Supplementary Text S2: Identification and classification of duplicated genes in Ensembl and Ensembl Genomes

Fabricio Almeida-Silva [1] ***and Yves Van de Peer*** [1]

[1]VIB-UGent Center for Plant Systems Biology, Ghent University, Ghent, Belgium

27 February 2024

Contents

Supplementary Text S2: Identification and classification of duplicated genes in Ensembl and Ensembl Genomes

1 Introduction

Here, we will describe the code to identify and classify duplicated genes in Ensembl and Ensembl Genomes species using the Bioconductor package *doubletrouble*.

```
library(syntenet)
library(doubletrouble)
library(biomaRt)
library(here)
## here() starts at /home/faalm/Dropbox/package_benchmarks/doubletrouble_paper
library(tidyverse)
## -- Attaching core tidyverse packages ----------------------- tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks biomaRt::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

set.seed(123) # for reproducibility
options(timeout = 1e10) # to allow download of big files

# Load helper functions
source(here("code", "utils.R"))
```

2 Data loading: species trees and metadata

Here, we will load the data frames of species metadata and `phylo` objects with species trees for each Ensembl instance.

```
# Load metadata
load(here("products", "result_files", "metadata_all.rda"))
names(metadata_all)
## [1] "fungi"    "plants"   "metazoa"  "protists" "ensembl"

# Load trees
load(here("products", "result_files", "trees", "fungi_busco_trees.rda"))
load(here("products", "result_files", "trees", "plants_busco_trees.rda"))
load(here("products", "result_files", "trees", "metazoa_busco_trees.rda"))
load(here("products", "result_files", "trees", "vertebrates_busco_trees.rda"))
load(here("products", "result_files", "trees", "protists_busco_trees.rda"))
```

3 Identification and classification of duplicated genes in Ensembl and Ensembl Genomes

Now, let's use *doubletrouble* to identify duplicated genes and classify them using the Ensembl and Ensembl Genomes data sets. Here, to avoid code repetition and optimize memory usage, we will use the wrapper function `ensembl2duplicates()` (in the file *utils.R*). For each species in the metadata data frame, this function:

1. Retrieves whole-genome protein sequences (`AAStringSet`) and gene annotation (`GRanges`) from an Ensembl instance;
2. Filters the `AAStringSet` object to include only the longest protein for each gene (i.e., the translated sequence of the primary transcript);
3. Processes the sequences and annotation with `syntenet::process_input()`;
4. Identifies the paranome with `syntenet::run_diamond()` + identifies orthologs between the query species and an outgroup (optional);
5. Classifies paralogs by duplication modes.

3.1 Ensembl Fungi

First, let's create a data frame with species and their outgroups. Here, we will use the basidiomycete *Cryptococcus neoformans* as outgroup for Ascomycota species, and the oomycete *Aphanomyces astaci* as outgroup for Basidiomycota species.

```
col_dir <- here("products", "result_files", "collinearity", "fungi")
if(!dir.exists(col_dir)) { dir.create(col_dir, recursive = TRUE) }

# Create data frame of query species and outgroup
fungi_outgroups <- metadata_all$fungi |>
    filter(phylum != "Oomycota") |>
    mutate(
        query = species,
        outgroup = case_when(
            phylum == "Ascomycota" ~ "cryptococcus_neoformans",
            TRUE ~ "aphanomyces_astaci"
        )
    ) |>
    select(query, outgroup)

# Identifying and classifying paralogs
fungi_duplicates <- ensembl2duplicates(
    metadata_all$fungi, ensembl = "fungi",
    outgroups = fungi_outgroups,
    collinearity_dir = col_dir
)

# Classify genes into unique duplication modes
fungi_duplicates_unique <- classify_genes(fungi_duplicates)

# Save classification results
```

```
## Duplicate pairs
save(
    fungi_duplicates,
    file = here("products", "result_files", "fungi_duplicates.rda"),
    compress = "xz"
)

## Duplicated genes (unique duplication modes)
save(
    fungi_duplicates_unique,
    file = here("products", "result_files", "fungi_duplicates_unique.rda"),
    compress = "xz"
)
```

3.2 Ensembl Protists

Since protists are not a real (i.e., monophyletic) group, defining an outgroup species is very problematic. For this reason, we will classify duplicates using the *standard* classification scheme here.

```
col_dir <- here("products", "result_files", "collinearity", "protists")
if(!dir.exists(col_dir)) { dir.create(col_dir, recursive = TRUE) }

# Identifying and classifying paralogs
protists_duplicates <- ensembl2duplicates(
    metadata_all$protists, ensembl = "protists", collinearity_dir = col_dir
)

# Classify genes into unique duplication modes
protists_duplicates_unique <- classify_genes(protists_duplicates)

# Save classification results
## Duplicate pairs
save(
    protists_duplicates,
    file = here("products", "result_files", "protists_duplicates.rda"),
    compress = "xz"
)

## Duplicated genes (unique duplication modes)
save(
    protists_duplicates_unique,
    file = here("products", "result_files", "protists_duplicates_unique.rda"),
    compress = "xz"
)
```

3.3 Ensembl Plants

Here, we will use different outgroups for different branches of the tree. The clades and outgroups are:

Supplementary Text S2: Identification and classification of duplicated genes in Ensembl and Ensembl Genomes

1. Angiosperms: *Amborella trichopoda* as outgroup.
2. *Amborella trichopoda* and *Nymphaea colorata*: *Chara braunii* as outgroup.
3. *Selaginella moellendorffii*, *Chara braunii*, *Marchantia polymorpha*, *Physcomitrium patens*: *Chlamydomonas reinhardtii* as outgroup.
4. *Chlamydomonas reinhardtii* and *Ostreococcus lucimarinus*: *Galdieria sulphuraria* as outgroup
5. Rhodophyta algae: no outgroup.

```
# Create data frame of query species and outgroup
angiosperms <- metadata_all$plants |>
    filter(
        phylum == "Streptophyta",
        !order %in% c(
            "Charales", "Selaginellales", "Funariales",
            "Marchantiales", "Nymphaeales"
        )
    ) |>
    pull(species)

ana <- c("amborella_trichopoda", "nymphaea_colorata")
bryophytes <- c(
    "selaginella_moellendorffii", "chara_braunii",
    "marchantia_polymorpha", "physcomitrium_patens"
)
chlorophyta <- c("chlamydomonas_reinhardtii", "ostreococcus_lucimarinus")

plants_outgroups <- metadata_all$plants |>
    filter(phylum != "Rhodophyta") |>
    mutate(
        query = species,
        outgroup = case_when(
            species %in% angiosperms ~ "amborella_trichopoda",
            species %in% ana ~ "chara_braunii",
            species %in% bryophytes ~ "chlamydomonas_reinhardtii",
            species %in% chlorophyta ~ "galdieria_sulphuraria"
        )
    ) |>
    select(query, outgroup)
```

Identifying and classifying duplicates:

```
col_dir <- here("products", "result_files", "collinearity", "plants")
if(!dir.exists(col_dir)) { dir.create(col_dir, recursive = TRUE) }

# Identifying and classifying paralogs
plants_duplicates <- ensembl2duplicates(
    metadata_all$plants, ensembl = "plants",
    outgroups = plants_outgroups,
    collinearity_dir = col_dir,
    threads = 4
```

```
)

# Classify genes into unique duplication modes
plants_duplicates_unique <- classify_genes(plants_duplicates)

# Save classification results
## Duplicate pairs
save(
    plants_duplicates,
    file = here("products", "result_files", "plants_duplicates.rda"),
    compress = "xz"
)

## Duplicated genes (unique duplication modes)
save(
    plants_duplicates_unique,
    file = here("products", "result_files", "plants_duplicates_unique.rda"),
    compress = "xz"
)
```

3.4 Ensembl Metazoa

Here, we will use different outgroups for different branches of the tree. The clades and outgroups are:

1. Arthropoda: *Hypsibius exemplaris* (Tardigrada) as outgroup.
2. Tardigrada, Acanthocephala, and Rotifera: *Brugia malayi* (Nematoda) as outgroup.
3. Nematoda: *Priapulus caudatus* (Priapulida) as outgroup
4. Priapulida, Echinodermata, Chordata, and Hemichordata: *Hofstenia miamia* (Xenacoelomorpha) as outgroup.
5. Xenacoelomorpha: *Actinia tenebrosa* (Cnidaria) as outgroup.
6. Cnidaria and Placozoa: *Amphimedon queenslandica* (Porifera) as outgroup.
7. Porifera: *Mnemiopsis leidyi* (Ctenophora) as outgroup.
8. Brachiopoda: *Haliotis rufescens* (Mollusca) as outgroup.
9. Mollusca, Annelida, and Platyhelminthes: *Adineta vaga* (Rotifera) as outgroup.

```
# Create data frame of query species and outgroup
by_phylum <- function(df, taxon) {
    return(
        df |>
            dplyr::filter(phylum == taxon) |>
            dplyr::pull(species)
    )
}

arthropoda <- by_phylum(metadata_all$metazoa, "Arthropoda")
tardigrada <- by_phylum(metadata_all$metazoa, "Tardigrada")
```

```
nematoda <- by_phylum(metadata_all$metazoa, "Nematoda")
priapulida <- by_phylum(metadata_all$metazoa, "Priapulida")
xenacoelomorpha <- by_phylum(metadata_all$metazoa, "Xenacoelomorpha")
cnidaria <- by_phylum(metadata_all$metazoa, "Cnidaria")
placozoa <- by_phylum(metadata_all$metazoa, "Placozoa")
porifera <- by_phylum(metadata_all$metazoa, "Porifera")
brachiopoda <- by_phylum(metadata_all$metazoa, "Brachiopoda")
mollusca <- by_phylum(metadata_all$metazoa, "Mollusca")
echinodermata <- by_phylum(metadata_all$metazoa, "Echinodermata")
annelida <- by_phylum(metadata_all$metazoa, "Annelida")
platyhelminthes <- by_phylum(metadata_all$metazoa, "Platyhelminthes")
acanthocephala <- by_phylum(metadata_all$metazoa, "Acanthocephala")
chordata <- by_phylum(metadata_all$metazoa, "Chordata")
hemichordata <- by_phylum(metadata_all$metazoa, "Hemichordata")
rotifera <- by_phylum(metadata_all$metazoa, "Rotifera")

metazoa_outgroups <- metadata_all$metazoa |>
    filter(phylum != "Ctenophora") |>
    mutate(
        query = species,
        outgroup = case_when(
            species %in% arthropoda ~ "hypsibius_exemplaris_gca002082055v1",
            species %in% c(tardigrada, acanthocephala, rotifera) ~ "brugia_malayi",
            species %in% nematoda ~ "priapulus_caudatus_gca000485595v2",

            species %in% c(priapulida, echinodermata, chordata, hemichordata) ~
                "hofstenia_miamia",
            species %in% xenacoelomorpha ~ "actinia_tenebrosa_gca009602425v1",
            species %in% c(cnidaria, placozoa) ~
                "amphimedon_queenslandica_gca000090795v2rs",
            species %in% porifera ~ "mnemiopsis_leidyi",
            species %in% brachiopoda ~ "haliotis_rufescens_gca023055435v1rs",
            species %in% c(mollusca, annelida, platyhelminthes) ~ "adineta_vaga"
        )
    ) |>
    select(query, outgroup)
```

Identifying and classifying duplicates:

```
col_dir <- here("products", "result_files", "collinearity", "metazoa")
if(!dir.exists(col_dir)) { dir.create(col_dir, recursive = TRUE) }

# Identifying and classifying paralogs
metazoa_duplicates <- ensembl2duplicates(
    metadata = metadata_all$metazoa,
    ensembl = "metazoa",
    outgroups = metazoa_outgroups,
    collinearity_dir = col_dir,
    threads = 4
)
```

```
# Classify genes into unique duplication modes
metazoa_duplicates_unique <- classify_genes(metazoa_duplicates)

# Save classification results
## Duplicate pairs
save(
    metazoa_duplicates,
    file = here("products", "result_files", "metazoa_duplicates.rda"),
    compress = "xz"
)

## Duplicated genes (unique duplication modes)
save(
    metazoa_duplicates_unique,
    file = here("products", "result_files", "metazoa_duplicates_unique.rda"),
    compress = "xz"
)
```

3.5 Ensembl (Vertebrates)

Here, we will use the following outgroups per taxa:

1. Amniota: *Xenopus tropicalis* (Amphibia) as outgroup;
2. Amphibia: *Latimeria chalumnae* (West Indian Ocean coelacanth)
3. All bony and cartilaginous fish: *Eptatretus burgeri* (hagfish, Agnatha)
4. Agnatha: *Ciona intestinalis* (Tunicata)

```
# Create a data frame of species and outgroups
amniota <- metadata_all$ensembl |>
    filter(
        class %in% c("Aves", "Mammalia", "Lepidosauria") |
            order %in% c("Testudines", "Crocodylia")
    ) |>
    pull(species)

amphibia <- metadata_all$ensembl |>
    filter(class == "Amphibia") |>
    pull(species)

fish <- metadata_all$ensembl |>
    filter(
        class %in% c("Actinopteri", "Chondrichthyes", "Cladistia") |
            order == "Coelacanthiformes"
    ) |>
    pull(species)

agnatha <- metadata_all$ensembl |>
    filter(
        class %in% c("Myxini", "Hyperoartia")
    ) |>
    pull(species)
```

Supplementary Text S2: Identification and classification of duplicated genes in Ensembl and Ensembl Genomes

```
ensembl_outgroups <- metadata_all$ensembl |>
    filter(!phylum %in% c("Nematoda", "Arthropoda", "Ascomycota")) |>
    mutate(
        query = species,
        outgroup = case_when(
            species %in% amniota ~ "xenopus_tropicalis",
            species %in% amphibia ~ "latimeria_chalumnae",
            species %in% fish ~ "eptatretus_burgeri",
            species %in% agnatha ~ "ciona_intestinalis"
        )
    ) |>
    select(query, outgroup) |>
    filter(!is.na(outgroup))
```

Identifying and classifying duplicates:

```
col_dir <- here("products", "result_files", "collinearity", "vertebrates")
if(!dir.exists(col_dir)) { dir.create(col_dir, recursive = TRUE) }

# Identifying and classifying paralogs
vertebrates_duplicates <- ensembl2duplicates(
    meta,
    ensembl = "ensembl",
    outgroups = ensembl_outgroups,
    collinearity_dir = col_dir,
    tsv_dir = "~/Documents/vertebrates_duplicates", # delete later
    threads = 4
)

# Classify genes into unique duplication modes
vertebrates_duplicates_unique <- classify_genes(vertebrates_duplicates)

# Save classification results
## Duplicate pairs
save(
    vertebrates_duplicates,
    file = here("products", "result_files", "vertebrates_duplicates.rda"),
    compress = "xz"
)

## Duplicated genes (unique duplication modes)
save(
    vertebrates_duplicates_unique,
    file = here("products", "result_files", "vertebrates_duplicates_unique.rda"),
    compress = "xz"
)
```

Session info

This document was created under the following conditions:

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.3.2 (2023-10-31)
##  os       Ubuntu 22.04.3 LTS
##  system   x86_64, linux-gnu
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  ctype    en_US.UTF-8
##  tz       Europe/Brussels
##  date     2024-02-27
##  pandoc   3.1.1 @ /usr/lib/rstudio/resources/app/bin/quarto/bin/tools/ (via rmarkdown)
##
## - Packages -------------------------------------------------------------------
##  package              * version     date (UTC) lib source
##  abind                  1.4-5       2016-07-21 [1] CRAN (R 4.3.2)
##  ade4                   1.7-22      2023-02-06 [1] CRAN (R 4.3.2)
##  AnnotationDbi          1.64.1      2023-11-03 [1] Bioconductor
##  ape                    5.7-1       2023-03-13 [1] CRAN (R 4.3.2)
##  Biobase                2.62.0      2023-10-24 [1] Bioconductor
##  BiocFileCache          2.10.1      2023-10-26 [1] Bioconductor
##  BiocGenerics           0.48.1      2023-11-01 [1] Bioconductor
##  BiocIO                 1.12.0      2023-10-24 [1] Bioconductor
##  BiocManager            1.30.22     2023-08-08 [1] CRAN (R 4.3.2)
##  BiocParallel           1.37.0      2024-01-19 [1] Github (Bioconductor/BiocParallel@79a1b2d)
##  BiocStyle            * 2.30.0      2023-10-24 [1] Bioconductor
##  biomaRt              * 2.58.2      2024-01-30 [1] Bioconductor 3.18 (R 4.3.2)
##  Biostrings             2.70.2      2024-01-28 [1] Bioconductor 3.18 (R 4.3.2)
##  bit                    4.0.5       2022-11-15 [1] CRAN (R 4.3.2)
##  bit64                  4.0.5       2020-08-30 [1] CRAN (R 4.3.2)
##  bitops                 1.0-7       2021-04-24 [1] CRAN (R 4.3.2)
##  blob                   1.2.4       2023-03-17 [1] CRAN (R 4.3.2)
##  bookdown               0.37        2023-12-01 [1] CRAN (R 4.3.2)
##  cachem                 1.0.8       2023-05-01 [1] CRAN (R 4.3.2)
##  cli                    3.6.2       2023-12-11 [1] CRAN (R 4.3.2)
##  coda                   0.19-4.1    2024-01-31 [1] CRAN (R 4.3.2)
##  codetools              0.2-19      2023-02-01 [4] CRAN (R 4.2.2)
##  colorspace             2.1-0       2023-01-23 [1] CRAN (R 4.3.2)
##  crayon                 1.5.2       2022-09-29 [1] CRAN (R 4.3.2)
##  curl                   5.2.0       2023-12-08 [1] CRAN (R 4.3.2)
##  DBI                    1.2.1       2024-01-12 [1] CRAN (R 4.3.2)
##  dbplyr                 2.4.0       2023-10-26 [1] CRAN (R 4.3.2)
##  DelayedArray           0.28.0      2023-10-24 [1] Bioconductor
##  digest                 0.6.34      2024-01-11 [1] CRAN (R 4.3.2)
##  doParallel             1.0.17      2022-02-07 [1] CRAN (R 4.3.2)
##  doubletrouble        * 1.3.4       2024-02-05 [1] Bioconductor
##  dplyr                * 1.1.4       2023-11-17 [1] CRAN (R 4.3.2)
```

```
##  evaluate              0.23        2023-11-01 [1] CRAN (R 4.3.2)
##  fansi                 1.0.6       2023-12-08 [1] CRAN (R 4.3.2)
##  fastmap               1.1.1       2023-02-24 [1] CRAN (R 4.3.2)
##  filelock              1.0.3       2023-12-11 [1] CRAN (R 4.3.2)
##  forcats             * 1.0.0       2023-01-29 [1] CRAN (R 4.3.2)
##  foreach               1.5.2       2022-02-02 [1] CRAN (R 4.3.2)
##  generics              0.1.3       2022-07-05 [1] CRAN (R 4.3.2)
##  GenomeInfoDb          1.38.6      2024-02-08 [1] Bioconductor 3.18 (R 4.3.2)
##  GenomeInfoDbData      1.2.11      2023-12-21 [1] Bioconductor
##  GenomicAlignments     1.38.2      2024-01-16 [1] Bioconductor 3.18 (R 4.3.2)
##  GenomicFeatures       1.54.3      2024-01-31 [1] Bioconductor 3.18 (R 4.3.2)
##  GenomicRanges         1.54.1      2023-10-29 [1] Bioconductor
##  ggnetwork             0.5.13      2024-02-14 [1] CRAN (R 4.3.2)
##  ggplot2             * 3.4.4       2023-10-12 [1] CRAN (R 4.3.2)
##  glue                  1.7.0       2024-01-09 [1] CRAN (R 4.3.2)
##  gtable                0.3.4       2023-08-21 [1] CRAN (R 4.3.2)
##  here                * 1.0.1       2020-12-13 [1] CRAN (R 4.3.2)
##  hms                   1.1.3       2023-03-21 [1] CRAN (R 4.3.2)
##  htmltools             0.5.7       2023-11-03 [1] CRAN (R 4.3.2)
##  htmlwidgets           1.6.4       2023-12-06 [1] CRAN (R 4.3.2)
##  httr                  1.4.7       2023-08-15 [1] CRAN (R 4.3.2)
##  igraph                2.0.1.1     2024-01-30 [1] CRAN (R 4.3.2)
##  intergraph            2.0-4       2024-02-01 [1] CRAN (R 4.3.2)
##  IRanges               2.36.0      2023-10-24 [1] Bioconductor
##  iterators             1.0.14      2022-02-05 [1] CRAN (R 4.3.2)
##  KEGGREST              1.42.0      2023-10-24 [1] Bioconductor
##  knitr                 1.45        2023-10-30 [1] CRAN (R 4.3.2)
##  lattice               0.22-5      2023-10-24 [4] CRAN (R 4.3.1)
##  lifecycle             1.0.4       2023-11-07 [1] CRAN (R 4.3.2)
##  lubridate           * 1.9.3       2023-09-27 [1] CRAN (R 4.3.2)
##  magrittr              2.0.3       2022-03-30 [1] CRAN (R 4.3.2)
##  MASS                  7.3-60      2023-05-04 [4] CRAN (R 4.3.1)
##  Matrix                1.6-3       2023-11-14 [4] CRAN (R 4.3.2)
##  MatrixGenerics        1.14.0      2023-10-24 [1] Bioconductor
##  matrixStats           1.2.0       2023-12-11 [1] CRAN (R 4.3.2)
##  mclust                6.0.1       2023-11-15 [1] CRAN (R 4.3.2)
##  memoise               2.0.1       2021-11-26 [1] CRAN (R 4.3.2)
##  MSA2dist              1.6.0       2023-10-24 [1] Bioconductor
##  munsell               0.5.0       2018-06-12 [1] CRAN (R 4.3.2)
##  network               1.18.2      2023-12-05 [1] CRAN (R 4.3.2)
##  networkD3             0.4         2017-03-18 [1] CRAN (R 4.3.2)
##  nlme                  3.1-163     2023-08-09 [4] CRAN (R 4.3.1)
##  pheatmap              1.0.12      2019-01-04 [1] CRAN (R 4.3.2)
##  pillar                1.9.0       2023-03-22 [1] CRAN (R 4.3.2)
##  pkgconfig             2.0.3       2019-09-22 [1] CRAN (R 4.3.2)
##  png                   0.1-8       2022-11-29 [1] CRAN (R 4.3.2)
##  prettyunits           1.2.0       2023-09-24 [1] CRAN (R 4.3.2)
##  progress              1.2.3       2023-12-06 [1] CRAN (R 4.3.2)
##  purrr               * 1.0.2       2023-08-10 [1] CRAN (R 4.3.2)
##  R6                    2.5.1       2021-08-19 [1] CRAN (R 4.3.2)
##  rappdirs              0.3.3       2021-01-31 [1] CRAN (R 4.3.2)
```

```
##  RColorBrewer           1.1-3       2022-04-03 [1] CRAN (R 4.3.2)
##  Rcpp                   1.0.12      2024-01-09 [1] CRAN (R 4.3.2)
##  RCurl                  1.98-1.14   2024-01-09 [1] CRAN (R 4.3.2)
##  readr                * 2.1.5       2024-01-10 [1] CRAN (R 4.3.2)
##  restfulr               0.0.15      2022-06-16 [1] CRAN (R 4.3.2)
##  rjson                  0.2.21      2022-01-09 [1] CRAN (R 4.3.2)
##  rlang                  1.1.3       2024-01-10 [1] CRAN (R 4.3.2)
##  rmarkdown              2.25        2023-09-18 [1] CRAN (R 4.3.2)
##  rprojroot              2.0.4       2023-11-05 [1] CRAN (R 4.3.2)
##  Rsamtools              2.18.0      2023-10-24 [1] Bioconductor
##  RSQLite                2.3.5       2024-01-21 [1] CRAN (R 4.3.2)
##  rstudioapi             0.15.0      2023-07-07 [1] CRAN (R 4.3.2)
##  rtracklayer            1.62.0      2023-10-24 [1] Bioconductor
##  S4Arrays               1.2.0       2023-10-24 [1] Bioconductor
##  S4Vectors              0.40.2      2023-11-23 [1] Bioconductor 3.18 (R 4.3.2)
##  scales                 1.3.0       2023-11-28 [1] CRAN (R 4.3.2)
##  seqinr                 4.2-36      2023-12-08 [1] CRAN (R 4.3.2)
##  sessioninfo            1.2.2       2021-12-06 [1] CRAN (R 4.3.2)
##  SparseArray            1.2.4       2024-02-11 [1] Bioconductor 3.18 (R 4.3.2)
##  statnet.common         4.9.0       2023-05-24 [1] CRAN (R 4.3.2)
##  stringi                1.8.3       2023-12-11 [1] CRAN (R 4.3.2)
##  stringr              * 1.5.1       2023-11-14 [1] CRAN (R 4.3.2)
##  SummarizedExperiment   1.32.0      2023-10-24 [1] Bioconductor
##  syntenet             * 1.4.0       2023-10-24 [1] Bioconductor
##  tibble               * 3.2.1       2023-03-20 [1] CRAN (R 4.3.2)
##  tidyr                * 1.3.1       2024-01-24 [1] CRAN (R 4.3.2)
##  tidyselect             1.2.0       2022-10-10 [1] CRAN (R 4.3.2)
##  tidyverse            * 2.0.0       2023-02-22 [1] CRAN (R 4.3.2)
##  timechange             0.3.0       2024-01-18 [1] CRAN (R 4.3.2)
##  tzdb                   0.4.0       2023-05-12 [1] CRAN (R 4.3.2)
##  utf8                   1.2.4       2023-10-22 [1] CRAN (R 4.3.2)
##  vctrs                  0.6.5       2023-12-01 [1] CRAN (R 4.3.2)
##  withr                  3.0.0       2024-01-16 [1] CRAN (R 4.3.2)
##  xfun                   0.42        2024-02-08 [1] CRAN (R 4.3.2)
##  XML                    3.99-0.16.1 2024-01-22 [1] CRAN (R 4.3.2)
##  xml2                   1.3.6       2023-12-04 [1] CRAN (R 4.3.2)
##  XVector                0.42.0      2023-10-24 [1] Bioconductor
##  yaml                   2.3.8       2023-12-11 [1] CRAN (R 4.3.2)
##  zlibbioc               1.48.0      2023-10-24 [1] Bioconductor
##
##  [1] /home/faalm/R/x86_64-pc-linux-gnu-library/4.3
##  [2] /usr/local/lib/R/site-library
##  [3] /usr/lib/R/site-library
##  [4] /usr/lib/R/library
##
## ------------------------------------------------------------------------------
```