2-3           glue_1.4.1
##  [37] polyclip_1.10-0          gtable_0.3.0             zlibbioc_1.32.0
##  [40] XVector_0.26.0           graphite_1.32.0          BiocSingular_1.2.2
##  [43] scales_1.1.1             DOSE_3.12.0              DBI_1.1.0
##  [46] Rcpp_1.0.5               viridisLite_0.3.0        xtable_1.8-4
##  [49] progress_1.2.2           htmlTable_2.0.1          gridGraphics_0.5-0
##  [52] dqrng_0.2.1              reactome.db_1.70.0       europepmc_0.4
##  [55] foreign_0.8-75           bit_4.0.4                rsvd_1.0.3
##  [58] Formula_1.2-3            htmlwidgets_1.5.1        httr_1.4.2
##  [61] fgsea_1.12.0             ellipsis_0.3.1           pkgconfig_2.0.3
##  [64] XML_3.99-0.3             farver_2.0.3             nnet_7.3-14
##  [67] dbplyr_1.4.4             locfit_1.5-9.4           ggplotify_0.0.5
##  [70] tidyselect_1.1.0         labeling_0.3             rlang_0.4.7
##  [73] reshape2_1.4.4           munsell_0.5.0            tools_3.6.3
##  [76] generics_0.0.2           RSQLite_2.2.0            ggridges_0.5.2
##  [79] evaluate_0.14            stringr_1.4.0            yaml_2.2.1
##  [82] bit64_4.0.2              tidygraph_1.2.0          caTools_1.18.0
##  [85] purrr_0.3.4              ggraph_2.0.3             formatR_1.7
##  [88] xml2_1.3.2               DO.db_2.9                compiler_3.6.3
##  [91] rstudioapi_0.11          curl_4.3                 png_0.1-7
##  [94] tibble_3.0.3             tweenr_1.0.1             geneplotter_1.64.0
##  [97] stringi_1.4.6            lattice_0.20-41          Matrix_1.2-18
## [100] vctrs_0.3.2              pillar_1.4.6             lifecycle_0.2.0
## [103] BiocManager_1.30.10      triebeard_0.3.0          data.table_1.13.0
## [106] cowplot_1.0.0            bitops_1.0-6             irlba_2.3.3
## [109] qvalue_2.18.0            R6_2.4.1                 latticeExtra_0.6-29
```

```
## [112] KernSmooth_2.23-17       gridExtra_2.3            MASS_7.3-52
## [115] gtools_3.8.2             assertthat_0.2.1         openssl_1.4.2
## [118] withr_2.2.0              GenomeInfoDbData_1.2.2   hms_0.5.3
## [121] grid_3.6.3               rpart_4.1-15             tidyr_1.1.1
## [124] rvcheck_0.1.8            rmarkdown_2.3            DelayedMatrixStats_1.8.0
## [127] ggforce_0.3.2            base64enc_0.1-3
```

```
writeLines(capture.output(sessionInfo()), "sessionInfo.txt")
```