SWATH2stats example script

Peter Blattmann

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 stable release package SWATH2stats can be directly installed from Bioconductor using the commands below. This file here was generated using the current development release SWATH2stats v.1.1.14 that can be downloaded from http://bioconductor.org/packages/devel/bioc/html/SWATH2stats.html.

```
## try http:// if https:// URLs are not supported
source('https://bioconductor.org/biocLite.R')
biocLite('SWATH2stats')

## Conversely, install from github
devtools::install_github("abelew/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)
library(data.table)
data('Spyogenes', package='SWATH2stats')
data <- Spyogenes
```

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(1: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,
                                    data_file_column="align_origfilename",
                                    check_files=FALSE)
```

```
## Not checking that the files are identical between the annotation and data.
```

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

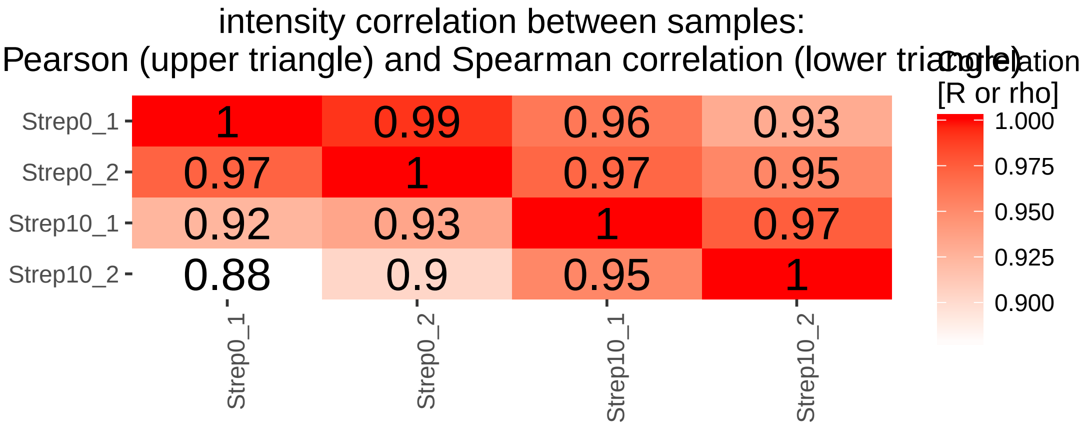
Count the different analytes for the different injections.

```
count_analytes(data.annotated.nodecoy)
```

```
##        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
```

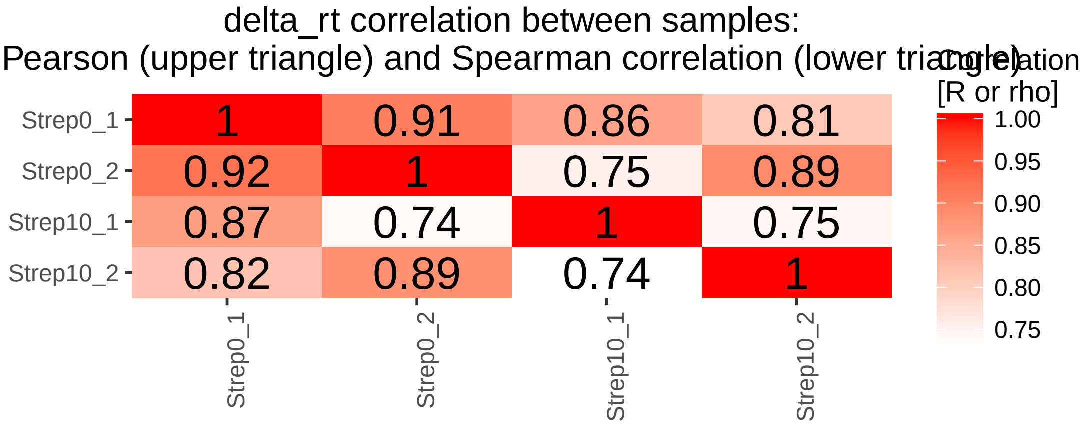
Plot the correlation of the signal intensity.

```
correlation <- plot_correlation_between_samples(data.annotated.nodecoy,
                                                column.values="intensity")
```



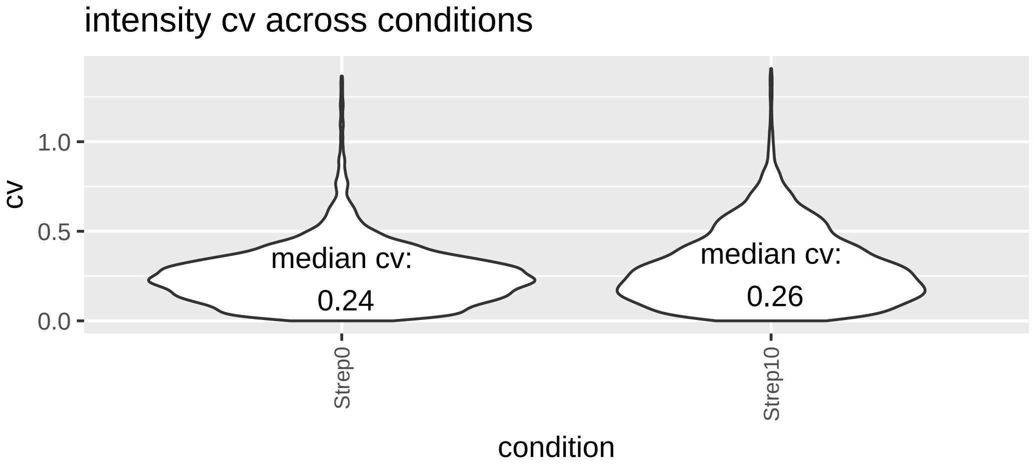
Plot the correlation of the delta_rt, which is the deviation of the retention time from the expected retention time.

```
correlation <- plot_correlation_between_samples(data.annotated.nodecoy,
                                                column.values="delta_rt")
```



Plot the variation of the signal across replicates.

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

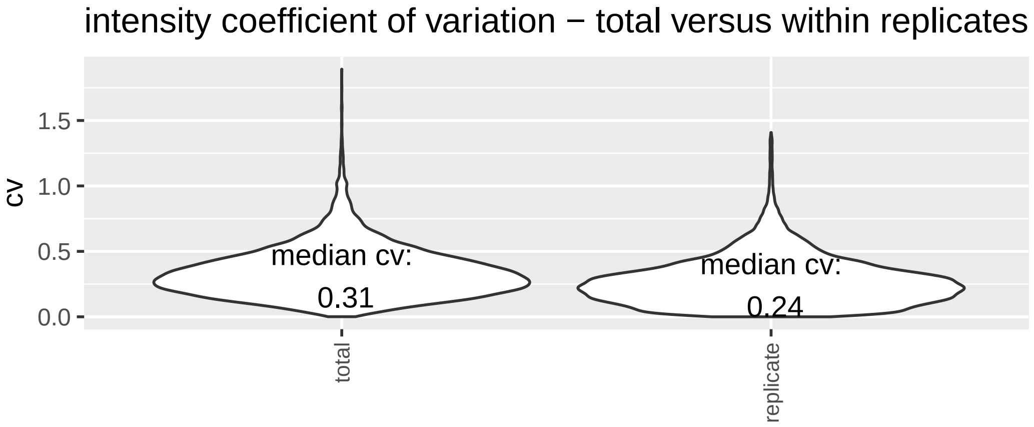


```
head(variation)
```

```
##                               transition_group_id condition      1      2
## 1                     1_FSWISTGGGASMELLEGK/2_run0    Strep0 135206 119997
## 2                     1_FSWISTGGGASMELLEGK/2_run0   Strep10 147766 110436
## 3               1000_TGIFSQDDENALENSIGFSSK/3_run0    Strep0   6946   4161
## 4                          10000_DIVEAVIPR/2_run0    Strep0 163405  67537
## 5                          10000_DIVEAVIPR/2_run0   Strep10  53345  20963
## 6 10001_SSGYNLGGEQSGHVIIMDYNTTGDGQLTAIQLAK/3_run0    Strep0  10798  10876
##            cv
## 1 0.084281039
## 2 0.204462368
## 3 0.354603833
## 4 0.587064396
## 5 0.616287124
## 6 0.005089446
```

Plot the total variation versus variation within replicates.

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



```
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.

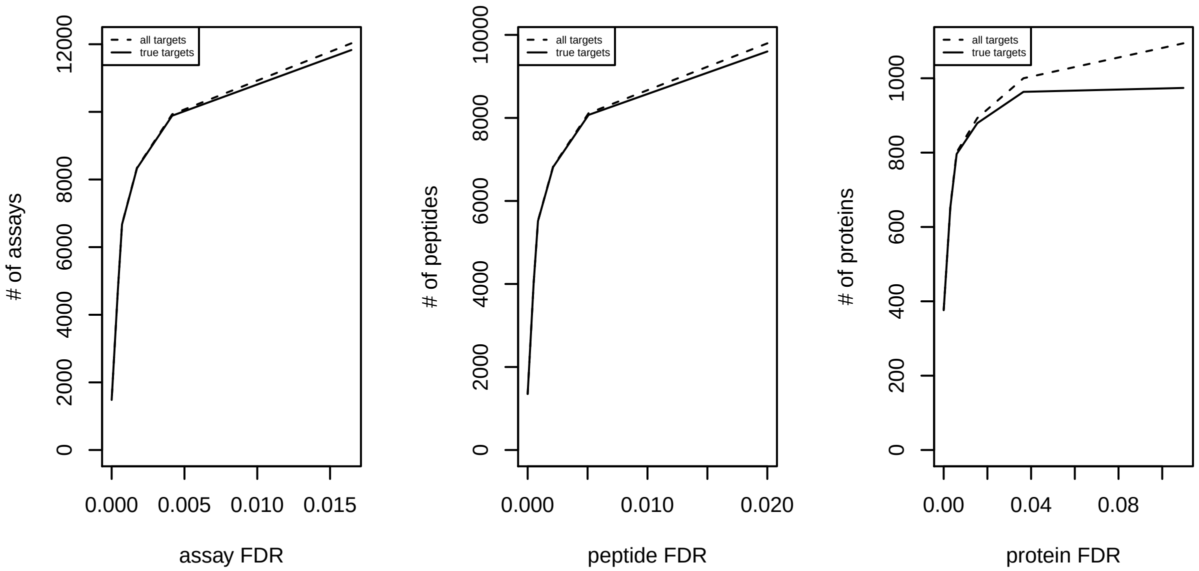
```
peptide_signal <- write_matrix_peptides(data.annotated.nodecoy)
protein_signal <- write_matrix_proteins(data.annotated.nodecoy)
head(protein_signal)
```

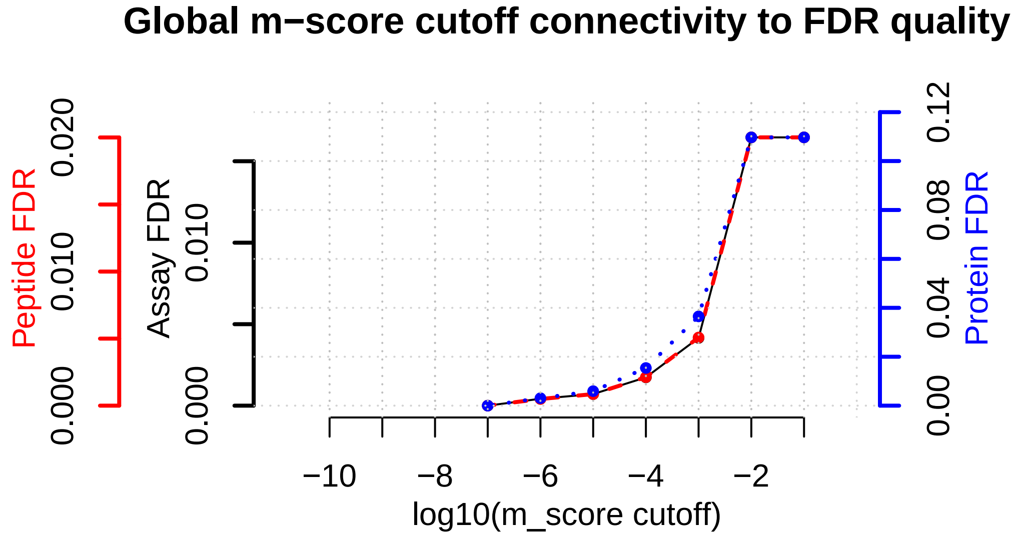
```
##                        proteinname Strep0_1_1 Strep0_2_2 Strep10_1_3
## 1  Spyo_Exp3652_DDB_SeqID_1571119     265206     163326       51831
## 2  Spyo_Exp3652_DDB_SeqID_1579753     185725     150672       21483
## 3  Spyo_Exp3652_DDB_SeqID_1631459     176686     132415       42165
## 4  Spyo_Exp3652_DDB_SeqID_1640263       3310       6617       98550
## 5  Spyo_Exp3652_DDB_SeqID_1709452     852502     747772      503581
## 6 Spyo_Exp3652_DDB_SeqID_17244480      17506      29578        7607
##   Strep10_2_4
## 1       45021
## 2      144314
## 3       32735
## 4       45169
## 5      504761
## 6        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")
```





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: NA
## achieving protein FDR: NA

## [1] NA
```

Part 4: Filtering

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

```r
data.filtered <- filter_mscore_condition(data.annotated, 0.001, n.replica = 2)
```

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

```
## Original dimension: 37061, new dimension: 29835, difference: 7226.
```

Select only the 10 peptides showing strongest signal per protein.

```r
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: 830
##   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: 830
##   Number of peptides: 4642
##
## Percentage of peptides removed: 0.3%
##
## After filtering:
##   Number of proteins: 716
##   Number of peptides: 4628
```

Part 5: Conversion

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

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

```
## The library contains between 4 and 6 transitions per precursor.
## The data table was transformed into a table containing one row per transition.
```

```
## 4 row(s) was(were) removed because they did not contain data due to different number of transitions
```

Convert the data into the format required by MSstats.

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

```
## One or several columns required by MSstats were not in the data. The columns were created and filled
## Missing columns: productcharge, isotopelabeltype
```

```
## isotopelabeltype was filled with light.
```

```
## Warning in convert_MSstats(data.transition): Intensity values which were 0
## have been replaced by NA.
```

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

```
##                        proteinname   peptidesequence precursorcharge
## 1 Spyo_Exp3652_DDB_SeqID_1571119      SLPEEDLDKNEK               2
## 2 Spyo_Exp3652_DDB_SeqID_1571119      SLPEEDLDKNEK               2
## 3 Spyo_Exp3652_DDB_SeqID_1571119     TIFDDEPISEETK               2
## 4 Spyo_Exp3652_DDB_SeqID_1571119     TIFDDEPISEETK               2
## 5 Spyo_Exp3652_DDB_SeqID_1571119 LSLPSQEPLLAAFHGEK               3
## 6 Spyo_Exp3652_DDB_SeqID_1571119 LSLPSQEPLLAAFHGEK               3
##                        fragmention productcharge isotopelabeltype intensity
## 1         118149_AHIAYLPSDGR/2_y8            NA            light      4036
## 2         118149_AHIAYLPSDGR/2_y8            NA            light      1642
## 3         118149_AHIAYLPSDGR/2_y8            NA            light      2405
## 4         118149_AHIAYLPSDGR/2_y8            NA            light       720
## 5 28903_EKAEAAIYQFLEAIGENPNR/3_y6            NA            light      3410
## 6 28903_EKAEAAIYQFLEAIGENPNR/3_y6            NA            light      1984
##   bioreplicate condition run
## 1            1    Strep0   1
## 2            1   Strep10   3
## 3            2    Strep0   2
## 4            2   Strep10   4
## 5            1    Strep0   1
## 6            2   Strep10   4
```

Convert the data into the format required by mapDIA.

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

```
##                      proteinname            peptidesequence
## 1 Spyo_Exp3652_DDB_SeqID_1571119               SLPEEDLDKNEK
## 2 Spyo_Exp3652_DDB_SeqID_1571119              TIFDDEPISEETK
## 3 Spyo_Exp3652_DDB_SeqID_1571119          LSLPSQEPLLAAFHGEK
## 4 Spyo_Exp3652_DDB_SeqID_1571119                 SLETEGKVDK
## 5 Spyo_Exp3652_DDB_SeqID_1579753 TLIDAYEAFC[160]PLDLSMEGDVK
## 6 Spyo_Exp3652_DDB_SeqID_1579753        SDTAGTIVSLNTDLPNQSK
##                       fragmention Strep0_1 Strep0_2 Strep10_1 Strep10_2
## 1         118149_AHIAYLPSDGR/2_y8     4036      NaN      1642       NaN
## 2         118149_AHIAYLPSDGR/2_y8      NaN     2405       NaN       720
## 3 28903_EKAEAAIYQFLEAIGENPNR/3_y6     3410      NaN       NaN      1984
## 4 28903_EKAEAAIYQFLEAIGENPNR/3_y6      NaN     2185       NaN       NaN
## 5  97491_LALAPNTPGQIVALELGEK/3_y7     5681     4099      3060      2301
## 6     56597_LNDGAFLALDGSAQYK/2_y9     3349     2552       NaN       860
```

Convert the data into the format required by aLFQ.

```
aLFQ.input <- convert_aLFQ(data.transition)
head(aLFQ.input)
```

```
##        run_id                     protein_id        peptide_id
## 1  Strep0_1_1 Spyo_Exp3652_DDB_SeqID_1571119      SLPEEDLDKNEK
## 2 Strep10_1_3 Spyo_Exp3652_DDB_SeqID_1571119      SLPEEDLDKNEK
## 3  Strep0_2_2 Spyo_Exp3652_DDB_SeqID_1571119     TIFDDEPISEETK
## 4 Strep10_2_4 Spyo_Exp3652_DDB_SeqID_1571119     TIFDDEPISEETK
## 5  Strep0_1_1 Spyo_Exp3652_DDB_SeqID_1571119 LSLPSQEPLLAAFHGEK
## 6 Strep10_2_4 Spyo_Exp3652_DDB_SeqID_1571119 LSLPSQEPLLAAFHGEK
##                                          transition_id
## 1                  AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8
## 2                  AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8
## 3                  AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8
## 4                  AHIAYLPSDGR 118149_AHIAYLPSDGR/2_y8
## 5 EKAEAAIYQFLEAIGENPNR 28903_EKAEAAIYQFLEAIGENPNR/3_y6
## 6 EKAEAAIYQFLEAIGENPNR 28903_EKAEAAIYQFLEAIGENPNR/3_y6
##       peptide_sequence precursor_charge transition_intensity concentration
## 1          AHIAYLPSDGR                2                 4036             ?
## 2          AHIAYLPSDGR                2                 1642             ?
## 3          AHIAYLPSDGR                2                 2405             ?
## 4          AHIAYLPSDGR                2                  720             ?
## 5 EKAEAAIYQFLEAIGENPNR                3                 3410             ?
## 6 EKAEAAIYQFLEAIGENPNR                3                 1984             ?
```

Session info on the R version and packages used.

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Debian GNU/Linux buster/sid
```

```
##
## Matrix products: default
## BLAS: /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.2.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.utf8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.utf8        LC_COLLATE=en_US.utf8
##  [5] LC_MONETARY=en_US.utf8    LC_MESSAGES=en_US.utf8
##  [7] LC_PAPER=en_US.utf8       LC_NAME=C
##  [9] LC_ADDRESS=C              LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.utf8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] data.table_1.11.4  SWATH2stats_1.11.3
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17         knitr_1.20           magrittr_1.5
##  [4] devtools_1.13.5      munsell_0.5.0        colorspace_1.3-2
##  [7] rlang_0.2.1          stringr_1.3.1        plyr_1.8.4
## [10] tools_3.5.0          grid_3.5.0           gtable_0.2.0
## [13] withr_2.1.2          htmltools_0.3.6      yaml_2.1.19
## [16] lazyeval_0.2.1       rprojroot_1.3-2      digest_0.6.15
## [19] tibble_1.4.2         reshape2_1.4.3       formatR_1.5
## [22] ggplot2_2.2.1        memoise_1.1.0        evaluate_0.10.1
## [25] rmarkdown_1.10       labeling_0.3         stringi_1.2.3
## [28] pillar_1.2.3         compiler_3.5.0       BiocInstaller_1.30.0
## [31] scales_0.5.0         backports_1.1.2
```