Handling metadata and annotations

AlpsNMR authors

2023-02-16

Abstract

This vignette shows some examples on how to explore sample metadata and add additional sample annotations, coming from one or more CSV or Excel files.

Package

AlpsNMR 4.0.4

Contents

1 Getting started

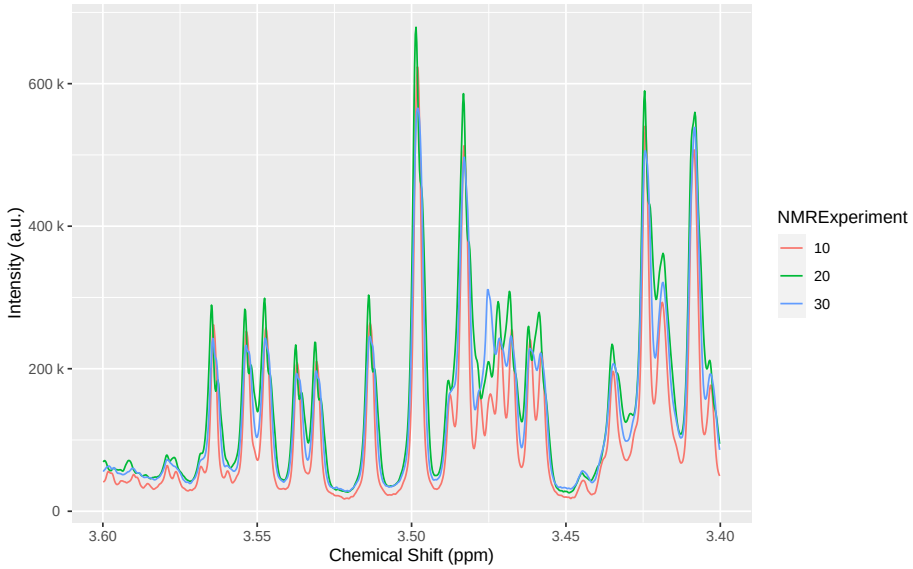
We start by loading `AlpsNMR` and some convenience libraries:

```
library(dplyr)
library(readxl)
library(AlpsNMR)
```

We also load the demo samples, see the introduction vignette for further details:

```
MeOH_plasma_extraction_dir <- system.file("dataset-demo", package = "AlpsNMR")
zip_files <- list.files(MeOH_plasma_extraction_dir, pattern = glob2rx("*.zip"), full.names = TRUE)
dataset <- nmr_read_samples(sample_names = zip_files)
dataset <- nmr_interpolate_1D(dataset, axis = NULL)
dataset
## An nmr_dataset_1D (3 samples)
```

```
plot(dataset, chemshift_range = c(3.4, 3.6))
```



2 Exploring the sample metadata

Most NMR formats include besides the actual NMR spectra, a lot of additional information describing the acquisition properties, instrument settings, and spectral processing information.

`AlpsNMR` parses all that information whenever possible, and stores it in the `nmr_dataset`object, so the user can inspect it. Since there may be a lot of information, the data is stored in several data frames.

The available data frames are:

```
nmr_meta_groups(dataset)
## [1] "info"     "orig"     "title"    "acqus"    "procs"    "levels"   "external"
```

We can further explore each of those groups.

For instance, for the `acqus` group we find 239 columns:

```
acqus_metadata <- nmr_meta_get(dataset, groups = "acqus")
acqus_metadata
## # A tibble: 3 x 239
##   NMRExperiment acqus_~1 acqus~2 acqus~3 acqus~4 acqus~5 acqus~6 acqus~7 acqus~8
##   <chr>         <chr>      <dbl> <chr>   <chr>   <chr>   <chr>   <list>    <dbl>
## 1 10            Paramet~       5 Parame~ "13\t$~ Bruker~ nmrsu   <chr>     -8.36
## 2 20            Paramet~       5 Parame~ "15\t$~ Bruker~ nmrsu   <chr>     -8.53
## 3 30            Paramet~       5 Parame~ "13\t$~ Bruker~ nmrsu   <chr>     -8.35
## # ... with 230 more variables: acqus_AMP <list>, acqus_AMPCOIL <list>,
## #   acqus_ANAVPT <dbl>, acqus_AQSEQ <dbl>, acqus_AQ_mod <dbl>,
## #   acqus_AUNM <chr>, acqus_AUTOPOS <chr>, acqus_BF1 <dbl>, acqus_BF2 <dbl>,
## #   acqus_BF3 <dbl>, acqus_BF4 <dbl>, acqus_BF5 <dbl>, acqus_BF6 <dbl>,
## #   acqus_BF7 <dbl>, acqus_BF8 <dbl>, acqus_BWFAC <list>, acqus_BYTORDA <dbl>,
## #   acqus_CAGPARS <list>, acqus_CHEMSTR <chr>, acqus_CNST <list>,
## #   acqus_CPDPRG <chr>, acqus_D <list>, acqus_DATE <dbl>, acqus_DE <dbl>, ...
```

Here follows a long list of all the columns available:

```
colnames(acqus_metadata)
##   [1] "NMRExperiment"      "acqus_TITLE"        "acqus_JCAMPDX"
##   [4] "acqus_DATATYPE"     "acqus_NPOINTS"      "acqus_ORIGIN"
##   [7] "acqus_OWNER"        "acqus_Stamp"        "acqus_ACQT0"
##  [10] "acqus_AMP"          "acqus_AMPCOIL"      "acqus_ANAVPT"
##  [13] "acqus_AQSEQ"        "acqus_AQ_mod"       "acqus_AUNM"
##  [16] "acqus_AUTOPOS"      "acqus_BF1"          "acqus_BF2"
##  [19] "acqus_BF3"          "acqus_BF4"          "acqus_BF5"
##  [22] "acqus_BF6"          "acqus_BF7"          "acqus_BF8"
##  [25] "acqus_BWFAC"        "acqus_BYTORDA"      "acqus_CAGPARS"
##  [28] "acqus_CHEMSTR"      "acqus_CNST"         "acqus_CPDPRG"
##  [31] "acqus_D"            "acqus_DATE"         "acqus_DE"
##  [34] "acqus_DECBNUC"      "acqus_DECIM"        "acqus_DECNUC"
##  [37] "acqus_DECSTAT"      "acqus_DIGMOD"       "acqus_DIGTYP"
##  [40] "acqus_DQDMODE"      "acqus_DR"           "acqus_DS"
##  [43] "acqus_DSPFIRM"      "acqus_DSPFVS"       "acqus_DTYPA"
##  [46] "acqus_EXP"          "acqus_FCUCHAN"      "acqus_FL1"
##  [49] "acqus_FL2"          "acqus_FL3"          "acqus_FL4"
##  [52] "acqus_FN_INDIRECT"  "acqus_FOV"          "acqus_FQ1LIST"
##  [55] "acqus_FQ2LIST"      "acqus_FQ3LIST"      "acqus_FQ4LIST"
##  [58] "acqus_FQ5LIST"      "acqus_FQ6LIST"      "acqus_FQ7LIST"
##  [61] "acqus_FQ8LIST"      "acqus_FRQLO3"       "acqus_FRQLO3N"
##  [64] "acqus_FS"           "acqus_FTLPGN"       "acqus_FW"
##  [67] "acqus_FnILOOP"      "acqus_FnMODE"       "acqus_FnTYPE"
##  [70] "acqus_GPNAM"        "acqus_GPX"          "acqus_GPY"
##  [73] "acqus_GPZ"          "acqus_GRDPROG"      "acqus_GRPDLY"
##  [76] "acqus_HDDUTY"       "acqus_HDRATE"       "acqus_HGAIN"
##  [79] "acqus_HL1"          "acqus_HL2"          "acqus_HL3"
##  [82] "acqus_HL4"          "acqus_HOLDER"       "acqus_HPMOD"
##  [85] "acqus_HPPRGN"       "acqus_IN"           "acqus_INF"
##  [88] "acqus_INP"          "acqus_INSTRUM"      "acqus_INTEGFAC"
```

```
##  [91] "acqus_L"            "acqus_LFILTER"      "acqus_LGAIN"
##  [94] "acqus_LINPSTP"      "acqus_LOCKED"       "acqus_LOCKFLD"
##  [97] "acqus_LOCKGN"       "acqus_LOCKPOW"      "acqus_LOCKPPM"
## [100] "acqus_LOCNUC"       "acqus_LOCPHAS"      "acqus_LOCSHFT"
## [103] "acqus_LOCSW"        "acqus_LTIME"        "acqus_MASR"
## [106] "acqus_MASRLST"      "acqus_MULEXPNO"     "acqus_NBL"
## [109] "acqus_NC"           "acqus_NLOGCH"       "acqus_NOVFLW"
## [112] "acqus_NS"           "acqus_NUC1"         "acqus_NUC2"
## [115] "acqus_NUC3"         "acqus_NUC4"         "acqus_NUC5"
## [118] "acqus_NUC6"         "acqus_NUC7"         "acqus_NUC8"
## [121] "acqus_NUCLEUS"      "acqus_NUSLIST"      "acqus_NusAMOUNT"
## [124] "acqus_NusFPNZ"      "acqus_NusJSP"       "acqus_NusSEED"
## [127] "acqus_NusSPTYPE"    "acqus_NusT2"        "acqus_NusTD"
## [130] "acqus_O1"           "acqus_O2"           "acqus_O3"
## [133] "acqus_O4"           "acqus_O5"           "acqus_O6"
## [136] "acqus_O7"           "acqus_O8"           "acqus_OVERFLW"
## [139] "acqus_P"            "acqus_PACOIL"       "acqus_PAPS"
## [142] "acqus_PARMODE"      "acqus_PCPD"         "acqus_PEXSEL"
## [145] "acqus_PHCOR"        "acqus_PHLIST"       "acqus_PHP"
## [148] "acqus_PH_ref"       "acqus_PL"           "acqus_PLSTEP"
## [151] "acqus_PLSTRT"       "acqus_PLW"          "acqus_PLWMAX"
## [154] "acqus_PQPHASE"      "acqus_PQSCALE"      "acqus_PR"
## [157] "acqus_PRECHAN"      "acqus_PRGAIN"       "acqus_PROBHD"
## [160] "acqus_PULPROG"      "acqus_PW"           "acqus_PYNM"
## [163] "acqus_ProjAngle"    "acqus_QNP"          "acqus_RD"
## [166] "acqus_RECCHAN"      "acqus_RECPH"        "acqus_RECPRE"
## [169] "acqus_RECPRFX"      "acqus_RECSEL"       "acqus_RG"
## [172] "acqus_RO"           "acqus_RSEL"         "acqus_S"
## [175] "acqus_SELREC"       "acqus_SFO1"         "acqus_SFO2"
## [178] "acqus_SFO3"         "acqus_SFO4"         "acqus_SFO5"
## [181] "acqus_SFO6"         "acqus_SFO7"         "acqus_SFO8"
## [184] "acqus_SOLVENT"      "acqus_SOLVOLD"      "acqus_SP"
## [187] "acqus_SPECTR"       "acqus_SPINCNT"      "acqus_SPNAM"
## [190] "acqus_SPOAL"        "acqus_SPOFFS"       "acqus_SPPEX"
## [193] "acqus_SPW"          "acqus_SUBNAM"       "acqus_SW"
## [196] "acqus_SWIBOX"       "acqus_SW_h"         "acqus_SWfinal"
## [199] "acqus_SigLockShift" "acqus_TD"           "acqus_TD0"
## [202] "acqus_TD_INDIRECT"  "acqus_TDav"         "acqus_TE"
## [205] "acqus_TE1"          "acqus_TE2"          "acqus_TE3"
## [208] "acqus_TE4"          "acqus_TEG"          "acqus_TE_MAGNET"
## [211] "acqus_TE_PIDX"      "acqus_TE_STAB"      "acqus_TL"
## [214] "acqus_TOTROT"       "acqus_TUBE_TYPE"    "acqus_USERA1"
## [217] "acqus_USERA2"       "acqus_USERA3"       "acqus_USERA4"
## [220] "acqus_USERA5"       "acqus_V9"           "acqus_VALIDCODE"
## [223] "acqus_VALIST"       "acqus_VCLIST"       "acqus_VDLIST"
## [226] "acqus_VPLIST"       "acqus_VTLIST"       "acqus_WBST"
## [229] "acqus_WBSW"         "acqus_XGAIN"        "acqus_XL"
## [232] "acqus_YL"           "acqus_YMAX_a"       "acqus_YMIN_a"
## [235] "acqus_ZGOPTNS"      "acqus_ZL1"          "acqus_ZL2"
## [238] "acqus_ZL3"          "acqus_ZL4"
```

We can check for instance that the nuclei used on all samples is 1H:

```
acqus_metadata[, c("NMRExperiment", "acqus_NUC1")]
## # A tibble: 3 x 2
##   NMRExperiment acqus_NUC1
##   <chr>         <chr>
## 1 10            1H
## 2 20            1H
## 3 30            1H
```

Similarly, we can obtain the processing settings:

```
procs_metadata <- nmr_meta_get(dataset, groups = "procs")
procs_metadata
## # A tibble: 3 x 137
##   NMRExperiment procs_~1 procs~2 procs~3 procs~4 procs~5 procs~6 procs~7 procs~8
##   <chr>         <chr>      <dbl> <chr>   <chr>   <chr>   <chr>   <list>    <dbl>
## 1 10            Paramet~       5 Parame~ "6\t$$~ Bruker~ nmrsu   <chr>         0
## 2 20            Paramet~       5 Parame~ "11\t$~ Bruker~ nmrsu   <chr>         0
## 3 30            Paramet~       5 Parame~ "6\t$$~ Bruker~ nmrsu   <chr>         0
## # ... with 128 more variables: procs_ABSF2 <dbl>, procs_ABSG <dbl>,
## #   procs_ABSL <dbl>, procs_ALPHA <dbl>, procs_AQORDER <dbl>,
## #   procs_ASSFAC <dbl>, procs_ASSFACI <dbl>, procs_ASSFACX <dbl>,
## #   procs_ASSWID <dbl>, procs_AUNMP <chr>, procs_AXLEFT <dbl>,
## #   procs_AXNAME <chr>, procs_AXNUC <chr>, procs_AXRIGHT <dbl>,
## #   procs_AXTYPE <dbl>, procs_AXUNIT <chr>, procs_AZFE <dbl>, procs_AZFW <dbl>,
## #   procs_BCFW <dbl>, procs_BC_mod <dbl>, procs_BYTORDP <dbl>, ...
```

3 Sample annotations

Besides the sample metadata, most studies usually have design variables or annotations, that describe the biological sample. These annotations do not come from the instrument itself, but rather usually are defined on an *external* CSV or Excel file.

`AlpsNMR` supports adding *external* annotations from data frames.

Let's load a table from an Excel file, that has some annotations for our demo dataset:

```
excel_file <- file.path(MeOH_plasma_extraction_dir, "dummy_metadata.xlsx")
subject_timepoint <- read_excel(excel_file, sheet = 1)
subject_timepoint
## # A tibble: 3 x 3
##   NMRExperiment SubjectID TimePoint
##   <chr>         <chr>     <chr>
## 1 10            Ana       baseline
## 2 20            Ana       3 months
## 3 30            Elia      baseline
```

Note how this table includes a first column named `NMRExperiment`. This column allows us to match the rows in the table with our samples.

We can embed these external annotations in our dataset:

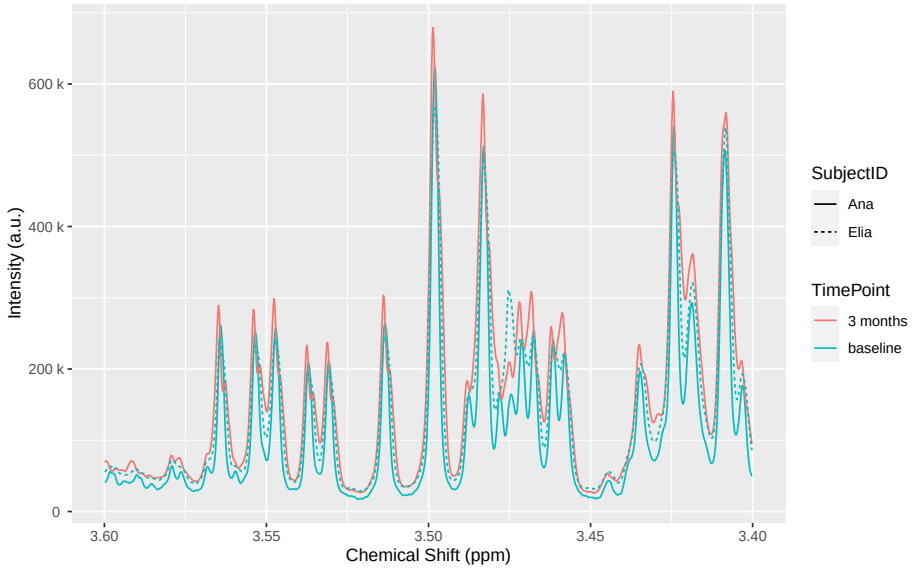
```
dataset <- nmr_meta_add(dataset, metadata = subject_timepoint, by = "NMRExperiment")
```

We can retrieve these *external* columns from the dataset:

```
nmr_meta_get(dataset, groups = "external")
## # A tibble: 3 x 3
##   NMRExperiment SubjectID TimePoint
##   <chr>         <chr>     <chr>
## 1 10            Ana       baseline
## 2 20            Ana       3 months
## 3 30            Elia      baseline
```

After adding the annotations to the dataset, we can use them in plots:

```
plot(dataset, color = "TimePoint", linetype = "SubjectID", chemshift_range = c(3.4, 3.6))
```



4 Further annotations

Sometimes due to the study design we have more than one table that we want to match with our data.

For instance, a collaborator just sent us this table:

```
additional_annotations <- data.frame(
    NMRExperiment = c("10", "20", "30"),
    SampleCollectionDay = c(1, 91, 3)
)
additional_annotations
##   NMRExperiment SampleCollectionDay
## 1            10                  1
## 2            20                 91
## 3            30                  3
```

Since we have the `NMRExperiment` column it is very easy to include it:

```
dataset <- nmr_meta_add(dataset, additional_annotations)
```

And the column has been added:

```
nmr_meta_get(dataset, groups = "external")
## # A tibble: 3 x 4
##   NMRExperiment SubjectID TimePoint SampleCollectionDay
##   <chr>         <chr>     <chr>                   <dbl>
## 1 10            Ana       baseline                    1
## 2 20            Ana       3 months                   91
## 3 30            Elia      baseline                    3
```

We received further information, but this time it is related to the `SubjectID` that we added before:

```
subject_related_information <- data.frame(
    SubjectID = c("Ana", "Elia"),
    Age = c(33, 3),
    Sex = c("female", "female")
)
subject_related_information
##   SubjectID Age    Sex
## 1       Ana  33 female
## 2      Elia   3 female
```

Note how in this case we only have two rows, and we don't have the `NMRExperiment` column anymore.

We can specify the `by` argument in `nmr_meta_add()` to use another column for merging:

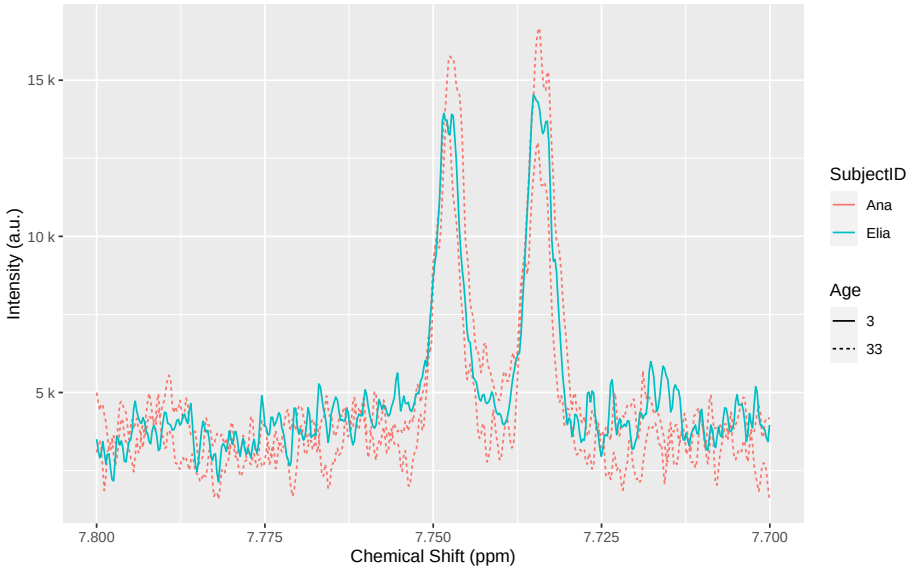
```
dataset <- nmr_meta_add(dataset, subject_related_information, by = "SubjectID")
```

And the Sex and Age columns will have been added:

```
nmr_meta_get(dataset, groups = "external")
## # A tibble: 3 x 6
##   NMRExperiment SubjectID TimePoint SampleCollectionDay   Age Sex
##   <chr>         <chr>     <chr>                   <dbl> <dbl> <chr>
## 1 10            Ana       baseline                    1    33 female
## 2 20            Ana       3 months                   91    33 female
## 3 30            Elia      baseline                    3     3 female
```

We can also use it in a plot:

```
plot(dataset, color = "SubjectID", linetype = "as.factor(Age)", chemshift_range = c(7.7, 7.8)) + ggplot2::lab
```



5 Summary

In this vignette we have seen how to explore the sample metadata, including acquisition and processing settings, and how to embed external annotations and use them in plots.

`AlpsNMR` is able to merge external annotations as long as there is a common annotation in the data that can be used as merging key.

To import external data, you may want to use the following functions:

File type	Suggested function
CSV	`readr::read_csv()`
TSV	`readr::read_tsv()`
SPSS	`haven::read_spss()`
xls/xlsx	`readxl::read_excel()`

6 Session Information

```
sessionInfo()
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.16-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.16-bioc/R/lib/libRlapack.so
##
## 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
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] AlpsNMR_4.0.4       future_1.31.0       BiocParallel_1.32.5
## [4] readxl_1.4.2        ggplot2_3.4.1       dplyr_1.1.0
## [7] BiocStyle_2.26.0
##
## loaded via a namespace (and not attached):
##  [1] matrixStats_0.63.0     fs_1.6.1               httr_1.4.4
##  [4] RColorBrewer_1.1-3     tools_4.2.2            doRNG_1.8.6
##  [7] utf8_1.2.3             R6_2.5.1               colorspace_2.1-0
## [10] withr_2.5.0            tidyselect_1.2.0       gridExtra_2.3
## [13] compiler_4.2.2         MassSpecWavelet_1.64.1 progressr_0.13.0
## [16] rvest_1.0.3            cli_3.6.0              SparseM_1.81
## [19] xml2_1.3.3             labeling_0.4.2         bookdown_0.32
## [22] scales_1.2.1           mvtnorm_1.1-3          randomForest_4.7-1.1
## [25] quadprog_1.5-8         stringr_1.5.0          digest_0.6.31
## [28] rmarkdown_2.20         pkgconfig_2.0.3        htmltools_0.5.4
## [31] parallelly_1.34.0      fastmap_1.1.0          itertools_0.1-3
## [34] rlang_1.0.6            impute_1.72.3          farver_2.1.1
## [37] generics_0.1.3         speaq_2.7.0            magrittr_2.0.3
## [40] Matrix_1.5-3           Rcpp_1.0.10            munsell_0.5.0
## [43] fansi_1.0.4            lifecycle_1.0.3        RcppZiggurat_0.1.6
## [46] stringi_1.7.12         yaml_2.3.7             MASS_7.3-58.2
## [49] plyr_1.8.8             grid_4.2.2             parallel_4.2.2
## [52] listenv_0.9.0          ggrepel_0.9.3          crayon_1.5.2
## [55] doSNOW_1.0.20          lattice_0.20-45        cowplot_1.1.1
## [58] knitr_1.42             pillar_1.8.1           igraph_1.4.0
## [61] rngtools_1.5.2         corpcor_1.6.10         reshape2_1.4.4
## [64] codetools_0.2-19       mixOmics_6.22.0        lpSolve_5.6.18
## [67] glue_1.6.2             evaluate_0.20          data.table_1.14.6
## [70] BiocManager_1.30.19    vctrs_0.5.2            missForest_1.5
## [73] foreach_1.5.2          cellranger_1.1.0       gtable_0.3.1
## [76] purrr_1.0.1            tidyr_1.3.0            xfun_0.37
## [79] limSolve_1.5.6         Rfast_2.0.7            RSpectra_0.16-1
## [82] baseline_1.3-4         pcaPP_2.0-3            rARPACK_0.11-0
## [85] signal_0.7-7           tibble_3.1.8           snow_0.4-4
## [88] iterators_1.0.14       tinytex_0.44           ellipse_0.4.3
## [91] cluster_2.1.4          globals_0.16.2
```