

SWATH2stats example script

Example R code showing the usage of the SWATH2stats package. The data processed is the publicly available dataset of *S.pyogenes* (Röst et al. 2014) (<http://www.peptideatlas.org/PASS/PASS00289>). The results file 'rawOpenSwathResults_1pcnt_only.tsv' can be found on PeptideAtlas (<ftp://PASS00289@ftp.peptideatlas.org/./Spyogenes/results/>). This is a R Markdown file, showing the result of processing this data. The lines shaded in grey represent the R code executed during this analysis.

The SWATH2stats package can be directly installed from Bioconductor using the commands below (<http://bioconductor.org/packages/devel/bioc/html/SWATH2stats.html>).

```
if (!require("BiocManager"))
  install.packages("BiocManager")
BiocManager::install("SWATH2stats")
```

Part 1: Loading and annotation

Load the SWATH-MS example data from the package, this is a reduced file in order to limit the file size of the package.

```
library(SWATH2stats)
```

```
## Warning: replacing previous import 'utils::findMatches' by
## 'S4Vectors::findMatches' when loading 'AnnotationDbi'
```

```
library(data.table)
data('Spyogenes', package = 'SWATH2stats')
```

Alternatively the original file downloaded from the Peptide Atlas can be loaded from the working directory.

```
data <- data.frame(fread('rawOpenSwathResults_1pcnt_only.tsv', sep='\t', header=TRUE))
```

Extract the study design information from the file names. Alternatively, the study design table can be provided as an external table.

```
Study_design <- data.frame(FileName = unique(data$align_origfilename))
Study_design$FileName <- gsub(".*strep_align/(.*)_all_peakgroups.*", "\\1", Study_design$FileName)
Study_design$Condition <- gsub("(Strep.*)_Repl.*", "\\1", Study_design$FileName)
Study_design$BioReplicate <- gsub(".*Repl([[:digit:]]).*", "\\1", Study_design$FileName)
Study_design$Run <- seq_len(nrow(Study_design))
head(Study_design)
```

```
##           Filename Condition BioReplicate Run
## 1 Strep0_Repl1_R02/split_hroest_K120808   Strep0         1  1
## 2 Strep0_Repl2_R02/split_hroest_K120808   Strep0         2  2
## 3 Strep10_Repl1_R02/split_hroest_K120808 Strep10         1  3
## 4 Strep10_Repl2_R02/split_hroest_K120808 Strep10         2  4
```

The SWATH-MS data is annotated using the study design table.

```
data.annotated <- sample_annotation(data, Study_design, column_file = "align_origfilename")
```

Remove the decoy peptides for a subsequent inspection of the data.

```
data.annotated.nodecoy <- subset(data.annotated, decoy==FALSE)
```

Part 2: Analyze correlation, variation and signal

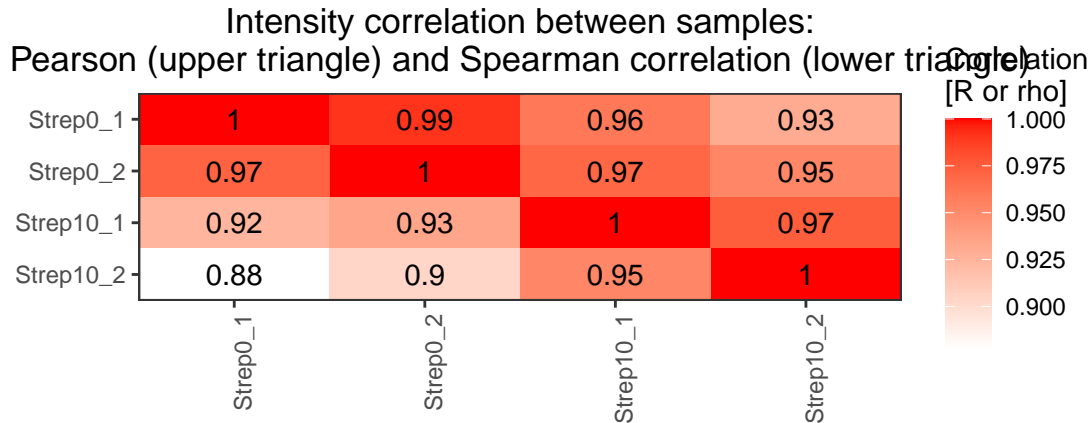
Count the different analytes for the different injections.

```
count_analytes(data.annotated.nodcoy)
```

```
##      run_id transition_group_id FullPeptideName ProteinName
## 1 Strep0_1_1          10229           8377       1031
## 2 Strep0_2_2           9716           7970       1003
## 3 Strep10_1_3          8692           7138        943
## 4 Strep10_2_4          8424           6941        910
```

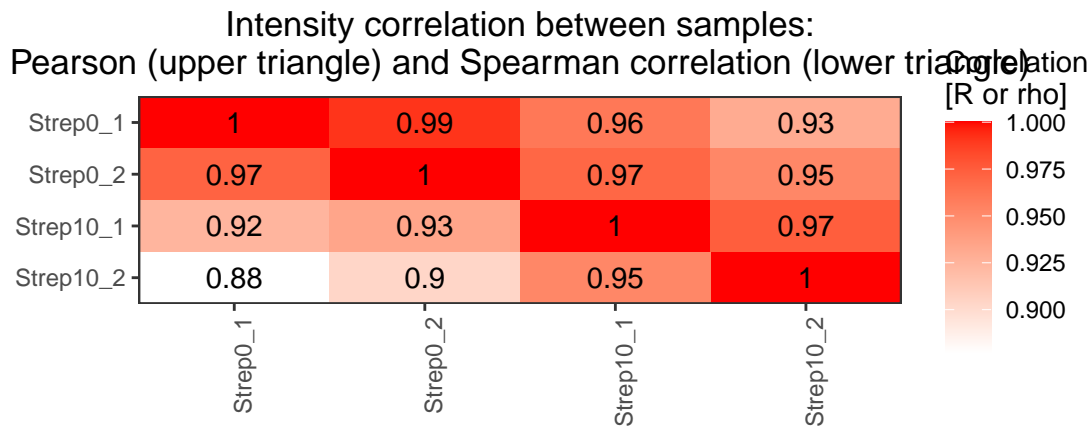
Plot the correlation of the signal intensity.

```
correlation <- plot_correlation_between_samples(data.annotated.nodcoy, column.values = 'Intensity')
```



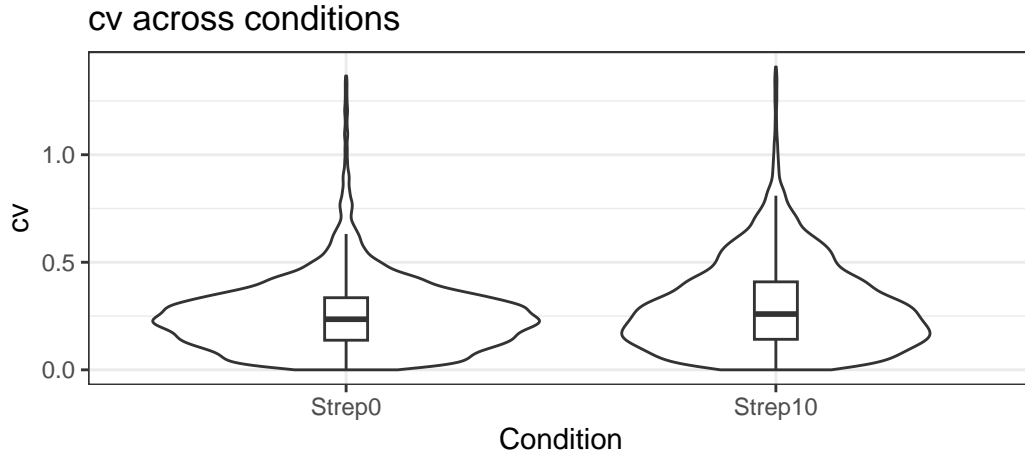
Plot the correlation of the delta_rt, which is the deviation of the retention time from the expected retention time.

```
correlation <- plot_correlation_between_samples(data.annotated.nodcoy, column.values = 'delta_rt')
```



Plot the variation of the signal across replicates.

```
variation <- plot_variation(data.annotated.nodcoy)
```

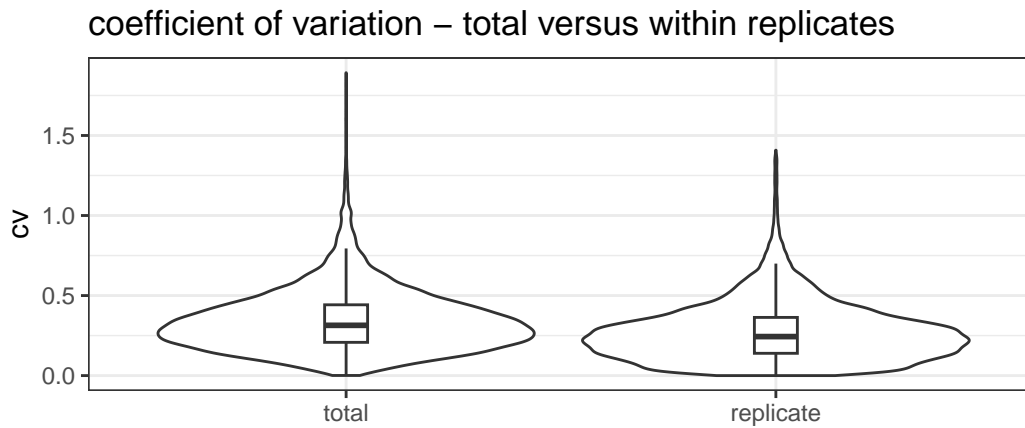


```
variation[[2]]
```

```
## Condition mode_cv mean_cv median_cv
## 1 Strep0 0.2280372 0.2545450 0.2351859
## 2 Strep10 0.1706934 0.2947144 0.2592725
```

Plot the total variation versus variation within replicates.

```
variation_total <- plot_variation_vs_total(data.annotated.nodcoy)
```



```
variation_total[[2]]
```

```
## scope mode_cv mean_cv median_cv
## 1 replicate 0.2209867 0.2728681 0.2438041
## 2 total 0.2655678 0.3439050 0.3139993
```

Calculate the summed signal per peptide and protein across samples.

```
peptide_signal <- write_matrix_peptides(data.annotated.nodcoy)
protein_signal <- write_matrix_proteins(data.annotated.nodcoy)
head(protein_signal)
```

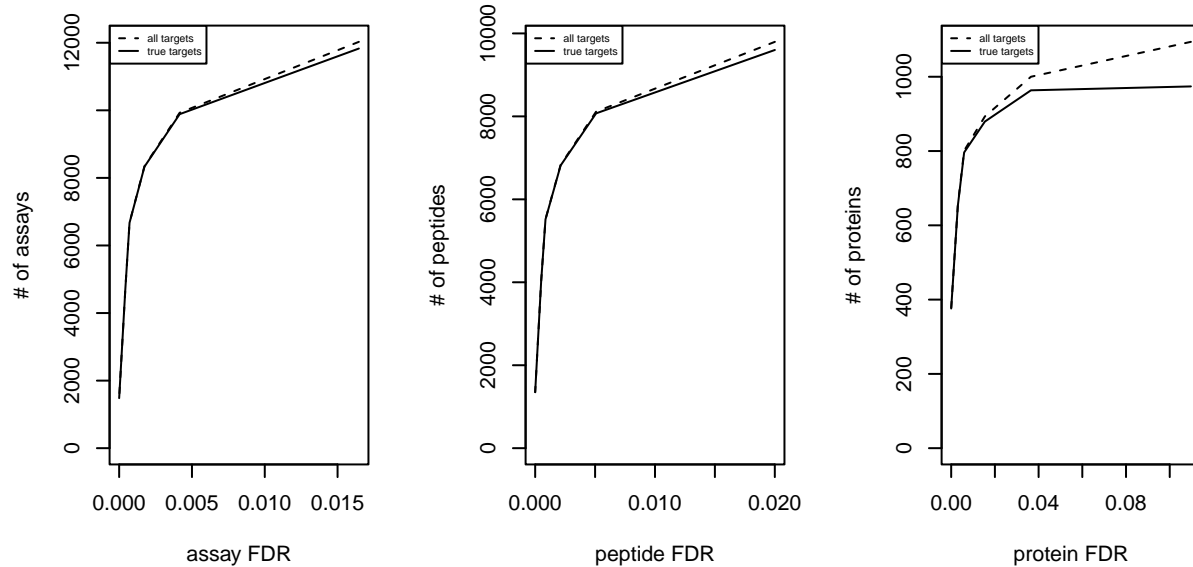
```
## ProteinName Strep0_1_1 Strep0_2_2 Strep10_1_3 Strep10_2_4
## 1 Spyo_Exp3652_DDB_SeqID_1571119 265206 163326 51831 45021
## 2 Spyo_Exp3652_DDB_SeqID_1579753 185725 150672 21483 144314
```

| | | | | | |
|------|---------------------------------|--------|--------|--------|--------|
| ## 3 | Spyo_Exp3652_DDB_SeqID_1631459 | 176686 | 132415 | 42165 | 32735 |
| ## 4 | Spyo_Exp3652_DDB_SeqID_1640263 | 3310 | 6617 | 98550 | 45169 |
| ## 5 | Spyo_Exp3652_DDB_SeqID_1709452 | 852502 | 747772 | 503581 | 504761 |
| ## 6 | Spyo_Exp3652_DDB_SeqID_17244480 | 17506 | 29578 | 7607 | 2482 |

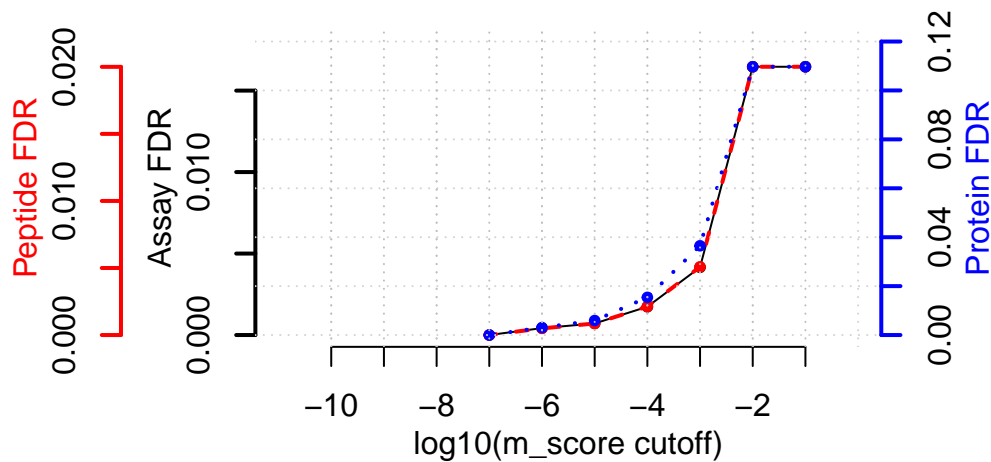
Part 3: FDR estimation

Estimate the overall FDR across runs using a target decoy strategy.

```
par(mfrow = c(1, 3))
fdr_target_decoy <- assess_fdr_overall(data.annotated, n_range = 10,
                                       FFT = 0.25, output = 'Rconsole')
```



Global m-score cutoff connectivity to FDR quality



According to this FDR estimation one would need to filter the data with a lower mscore threshold to reach an overall protein FDR of 5%.

```
mscore4protfdr(data, FFT = 0.25, fdr_target = 0.05)
```

```
## Target protein FDR:0.05
## Required overall m-score cutoff:0.0017783
## achieving protein FDR =0.0488
## [1] 0.001778279
```

Part 4: Filtering

Filter data for values that pass the 0.001 mscore criteria in at least two replicates of one condition.

```
data.filtered <- filter_mscore_condition(data.annotated, 0.001, n_replica = 2)
```

```
## Fraction of peptides selected: 0.67
```

```
## Dimension difference: 7226, 0
```

Select only the 10 peptides showing strongest signal per protein.

```
data.filtered2 <- filter_on_max_peptides(data.filtered, n_peptides = 10)
```

```
## Before filtering:
```

```
##   Number of proteins: 884
```

```
##   Number of peptides: 6594
```

```
##
```

```
## Percentage of peptides removed: 29.6%
```

```
##
```

```
## After filtering:
```

```
##   Number of proteins: 884
```

```
##   Number of peptides: 4642
```

Filter for proteins that are supported by at least two peptides.

```
data.filtered3 <- filter_on_min_peptides(data.filtered2, n_peptides = 2)
```

```
## Before filtering:
##   Number of proteins: 884
##   Number of peptides: 4642
##
## Percentage of peptides removed: 3.6%
##
## After filtering:
##   Number of proteins: 717
##   Number of peptides: 4475
```

Part 5: Conversion

Convert the data into a transition-level format (one row per transition measured).

```
data.transition <- disaggregate(data.filtered3)
```

```
## The library contains 6 transitions per precursor.
##
## The data table was transformed into a table containing one row per transition.
```

Convert the data into the format required by MSstats.

```
MSstats.input <- convert4MSstats(data.transition)
```

```
## One or several columns required by MSstats were not in the data.
##           The columns were created and filled with NAs.
## Missing columns: ProductCharge, IsotopeLabelType
## IsotopeLabelType was filled with light.
## Warning in convert4MSstats(data.transition): Intensity values that were 0, were
## replaced by NA
```

```
head(MSstats.input)
```

```
##           ProteinName      PeptideSequence PrecursorCharge
## 1 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR      3
## 2 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR      3
## 3 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR      3
## 4 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR      3
## 5 Spyo_Exp3652_DDB_SeqID_1571119      AHIAYLPSDGR      2
## 6 Spyo_Exp3652_DDB_SeqID_1571119      AHIAYLPSDGR      2
##           FragmentIon ProductCharge IsotopeLabelType Intensity
## 1 105801_AEAAIYQFLEAIGENPNR/3_y6      NA      light      4752
## 2 105801_AEAAIYQFLEAIGENPNR/3_y6      NA      light      6144
## 3 105801_AEAAIYQFLEAIGENPNR/3_y6      NA      light      3722
## 4 105801_AEAAIYQFLEAIGENPNR/3_y6      NA      light      6624
## 5      118149_AHIAYLPSDGR/2_y8      NA      light      4036
## 6      118149_AHIAYLPSDGR/2_y8      NA      light      1642
## BioReplicate Condition Run
## 1           2      Strep0      2
## 2           1      Strep10     3
## 3           2      Strep10     4
## 4           1      Strep0      1
```



```
## 5          1      Strep0    1
## 6          1      Strep10   3
```

Convert the data into the format required by mapDIA.

```
mapDIA.input <- convert4mapDIA(data.transition)
head(mapDIA.input)
```

```
##          ProteinName      PeptideSequence
## 1 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR
## 2 Spyo_Exp3652_DDB_SeqID_1571119      AHIAYLPSDGR
## 3 Spyo_Exp3652_DDB_SeqID_1571119      EEFTAVFK
## 4 Spyo_Exp3652_DDB_SeqID_1571119 EKAEAAIYQFLEAIGENPNR
## 5 Spyo_Exp3652_DDB_SeqID_1571119      EQHEDVVIVK
## 6 Spyo_Exp3652_DDB_SeqID_1571119      LTSQIADALVEALNPK
##          FragmentIon Strep0_1 Strep0_2 Strep10_1 Strep10_2
## 1 105801_AEAAIYQFLEAIGENPNR/3_y6 6624 4752 6144 3722
## 2 118149_AHIAYLPSDGR/2_y8 4036 2405 1642 720
## 3 35179_EEFTAVFK/2_y5 2307 1541 1561 NaN
## 4 28903_EKAEAAIYQFLEAIGENPNR/3_y6 3410 2185 NaN 1984
## 5 73581_EQHEDVVIVK/2_b6 2423 1343 NaN NaN
## 6 115497_LTSQIADALVEALNPK/2_y11 6553 6349 NaN NaN
```

Convert the data into the format required by aLFQ.

```
aLFQ.input <- convert4aLFQ(data.transition)
```

```
## Checking the integrity of the transitions takes a lot of time.
##          To speed up consider changing the option.
```

```
head(aLFQ.input)
```

```
##          run_id          protein_id          peptide_id
## 1 Strep0_2_2 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR
## 2 Strep10_1_3 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR
## 3 Strep10_2_4 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR
## 4 Strep0_1_1 Spyo_Exp3652_DDB_SeqID_1571119 AEAAIYQFLEAIGENPNR
## 5 Strep0_1_1 Spyo_Exp3652_DDB_SeqID_1571119      AHIAYLPSDGR
## 6 Strep10_1_3 Spyo_Exp3652_DDB_SeqID_1571119      AHIAYLPSDGR
##          transition_id  peptide_sequence
## 1 AEAAIYQFLEAIGENPNR 105801_AEAAIYQFLEAIGENPNR/3_y6 AEAAIYQFLEAIGENPNR
## 2 AEAAIYQFLEAIGENPNR 105801_AEAAIYQFLEAIGENPNR/3_y6 AEAAIYQFLEAIGENPNR
## 3 AEAAIYQFLEAIGENPNR 105801_AEAAIYQFLEAIGENPNR/3_y6 AEAAIYQFLEAIGENPNR
## 4 AEAAIYQFLEAIGENPNR 105801_AEAAIYQFLEAIGENPNR/3_y6 AEAAIYQFLEAIGENPNR
## 5      AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8      AHIAYLPSDGR
## 6      AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8      AHIAYLPSDGR
## precursor_charge transition_intensity concentration
## 1          3          4752          ?
## 2          3          6144          ?
## 3          3          3722          ?
## 4          3          6624          ?
## 5          2          4036          ?
## 6          2          1642          ?
```

Session info on the R version and packages used.

```
sessionInfo()
```

```

## R version 4.3.0 RC (2023-04-13 r84269)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.2 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.17-bioc/R/lib/libRblas.so
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
## [3] LC_TIME=en_GB 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
##
## time zone: America/New_York
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats graphics grDevices utils datasets methods base
##
## other attached packages:
## [1] data.table_1.14.8 SWATH2stats_1.30.0
##
## loaded via a namespace (and not attached):
## [1] KEGGREST_1.40.0 gtable_0.3.3 xfun_0.39
## [4] ggplot2_3.4.2 Biobase_2.60.0 vctrs_0.6.2
## [7] tools_4.3.0 bitops_1.0-7 generics_0.1.3
## [10] stats4_4.3.0 curl_5.0.0 tibble_3.2.1
## [13] fansi_1.0.4 AnnotationDbi_1.62.0 RSQLite_2.3.1
## [16] highr_0.10 blob_1.2.4 pkgconfig_2.0.3
## [19] dbplyr_2.3.2 S4Vectors_0.38.0 lifecycle_1.0.3
## [22] GenomeInfoDbData_1.2.10 farver_2.1.1 compiler_4.3.0
## [25] stringr_1.5.0 Biostrings_2.68.0 progress_1.2.2
## [28] munsell_0.5.0 GenomeInfoDb_1.36.0 htmltools_0.5.5
## [31] RCurl_1.98-1.12 yaml_2.3.7 pillar_1.9.0
## [34] crayon_1.5.2 cachem_1.0.7 tidyselect_1.2.0
## [37] digest_0.6.31 stringi_1.7.12 reshape2_1.4.4
## [40] dplyr_1.1.2 labeling_0.4.2 biomaRt_2.56.0
## [43] fastmap_1.1.1 grid_4.3.0 colorspace_2.1-0
## [46] cli_3.6.1 magrittr_2.0.3 XML_3.99-0.14
## [49] utf8_1.2.3 withr_2.5.0 prettyunits_1.1.1
## [52] filelock_1.0.2 scales_1.2.1 rappdirs_0.3.3
## [55] bit64_4.0.5 rmarkdown_2.21 XVector_0.40.0
## [58] httr_1.4.5 bit_4.0.5 png_0.1-8
## [61] hms_1.1.3 memoise_2.0.1 evaluate_0.20
## [64] knitr_1.42 IRanges_2.34.0 BiocFileCache_2.8.0
## [67] rlang_1.1.0 Rcpp_1.0.10 glue_1.6.2
## [70] DBI_1.1.3 formatR_1.14 xml2_1.3.3
## [73] BiocGenerics_0.46.0 plyr_1.8.8 R6_2.5.1
## [76] zlibbioc_1.46.0

```