Report:Transcriptomic effect of Glucose vs Galactose in PDAC

Andrés Gordo Ortiz

2024-01-18

Contents

What can I find in this Report?

This analysis performs a fully crafted **Transcriptomics Protocol**, from read mapping, filtering and normalisation, up to Differential Gene Expression, Ontology enrichment or qualitative Isoform usage analysis.This a list of everything you will find within the folders provided.

Mapping Folder

1. **Kallisto**: All the output from *Kallisto*, the pseudoaligner of your reads to the reference genome.
2. **FastQC**: Report of the quality and depth of your reads for each sample.
3. **MultiQC**: Summarised report of the quality, depth and alignment success of your reads for every sample, summarized in a single convenient file.

Report Folder

Data

1. **log2__cpm.csv**: Excel file of the Counts per million of all genes across samples, non filtered, non normalised.

2.**log2__cpm__filtered.csv**: Excel file of the Counts per million of all genes across samples, filtered, non normalised.

3.**log2__cpm__filtered__norm.csv**: Excel file of the Counts per million of all genes across samples, filtered, normalised.

4. **pca__results.csv**: Excel file with the results and loadings of the PCA analysis.

Alignment Protocol Used

1. Data Acquisition

- Raw sequencing reads for the project (PRJEB10204) were obtained from the ENA.

2. Read Mapping

- The obtained raw reads were mapped to the human reference transcriptome GRCh38.cdna using Kallisto version 0.48.
  - Parameters used for Kallisto:
    - Average length of reads: 250
    - Standard deviation: 30

3. Quality Analysis

- The quality of the mapped reads was assessed using **fastqc** and **multiqc**.

4. Experimental Design

- The project includes a total of 20 single-end samples, comprising 10 replicates for each of the two experimental conditions: **with adherent media** and **with sphere media**.

5. Pseudo-alignment and Automated Script

- Pseudo-alignment of the samples to the human reference genome was performed using Kallisto. While a suitable code is provided for the alignment process, a custom, fully automated script Automatic Kallisto Gene aligner was developed *ad hoc* for streamlined analysis. The script requires the input of *.fastqc.gz* files and a cDNA reference genome.

6. Pre-processing

- For each sample, *fastqc* was executed to evaluate the sequencing quality before further analysis. If any samples lacked the quality required to perform downstream analysis that will be noted in its section.

7. **Data Integration**

- Following read mapping with Kallisto, the TxImport package was utilized to import Kallisto outputs into the R environment.

8. **Data Summarization**

- Annotation data from Biomart was employed to summarize the data from transcript-level to gene-level, providing a comprehensive view of the gene expression landscape.

Preprocessing

Filtering & normalisation

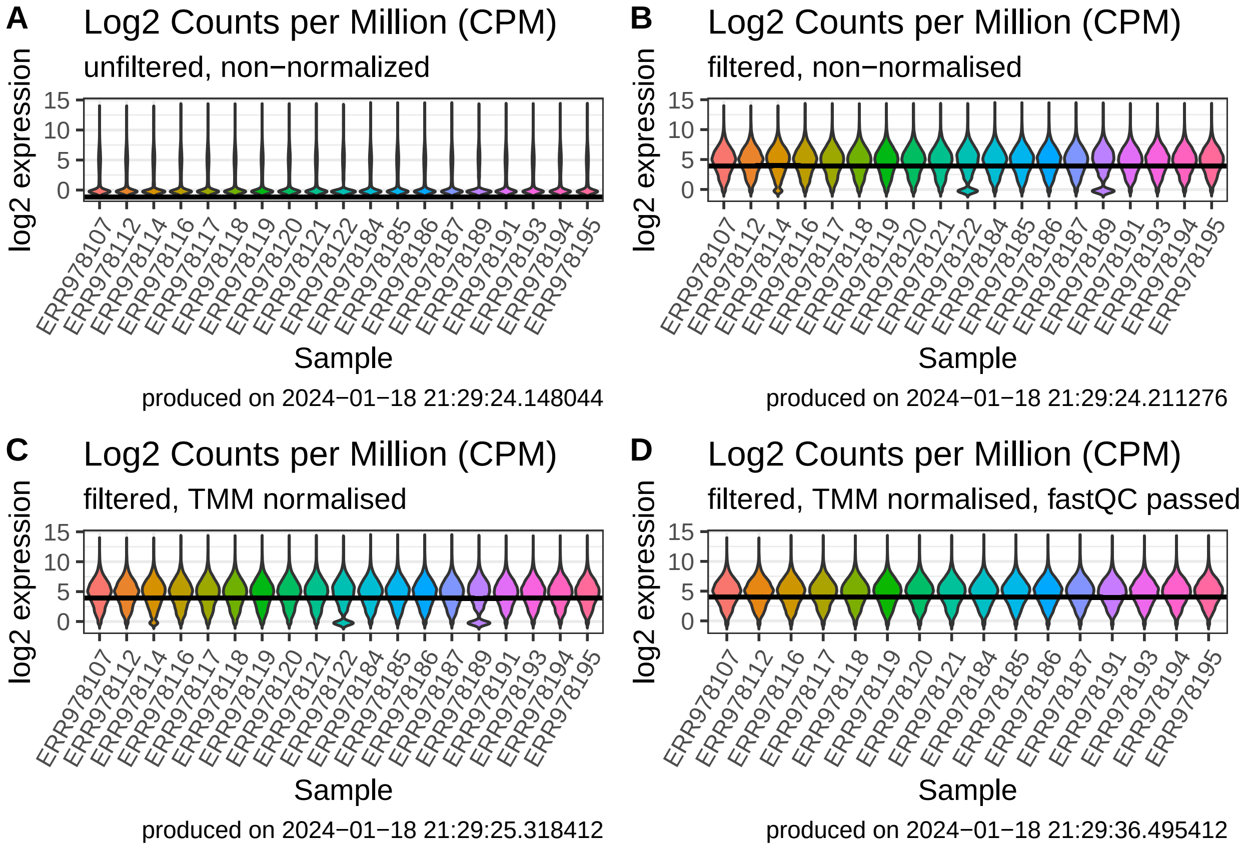


Figure 1: Violin plots of log2 CPM values for each sample. A) Raw reads after mapping. B) Genes with more than 9 samples with CPM < 1 were filtered out. C) Normalisation with TMM., D) Tidy Samples with a positive FastQC Result to be used in the downstream analysis

Filtering was carried out to remove lowly expressed genes. Genes with less than 1 count per million (CPM) in at least 9 or more samples were filtered out. This is done to make sure that the low expression is due to a lack of significance across **all** conditions, and not an intrinsic property of one of them. This filtering reduced the number of genes from 35371 to 13291. Normalisation of samples was performed with edgeR, using *Trimmed Mean of M-value* or *TMM*. This method is based on the assumption that most genes are not differentially expressed. It **calculates a scaling factor for each sample**, which is the median of the ratio of each gene's expression to the geometric mean of all samples. This scaling factor is then used to normalise the expression of each gene in each sample. The normalised expression is then expressed as log2(CPM). The normalisation step is important because it **allows for the comparison of expression between samples**.

According to *fastQC* results, a total of 4 samples were removed due to low quality. This is 20% of the total reads. Filtered out samples are: ERR978114, ERR978122, ERR978189, ERR978196.

Filtered and Normalised data

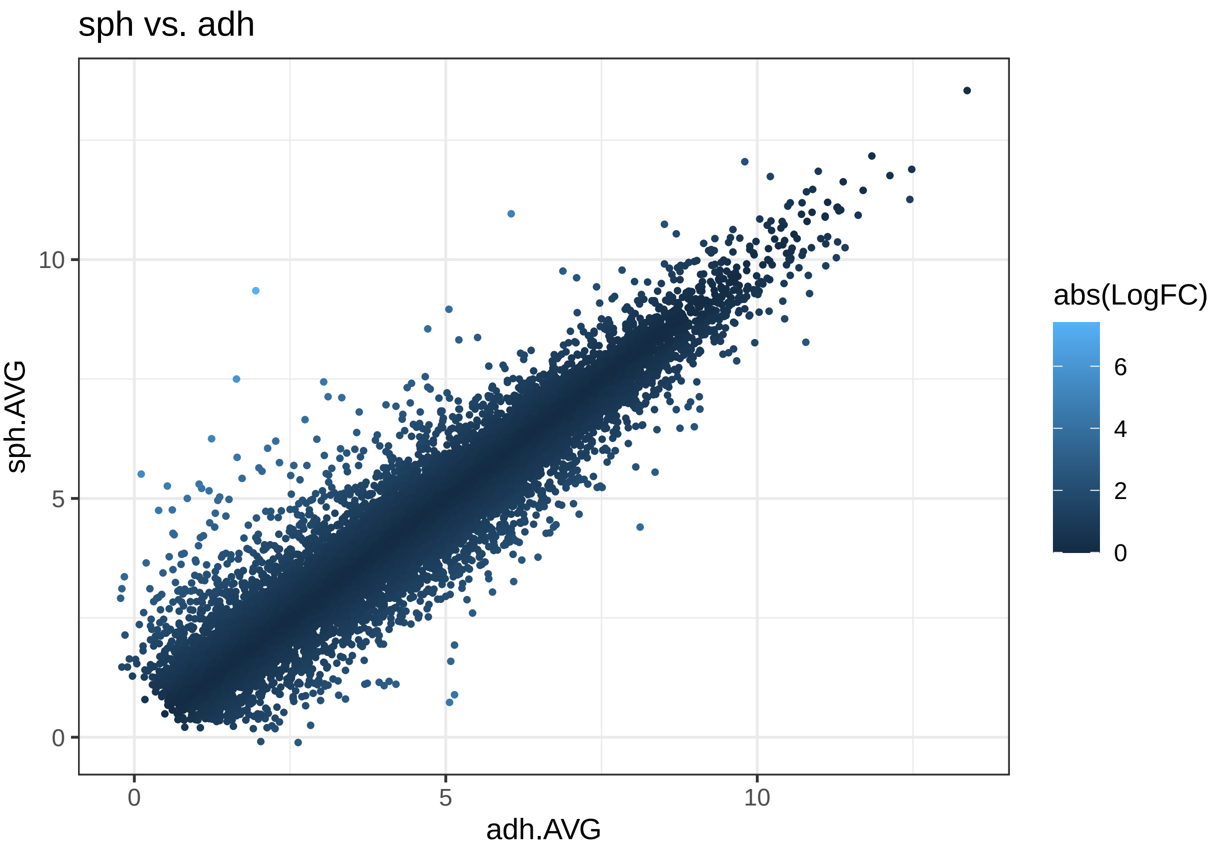


Figure 2: Scatter plot showing the divergence in the average expression of every gene between the two conditions

The **Table 1** includes expression data for 13291 genes. You can sort and search the data directly from the table in the *html* file.

PCA plot

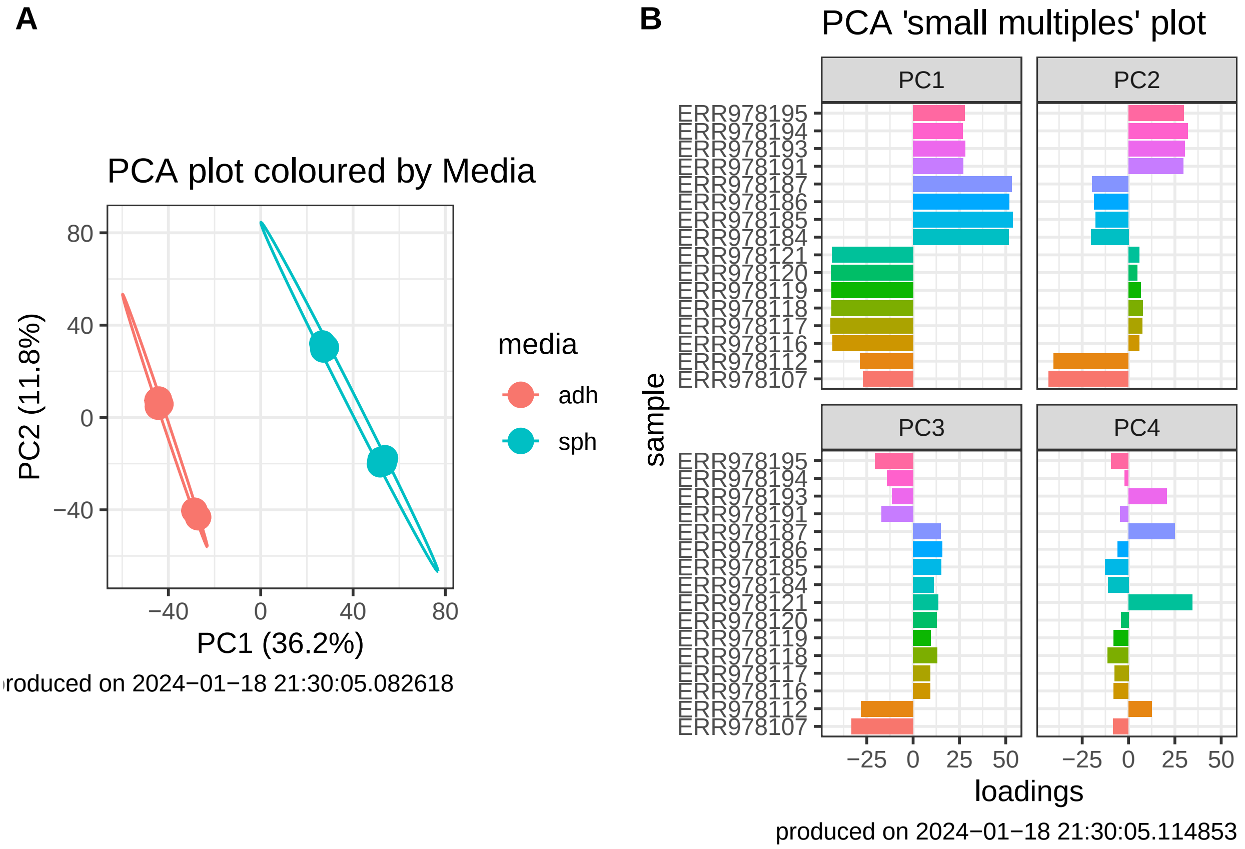


Figure 3: A) Principal Component Analysis plot coloured by condition. The 95% confidence interval is shown. B) Small Multiples Plot showing the loadings, or contribution, of sample gene to the first 4 principal components.

Volcano plot

An **important** note: all the contrasts, design matrix and differential expression analysis are made according to this formula ***(media:sph - media: adh)***. This means that a *logfold change* > 0 shows that the first condition is upregulated, whereas a *logfold change* < 0 shows downregulation.

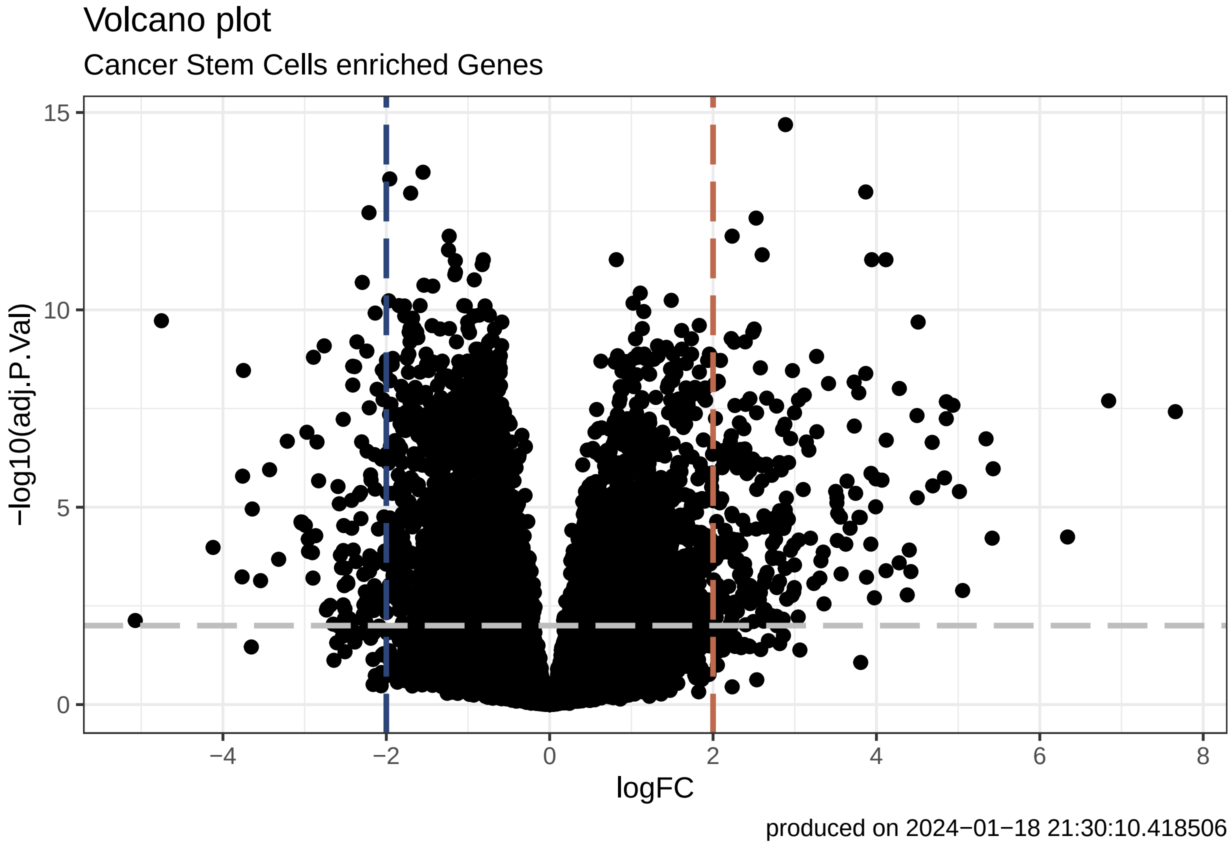


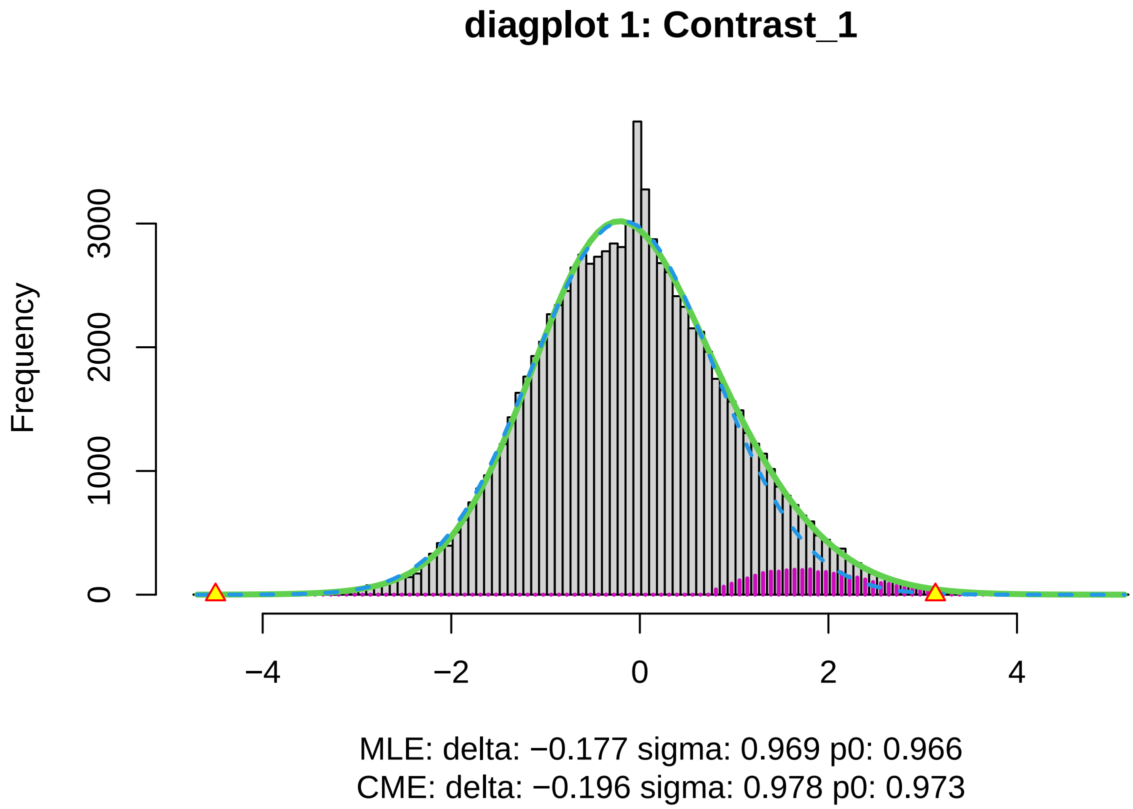
Figure 4: Volcano plot of differencial gene expression.The lines delimit the 2 log-fold change and 0.01 adjusted p-value (FDR) thresholds.

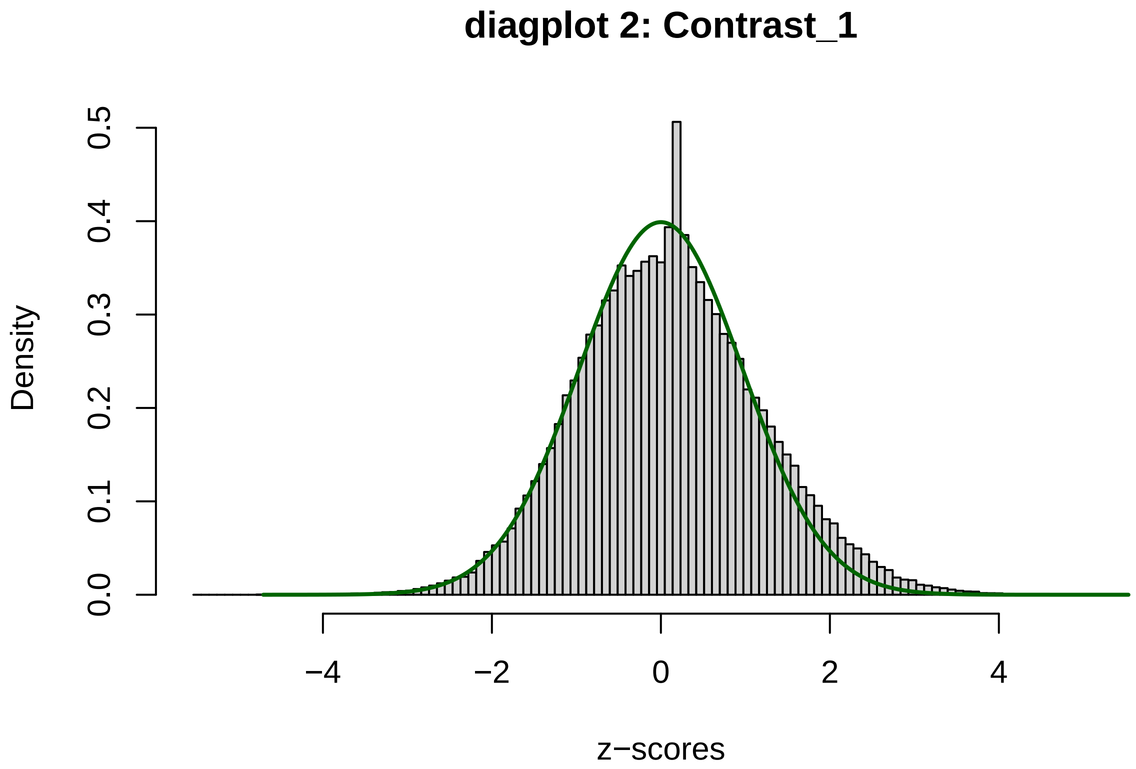
Table of DEGs

To identify differentially expressed genes, precision weights were first applied to each gene based on its mean-variance relationship using VOOM, then data was normalized using the TMM method in EdgeR. Linear modeling and bayesian stats were employed via Limma to find genes that were up- or down-regulated by **2-logfold or more, with a false-discovery rate (FDR) of 0.01**. An interactive table can be found in the *html* version of this report.

Isoform Switch Analyzer

Transcript isoforms expression and usage change was analysed through the well-established IsoformSwitchAnalyzeR. All differential **expressed/usage** genes will have their own *Switch Plot* and *Switch Table* can be found in the *html* version of this report. Here is the *Switch Plot* of the top 1 gene. All credits to: *Soneson et al. Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. F1000Research 4, 1521 (2015).*





```
##   Comparison nrIsoforms nrSwitches nrGenes
## 1 adh vs sph          9         11       8
```

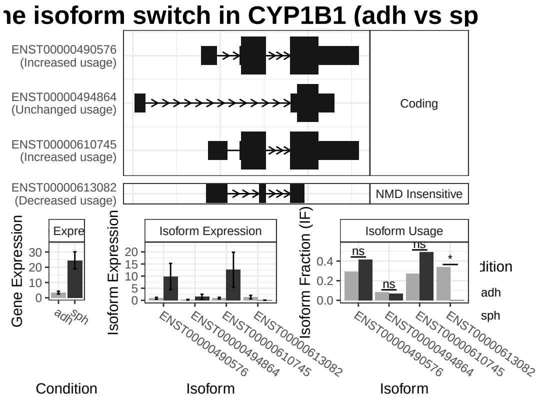


Figure 5: Sample of a SwitchPlot. All SwitchPlots generated can be found in their corresponding folder.

Heatmaps and Modules

Pearson correlation was used to cluster **316** differentially expressed genes, which were then represented as heatmap with the data scaled by *Zscore* for each row. On the other hand, Spearman correlation was used to cluster all samples. Two *modules* can be seen, being either up- or down-regulated depending on the condition of the samples.

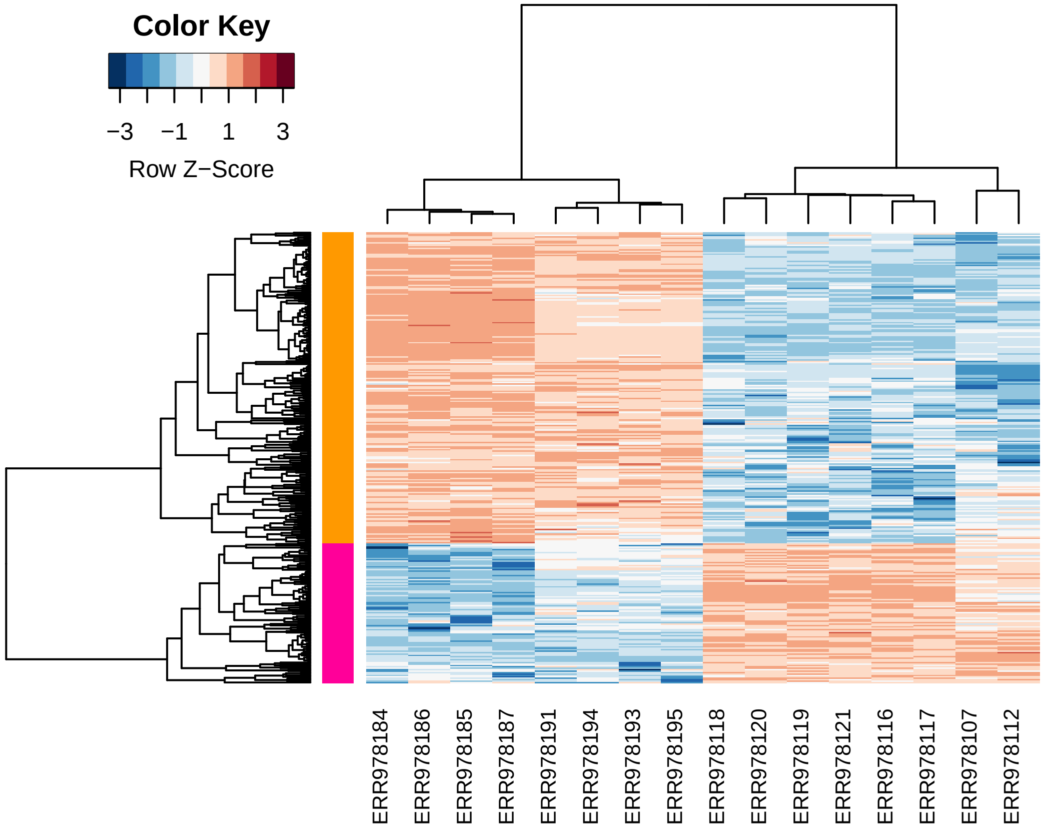


Figure 6: Heatmap of all differentially expressed genes, clustered by genes and samples. Scaled by rows to facilitate sample recognition and differentiation

GO enrichment

Gene Ontology enrichment for the 13291 genes differentially expressed.

Ontology of Up-Regulated Genes

Ontology of Down-Regulated Genes

GSEA

The 34550 gene sets in the Human Molecular Signatures Database **(MSigDB)** are divided into 9 major collections, and several subcollections. In this analysis, we will use the **C2** collection, which contains gene sets that represent canonical pathways, gene ontology, and other gene sets derived from knowledge in the

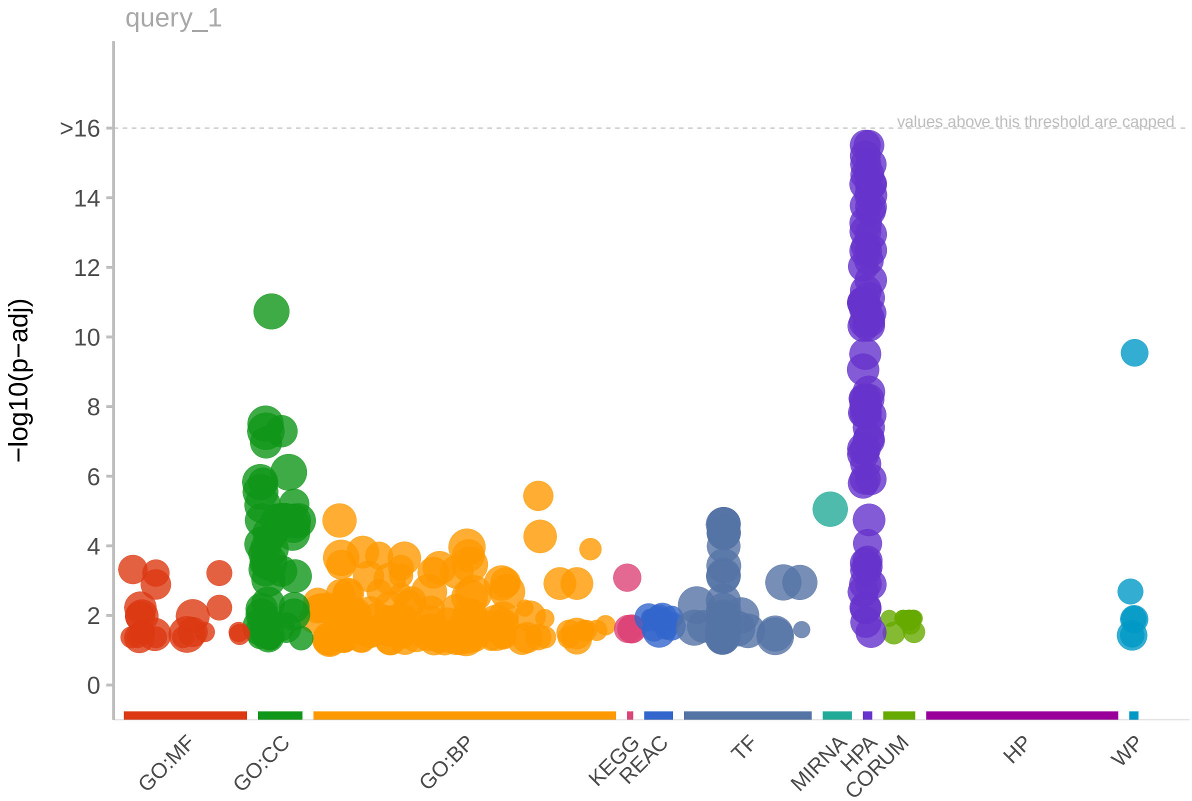


Figure 7: Manhattan plot of GO enrichment for upregulated genes

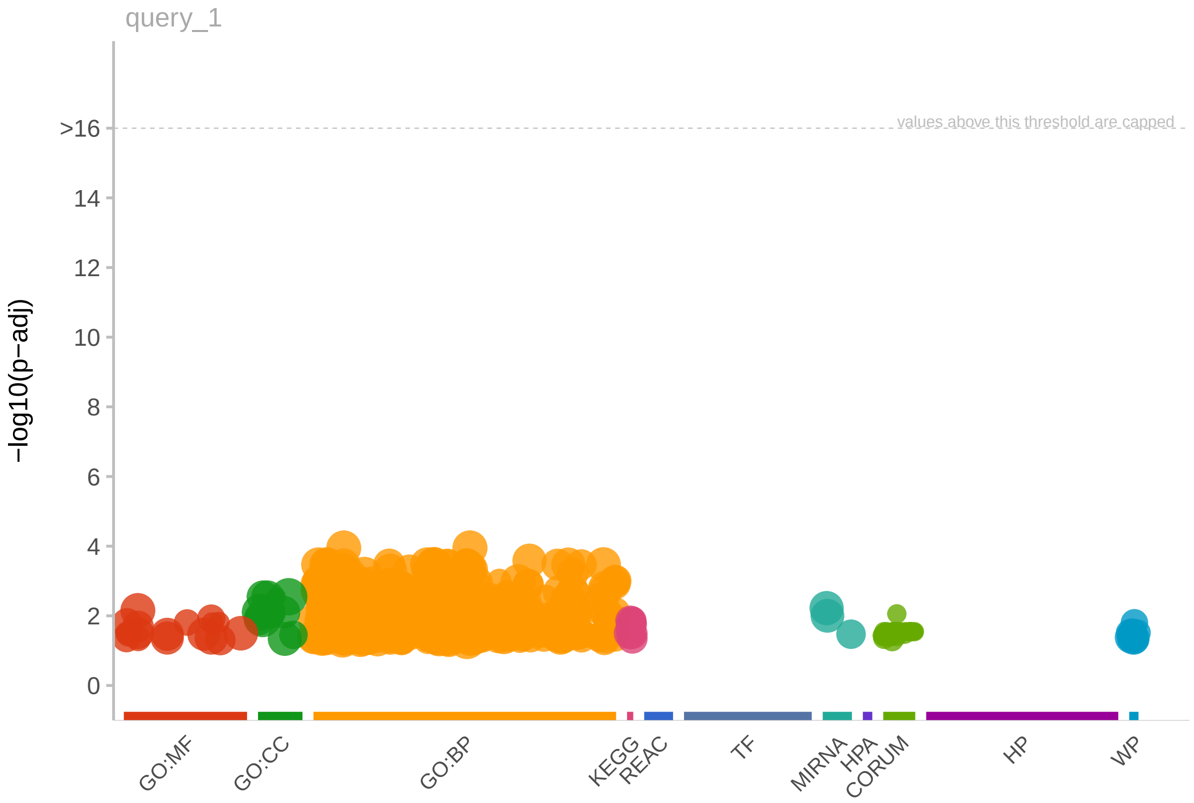


Figure 8: Manhattan plot of GO enrichment for downregulated genes

literature. The top 15 genes set for each conditions will have their own *Gene Set Enrichment Plot*. The table will contain all the results for each gene set. A sample plot is shown below.

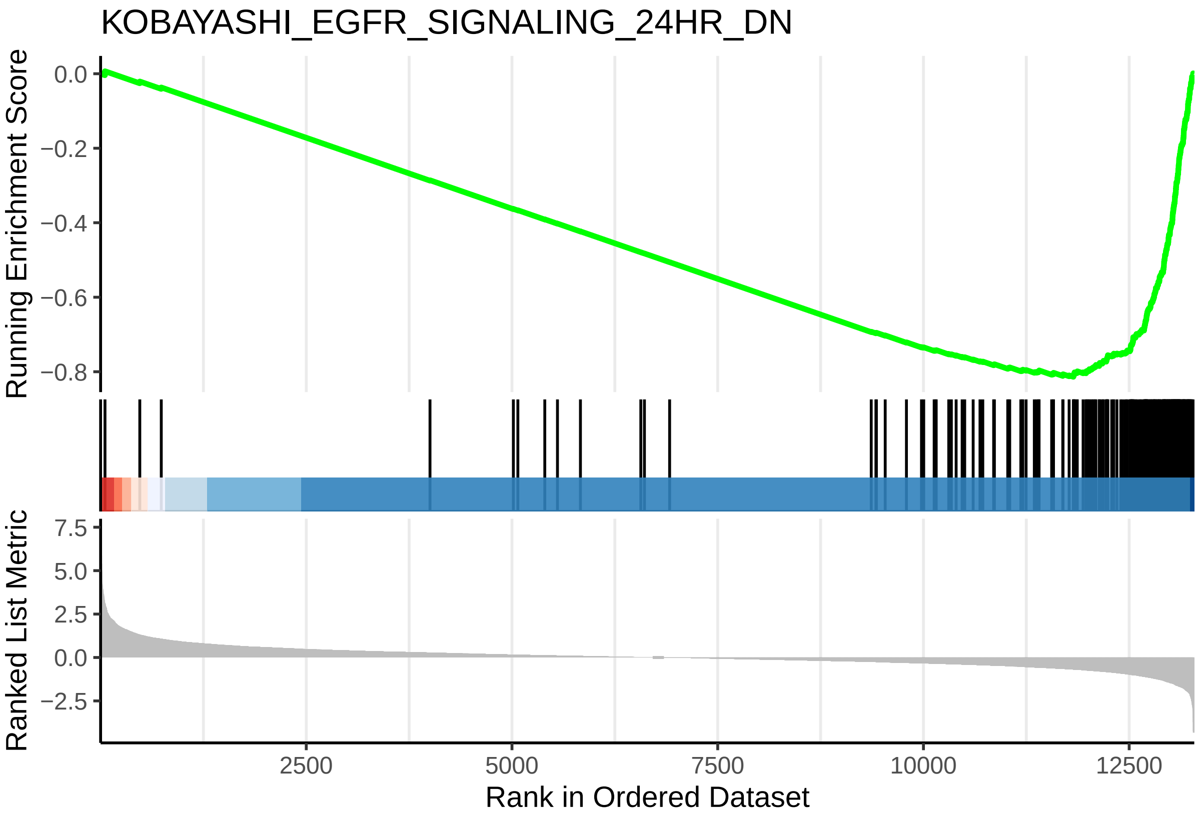


Figure 9: Sample of a Gene Set Enrichment Plot

Finnally, a *Bubble Plot* is produced for the top 20 gene sets. The size of the bubble is proportional to the number of genes in the gene set, the color is proportional to the **NES**, and the alpha is proportional to the *-log10(p.adjust)*. The plot is shown below. **NES** means normalized enrichment score, and it is the primary statistic for ranking genes in a GSEA analysis. It represents the degree to which a gene set is overrepresented at the top or bottom of a ranked list of genes. The score is normalized to account for differences in gene set size and in correlations between gene set members and the expression dataset. The **NES** score can be either >0, which means is enriched in the *sph* condition, or <0, which means it is enriched in the *adh* condition. The table provided shows the absolute score beacause it has already been classified in one the two conditions.

Session info

The output from running ‘sessionInfo’ is shown below and details all packages and version necessary to reproduce the results in this report.

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22621)
##
```

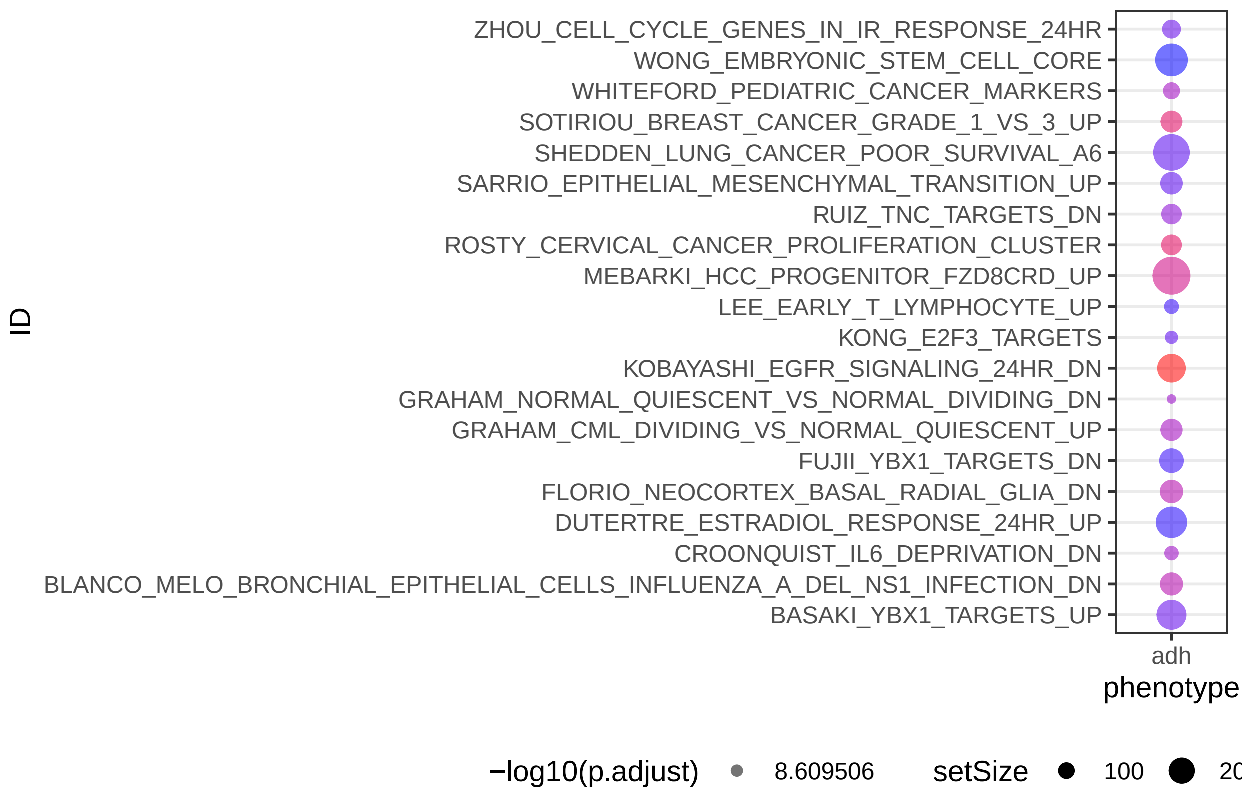


Figure 10: Bubble plot of the top 20 gene sets within the C2 Collection

```
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.utf8
## [2] LC_CTYPE=English_United Kingdom.utf8
## [3] LC_MONETARY=English_United Kingdom.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.utf8
##
## time zone: Europe/Madrid
## tzcode source: internal
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices datasets  utils     methods
## [8] base
##
## other attached packages:
##  [1] IsoformSwitchAnalyzeR_2.2.0 pfamAnalyzeR_1.2.0
##  [3] sva_3.50.0                  genefilter_1.84.0
##  [5] mgcv_1.8-42                 nlme_3.1-162
##  [7] satuRn_1.10.0               DEXSeq_1.48.0
##  [9] DESeq2_1.42.0               SummarizedExperiment_1.32.0
## [11] MatrixGenerics_1.14.0       BiocParallel_1.36.0
## [13] enrichplot_1.22.0           msigdbr_7.5.1
## [15] clusterProfiler_4.10.0      gprofiler2_0.2.2
## [17] GSVA_1.50.0                 GSEABase_1.64.0
## [19] graph_1.80.0                annotate_1.80.0
## [21] XML_3.99-0.16               heatmaply_1.5.0
## [23] viridis_0.6.4               viridisLite_0.4.2
## [25] RColorBrewer_1.1-3          gplots_3.1.3
## [27] plotly_4.10.3               gt_0.10.0
## [29] DT_0.31                     svglite_2.1.3
## [31] cowplot_1.1.2               matrixStats_1.2.0
## [33] edgeR_4.0.5                 limma_3.58.1
## [35] EnsDb.Hsapiens.v86_2.99.0   ensembldb_2.26.0
## [37] AnnotationFilter_1.26.0     GenomicFeatures_1.54.1
## [39] AnnotationDbi_1.64.1        Biobase_2.62.0
## [41] GenomicRanges_1.54.1        GenomeInfoDb_1.38.5
## [43] IRanges_2.36.0              S4Vectors_0.40.2
## [45] BiocGenerics_0.48.1         tximport_1.30.0
## [47] lubridate_1.9.3             forcats_1.0.0
## [49] stringr_1.5.1               dplyr_1.1.4
## [51] purrr_1.0.2                 readr_2.1.4
## [53] tidyr_1.3.0                 tibble_3.2.1
## [55] ggplot2_3.4.4               tidyverse_2.0.0
## [57] beepr_1.3                   knitr_1.45
## [59] tinytex_0.49                rmarkdown_2.25
##
## loaded via a namespace (and not attached):
##   [1] vroom_1.6.5                   progress_1.2.3
##   [3] locfdr_1.1-8                  Biostrings_2.70.1
##   [5] HDF5Array_1.30.0              vctrs_0.6.5
##   [7] digest_0.6.33                 png_0.1-8
##   [9] registry_0.5-1                ggrepel_0.9.4
##  [11] renv_1.0.3                    MASS_7.3-60
##  [13] reshape2_1.4.4                foreach_1.5.2
##  [15] httpuv_1.6.13                 qvalue_2.34.0
##  [17] withr_2.5.2                   xfun_0.41
##  [19] ggfun_0.1.3                   ellipsis_0.3.2
##  [21] survival_3.5-5                memoise_2.0.1
##  [23] tximeta_1.20.1                gson_0.1.0
##  [25] systemfonts_1.0.5             ragg_1.2.7
##  [27] tidytree_0.4.6                gtools_3.9.5
##  [29] pbapply_1.7-2                 prettyunits_1.2.0
##  [31] KEGGREST_1.42.0               promises_1.2.1
##  [33] httr_1.4.7                    restfulr_0.0.15
##  [35] rhdf5filters_1.14.1           rhdf5_2.46.1
##  [37] rstudioapi_0.15.0             generics_0.1.3
##  [39] DOSE_3.28.2                   babelgene_22.9
##  [41] curl_5.2.0                    zlibbioc_1.48.0
##  [43] ScaledMatrix_1.10.0           ggraph_2.1.0
##  [45] polyclip_1.10-6               ca_0.71.1
##  [47] GenomeInfoDbData_1.2.11       SparseArray_1.2.3
##  [49] interactiveDisplayBase_1.40.0 xtable_1.8-4
##  [51] evaluate_0.23                 S4Arrays_1.2.0
##  [53] BiocFileCache_2.10.1          hms_1.1.3
##  [55] irlba_2.3.5.1                 colorspace_2.1-0
##  [57] filelock_1.0.3                VennDiagram_1.7.3
##  [59] magrittr_2.0.3                later_1.3.2
##  [61] ggtree_3.10.0                 lattice_0.21-8
##  [63] shadowtext_0.1.2              pillar_1.9.0
##  [65] iterators_1.0.14              caTools_1.18.2
##  [67] compiler_4.3.1                beachmat_2.18.0
##  [69] stringi_1.8.3                 TSP_1.2-4
##  [71] dendextend_1.17.1             GenomicAlignments_1.38.0
##  [73] plyr_1.8.9                    crayon_1.5.2
##  [75] abind_1.4-5                   BiocIO_1.12.0
##  [77] gridGraphics_0.5-1            locfit_1.5-9.8
##  [79] graphlayouts_1.0.2            bit_4.0.5
##  [81] fastmatch_1.1-4               codetools_0.2-19
##  [83] textshaping_0.3.7             BiocSingular_1.18.0
##  [85] mime_0.12                     splines_4.3.1
##  [87] Rcpp_1.0.11                   dbplyr_2.4.0
##  [89] sparseMatrixStats_1.14.0      HDO.db_0.99.1
##  [91] blob_1.2.4                    utf8_1.2.4
##  [93] BiocVersion_3.18.1            fs_1.6.3
##  [95] DelayedMatrixStats_1.24.0     ggplotify_0.1.2
##  [97] Matrix_1.5-4.1                statmod_1.5.0
##  [99] tzdb_0.4.0                    tweenr_2.0.2
## [101] pkgconfig_2.0.3               tools_4.3.1
## [103] cachem_1.0.8                  RSQLite_2.3.4
## [105] DBI_1.2.0                     fastmap_1.1.1
## [107] scales_1.3.0                  grid_4.3.1
## [109] audio_0.1-11                  geneplotter_1.80.0
## [111] Rsamtools_2.18.0              AnnotationHub_3.10.0
## [113] patchwork_1.1.3               BiocManager_1.30.22
## [115] snow_0.4-4                    farver_2.1.1
## [117] tidygraph_1.3.0               scatterpie_0.2.1
## [119] yaml_2.3.8                    rtracklayer_1.62.0
## [121] cli_3.6.2                     webshot_0.5.5
## [123] lifecycle_1.0.4               lambda.r_1.2.4
## [125] timechange_0.2.0              gtable_0.3.4
## [127] rjson_0.2.21                  parallel_4.3.1
## [129] ape_5.7-1                     jsonlite_1.8.8
## [131] seriation_1.5.4               bitops_1.0-7
## [133] bit64_4.0.5                   assertthat_0.2.1
## [135] yulab.utils_0.1.2             futile.options_1.0.1
## [137] highr_0.10                    GOSemSim_2.28.0
## [139] lazyeval_0.2.2                shiny_1.8.0
## [141] htmltools_0.5.7               GO.db_3.18.0
## [143] rappdirs_0.3.3                formatR_1.14
## [145] glue_1.6.2                    XVector_0.42.0
## [147] RCurl_1.98-1.13               treeio_1.26.0
## [149] BSgenome_1.70.1               futile.logger_1.4.3
## [151] gridExtra_2.3                 boot_1.3-28.1
## [153] igraph_1.6.0                  R6_2.5.1
## [155] SingleCellExperiment_1.24.0   labeling_0.4.3
## [157] Rhdf5lib_1.24.1               aplot_0.2.2
## [159] DelayedArray_0.28.0           tidyselect_1.2.0
## [161] ProtGenerics_1.34.0           ggforce_0.4.1
## [163] xml2_1.3.6                    rsvd_1.0.5
## [165] munsell_0.5.0                 KernSmooth_2.23-21
## [167] data.table_1.14.10            htmlwidgets_1.6.4
## [169] fgsea_1.28.0                  hwriter_1.3.2.1
## [171] biomaRt_2.58.0                rlang_1.1.2
## [173] fansi_1.0.6
```