REcoTox - a workflow to process US EPA ECOTOX Knowledgebase ASCII files (PDF version)

Tobias Schulze
Helmholtz Centre for Environmental Research - UFZ, Leipzig, Germany
tsufz1@gmail.com

2023-11-04

Contents

1 Background

The search and extraction of experimental ecotoxicological information is often a tedious work. A good and comprehensive data source is the US EPA ECOTOX Knowledgebase. It contains more than 1 million data points for almost 13,000 chemicals and 14,000 single species. However, for a high-throughput hazard assessment, it is not possible to extract all relevant data of the online database. The purpose of REcoTox is to extract the relevant information and to aggregate the data based on the user criteria out of the entire database ASCII files.

2 Introduction

REcoTox is a semi-automated, interactive workflow to process US EPA ECOTOX Knowledgebase entire database ASCII files to extract and process ecotoxicological data relevant (but not restricted) to the ecotoxicity groups algae, crustaceans, and fish in the aquatic domain. The focus is aquatic ecotoxicity and the unit of the retrieved data is `mg/L`.

3 Input files and folders

`REcoTox` requires an unzipped `US EPA Knowlegdebase` database in `ASCII` format (Zitat). The database is preferable expanded in an own database folder to be defined during the processing, The database consists of relatively referenced text files. The separator of the data is the pipeline `|` symbol.

In the first session of `REcoTox`, a file `chemical_properties.csv` is created in the database folder. This files contains chemical identifiers and chemical properties required for the processing of the chemical data in the knowlegdebase and to tag the results.

The chemical property file is dynamically updated and requires also some manual curation. It will grow as soon new chemicals are added to the knowledgebase.

The `project_folder` contains the `R` script for processing as well as the intermediate and final processing files. The naming of the folder is arbitrary, but do not use spaces, but underscores (`_`) or hyphens (`-`) for separating parts.

To run the queries, a predefined processing script is available on `GitHub` (`Query_EcoTox_DB.R`) or in the local `REcoTox` package folder.

4 Using REcoTox

The following tutorial explains the different steps of `REcoTox` in a comprehensive demonstration. `REcoTox` includes different interactive steps, which require the evaluation of comma separated text files (`*.csv`) in an external spreadsheet application (preferable LibreOffice [1]).

4.1 Load the REcoTox package

```
# Load the REcoTox package
library(REcoTox)
```

4.2 Documentation for MZquant

A detailed description of all functions of `REcoTox` functions is available in the `R Documentation`.

```
# Documentation of REcoTox
help(package = "REcoTox")
```

4.3 Preparation of the working environment (for beginners)

The processing in `REcoTox` is interactivally controlled by a processing script `Query_EcoTox_DB.R`.

If you run `REcoTox` for the first time, a tutorial project is available to demonstrate all important steps of `REcoTox` processing. The following script is preparing an example folder in your home directory and copies all necessary files in the folder.

```
# Path of the project folder
project_folder <- "REcoTox_demo"

database_folder <- system.file("extdata/database_folder", package="REcoTox")
# The project folder is created in the home directory
project_path <- normalizePath(ifelse(.Platform$OS.type == "unix",
    paste0("~/", project_folder),
    paste0(
        Sys.getenv("HOMEPATH"),
        "\\",
        project_folder
    )
))

# An existing folder is deleted
#if (dir.exists(project_folder)) {
#    unlink(project_folder, recursive = TRUE)
#}
```

This command initializes the project folder and the database folder. It copies also the processing script to the project folder.

```
project <- REcoTox::create_project(database_path = database_folder,
                           project_path,
                           initalise_database_project = TRUE, # create the basic project from current AS
                           initalise_project = TRUE, # initializes the project folder
                           load_default = FALSE) # loads the default project in the project folder in th

file.copy(
                     from = system.file(
                         "extdata",
                         "Query_EcoTox_DB.R",
                         package = "REcoTox"
                     ),
                     to = normalizePath(
                         path = file.path(
                             project_folder,
                             "Query_EcoTox_DB.R"
```

```
                        ),
                        winslash = "\\",
                        mustWork = FALSE
                    ),
                    overwrite = TRUE
                )
```

```
## [1] FALSE
```

The `project_folder` contains the following files:

```
# List files and directories in project_folder
list.files(project_folder, recursive = TRUE, include.dirs = TRUE)
```

```
## character(0)
```

The `database_folder` contains the following files and folders: `chemical_properties.csv` is the file containing the curated chemical properties, `results.txt` contains the testing results collected in the knowledgebase, and `test.txt` contains the the metadate of the tests.

The folder `validation` contains the files `chemicals.txt` with chemical information, the file `references.txt` contains the references and `species.txt` the species.

```
# List files and directories in project_folder
list.files(database_folder, recursive = TRUE, include.dirs = TRUE)
```

```
## [1] "chemical_properties.csv"   "project.RData"
## [3] "results.txt"               "tests.txt"
## [5] "validation"                "validation/chemicals.txt"
## [7] "validation/references.txt" "validation/species.txt"
```

It contains only the `Query_EcoTox_DB.R` file.

4.4 Review of the input data

To review the input data, let us look in the data:

```
# Review of the chemical properties
chemical_properties <- readr::read_csv(file = normalizePath(path = file.path(
    database_folder,
    "chemical_properties.csv"
), ), show_col_types = FALSE)

kable(
    chemical_properties %>%
        select(cas_number:dtxsid_ecotox) %>%
        head(5),
    format = "latex", digits = 2
)
```

cas_number	cas	chemical_name	dtxsid_ecotox
1912249	1912-24-9	6-Chloro-N-ethyl-N'-(1-methylethyl)-1,3,5-triazine-2,4-diamine	DTXSID9020112

```
# Review of the result table
results <-
    readr::read_delim(
        file = normalizePath(
            path = file.path(
                database_folder,
                "results.txt"
            ),
        ),
        show_col_types = FALSE,
        delim = "|"

    )

kable(
    results  %>%
        select(result_id:sample_size_mean) %>%
        head(5),
    format = "latex", digits = 2
)
```

result_id	test_id	sample_size_mean_op	sample_size_mean
198582	1212738	NA	NA
2176441	2095522	NA	NA
713354	1262587	NA	NA
2200676	2110007	NA	NA
2380412	2190107	NA	NA

```
# Review of the substance_table
substances <-
    readr::read_delim(
        file = normalizePath(
            path = file.path(
                database_folder,
                "validation",
                "chemicals.txt"
            ),
        ),
        show_col_types = FALSE,
        delim = "|"

    )

kable(
    substances %>%
        select(cas_number:ecotox_group) %>%
        head(5),
    format = "latex", digits = 2
)
```

cas_number	chemical_name	ecotox_group
1912249	6-Chloro-N-ethyl-N'-(1-methylethyl)-1,3,5-triazine-2,4-diamine	Endocrine Disrupting Chemicals (EDCs)

```
# Review of the substance_table
references <-
    readr::read_delim(
        file = normalizePath(
            path = file.path(
                database_folder,
                "validation",
                "references.txt"
            ),
        ),
        show_col_types = FALSE,
        delim = "|"

    )

kable(
    references %>%
        select(reference_number:author) %>%
        head(5),
    format = "latex", digits = 2
)
```

reference_number	reference_db	reference_type	author
95	A	NA	Abou-Waly,H., M.M. Abou-Setta, H.N. Nigg, and L.L. Mallory
344	AT	OPP	U.S. Environmental Protection Agency
393	A	NA	Hersh,C.M., and W.G. Crumpton
624	A	NA	Valentine,J.P., and S.W. Bingham
682	A	NA	Isensee,A.R.

```
# Review of the substance_table
species <-
    readr::read_delim(
        file = normalizePath(
            path = file.path(
                database_folder,
                "validation",
                "species.txt"
            ),
        ),
        show_col_types = FALSE,
        delim = "|"

    )

kable(
    species %>%
        select(species_number:kingdom) %>%
        head(5),
    format = "latex", digits = 2
)
```

species_number	common_name	latin_name	kingdom
31	Green Algae	Chlorella ovalis	Plantae
47	Green Algae	Scenedesmus sp.	Plantae
147	Green Algae	Spirogyra crassa	Plantae
298	Green Algae	Dunaliella tertiolecta	Plantae
300	Green Algae	Chlorococcum sp.	Plantae

4.5 Preparation of the database environment and initialisation of the project folder

In the first step, the function `create_project` creates the database project and initializes the database and project folders:

1. Load the ASCII files.
2. Create a file `chemical_properties.csv` based o the `chemicals.txt` table. If this table exists, it is loaded.
3. Store the initial database project in `project.Rdata` in the database folder.
4. Store the initial projcet in `initial_project.Rdata` in the project folder.

4.5.1 Parameters

- `initialise_database_project` (TRUE/FALSE): Creates the basic database project from the current ASCII files in the database folder and (if not existing) stores the `chemical_properties.csv` in the database folder.
- `initialise_project` (TRUE/FALSE): Stores the `REcoTox` environment in an initial `Rdata` object named `initial_project.Rdata` in the project folder.
- `load_default` (TRUE/FALSE): Loads an existing basic database project from the database folder and stores it in the project.

`chemical_properties.csv`: This tables contains the internal chemical ID `cas_number` (i.e. the cas number in integer format) and related user-curated metadata (e.g., chemical identifiers such as InChIKey, or PubChem CIDs) and chemical property data (i.e. log S values). It will be re-used and extended in future analyses to minimize curation efforts. If this file exists, it will be loaded to the project environment. Because the `chemicals.txt` table only contains `CAS` numbers in integer format, a regular CAS number is added (e.g., 1912-24-9 for 1912249).

```
project <- REcoTox::create_project(database_path = database_folder,
                           project_path,
                           initalise_database_project = TRUE,
                           initalise_project = TRUE,
                           load_default = FALSE)
)
```

4.6 Preparation of the initial project

In the second step, utilizing the function `prepare_data`, the tables' `test`, `results`, `species`, `chemicals`, and `references` are joined IDs `test_id`, `cas_number`, `species_number`, and `reference_number`. The initial environment is stored in the file `initial_project.Rdata` in the project folder. This file will be the same for all analyses related to a database revision. Thus, it could be copied from another project to avoid rerunning initial steps.

4.6.1 Parameters

- `project`: Name of the project environment.
- `load_initial_project` (TRUE/FALSE): Loads the `initial_project.Rdata` of the project folder.
- `new_project_path`: The `initial_project.Rdata` contains the project folder path where it was initially created. For example, in case of moving the project folder or if the `initial_project.Rdata` was copied from another folder, it is required to set a new project path.
- `save_project` (TRUE/FALSE): Save the `initial_project.Rdata`. For example, in case, the project folder was renewed.

```
project <- REcoTox::prepare_data(project = project,
                         load_initial_project = FALSE,
                         new_project_path = NA,
                         save_project = TRUE
)
```

4.7 Processing the data

In the third step, the function `process_data` reads the following settings to query the database accordingly.

A list of relevant endpoints (e.g., EC50) and all relevant species are generated and exported to the project folder for review. The two files are `ecotoxgroup_endpoint_selection.csv` and `ecotoxgroup_species_selection.csv`. The review could be performed in any spreadsheet program. The data must be stored in the `comma` delimited format!

The former table contains a field `include_endpoint`, this controls the inclusion of each endpoint by setting the value to 0 or 1 (0 = not included, 1 = included). Other values are not accepted, and the import of the file in the next processing step will be declined. The value 0 is the default, and thus, the endpoints to be included should be marked with 1.

The latter table contains a field include_species, this controls the inclusion of each species by setting the value to 0 or 1 (0 = not included, 1 = included). Depending on the settings of species_selection, the preset is different:

- `include_species` is set to 1
- `include_species` is set to 0
- `include_species` is set to 1 for standard test species and set to 0 for other species

Review and edit the tables in a preferred spreadsheet program. If changed, save the changes in the same file. The separator must be `comma`.

In this step, the database is queried to select the datasets related to the goals of the analysis.

The queries can be controlled by the following parameters:

- `dosing_group`: Specifies the compartment to which the dosing is referenced (so far only "water_concentration", i.e. result is mg/L)
- `duration_d`: Duration of the exposure in days (e.g. `d`, `dph`, `dpf`)
- `duration_h`: Duration of the exposure in hours (e.g. `h`, `ht`, `hph`, `hpf`, `hbf`, `hv`)

- `duration_m`: Duration of the exposure in minutes (e.g. `mi`)
- `ecotoxgroup`: Species group (e.g. `Algae`, `Crustacean`, `Fish`)
- `effects`: Effect endpoints (e.g. `MOR`, `GRO`, `DEV`)
- `habitat`: Habitat of the ecotoxgroup (i.e. `Non-Soil`, `Water`, `Soil`)
- `kingdoms`: Specification of the `algae` kingdoms (e.g. `Chromista`, `Plantae`, `Monera`)
- `measurements`: Specification of specific measurements
- `min_h`: Minimum duration of the experiment in hours
- `max_h`: Maximum duration of the experiment in hours
- `min_d`: Minimum duration of the experiment in days
- `max_d`: Maximum duration of the experiment in days
- `min_m`: Minimum duration of the experiment in minutes
- `max_m`: Maximum duration of the experiment in minutes
- `species_seleciton`: Selection of species (i.e. `all`, `manual`, `standard_test_species`)

Where `all` selects all species of an `ecotoxgroup`, `manual` expects manual selection in the files mentioned above and `standard_test_species` selects only species marked as standardized species.

4.7.1 Filtering the data

In the processing step 1, the data in the database is filtered based on the settings to extract relevant data of the database.

```
# set the parameters
dosing_group = "water_concentration" # i.e. mg/L (only available group in this version)
duration_d = c("d", "dph", "dpf")
duration_h = c("h", "ht", "hph", "hpf", "hbf", "hv")
duration_m = "mi"
ecotoxgroup = "Algae" # c("Algae", "Crustacean", "Fish")
effects = c("MOR", "GRO", "POP", "REP", "MPH", "DEV") # Algae/Fish
#effects = c("MOR", "GRO", "POP", "REP", "MPH", "DEV", "ITX") # Crustacean
habitat = "Water" #c("Non-Soil","Water","Soil")
kingdoms = NA # vector of specific algae kingdoms: c("Chromista","Plantae","Monera")
measurements = NA # vector of specific measurements
min_h = 0
min_d = 0
max_h = 120
max_d = 5
min_m = 0
max_m = 7200
species_selection = "all" # c("all", "manual", "standard_test_species")

# run the processing step
project <- REcoTox::process_data(project,
                         dosing_group = dosing_group,
                         duration_d = duration_h,
```

```
                        duration_h = duration_h,
                        duration_m = duration_m,
                        ecotoxgroup = ecotoxgroup,
                        effects = effects,
                        habitat = habitat,
                        kingdoms = kingdoms,
                        measurements = measurements,
                        max_d = max_d,
                        min_d = min_d,
                        max_h = max_h,
                        min_h = min_h,
                        max_m = max_m,
                        min_m = min_m,
                        remove_formulation = FALSE,
                        save_project_steps = FALSE,
                        species_selection = species_selection
)
```

This step stores two files in the `project_folder`, `ecotoxgroup_species_selection.csv` and `ecotoxgroup_endpoint_selection.csv`. The first block of the file is related to the ecotoxgroup specified. The species selection file contains all species extracted for review and the endpoint the respective endpoints (e.g. EC50). To include species or endpoints, mark the data with `1`, otherwise to exclude, mark with `0`.

4.7.2 Filtering species and endpoints

After review and saving the files, run the following command. This command reads the files and the data is filtered accordingly.

The units in the database are quite divergent and thus a unit conversion is performed to transform all units and values to `mg/L`. In case of mol related units, the transformation is automated so far the chemical and the molecular weight is already in the database. If not, the file `ecotoxgroup_mol_weight.csv` is exported to the `project_folder`.

```
project <- REcoTox::process_data(project, save_project_steps = FALSE)
```

4.7.3 Unit conversion

Review and edit the file `ecotoxgroup_mol_weight.csv` to add the molecular weight to the list. The ecotoxicity data is interactivitely enriched with chemical information (e.g. the average mass).

In best case with data linked to US EPA CompTox Chemicals Dashboard for example by using the output of the batch search according to Figure 1 and Figure 2.

After update of the mol weight table, run the following command to finalise the unit conversion step.

```
project <- REcoTox::process_data(project, save_project_steps = FALSE)
```

4.7.4 Chemical properties data and final processing

The former processing step creates a file named `ecotoxgroup_chemical_list.csv`. Edit this list to include newly added compounds (imputation of phys.-chem. properties and metadata).

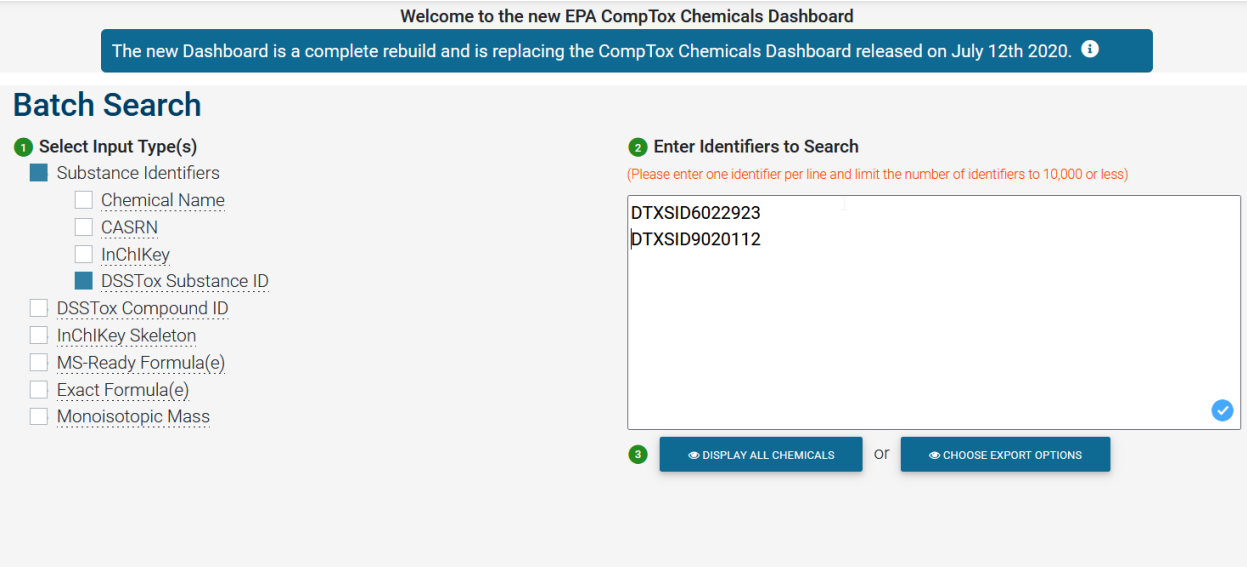

Figure 1: Figure1: US EPA CompTox Chemicals Dashboard Batch Search - Enter Identifiers to Search

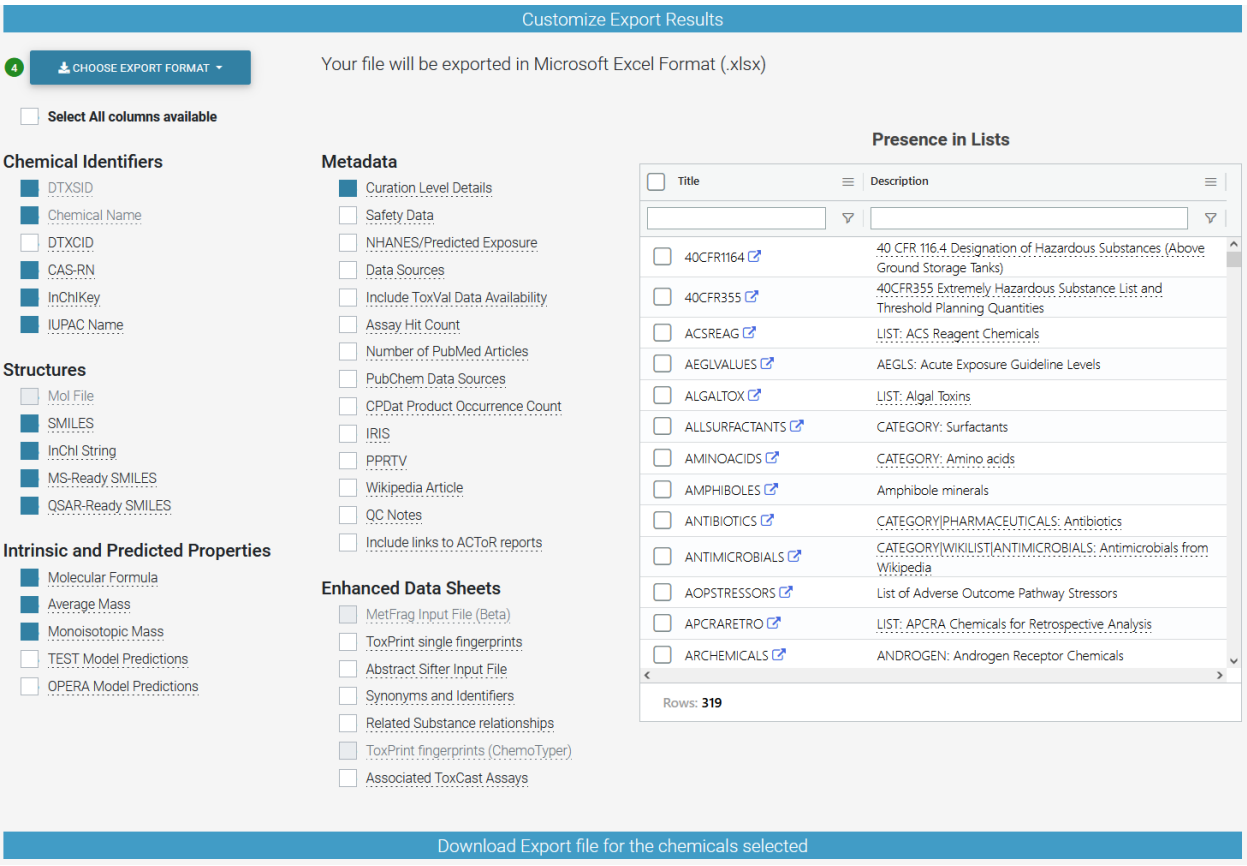

Figure 2: Figure 2: US EPA CompTox Chemicals Dashboard Batch Search - Recommended selection of identifiers and properties

To score the quality of the data, the solubility domain of the result is calculated. The calculation requires the experimental or predicted solubility of the chemical.

```
project <- REcoTox::process_data(project, save_project_steps = FALSE, update_chemicals = FALSE)
```

The file `ecotoxgroup_final_results.csv` is stored in the `project_folder`. It contains the results of the processing in the long pivot format.

4.8 Preparation of the wide pivot table with the aggregated ecotoxicity information

For final processing and to aggregate the data in the wide pivot format, run the following final step.

```
project <- REcoTox::aggregate_results(project = project, quantile = 0.05)
```

5 SessionInfo

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 23.10
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.23.so;  LAPACK version 3.11.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=C
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Berlin
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.3  forcats_1.0.0    stringr_1.5.0    dplyr_1.1.3
##  [5] purrr_1.0.2      readr_2.1.4      tidyr_1.3.0      tibble_3.2.1
##  [9] ggplot2_3.4.4    tidyverse_2.0.0  kableExtra_1.3.4 desc_1.4.2
## [13] BiocStyle_2.28.1 REcoTox_0.4.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0    viridisLite_0.4.2   fastmap_1.1.1
##  [4] promises_1.2.1      digest_0.6.33       timechange_0.2.0
##  [7] mime_0.12           lifecycle_1.0.3     ellipsis_0.3.2
## [10] processx_3.8.2      magrittr_2.0.3      compiler_4.3.1
## [13] rlang_1.1.1         sass_0.4.7          progress_1.2.2
## [16] tools_4.3.1         utf8_1.2.4          yaml_2.3.7
## [19] data.table_1.14.8   knitr_1.44          prettyunits_1.2.0
## [22] htmlwidgets_1.6.2   bit_4.0.5           pkgbuild_1.4.2
## [25] xml2_1.3.5          pkgload_1.3.3       miniUI_0.1.1.1
## [28] withr_2.5.1         grid_4.3.1          fansi_1.0.5
## [31] roxygen2_7.2.3      urlchecker_1.0.1    profvis_0.3.8
## [34] xtable_1.8-4        colorspace_2.1-0    data.tree_1.0.0
## [37] scales_1.2.1        cli_3.6.1           rmarkdown_2.25
## [40] crayon_1.5.2        generics_0.1.3      remotes_2.4.2.1
## [43] webchem_1.3.0       rstudioapi_0.15.0   httr_1.4.7
## [46] tzdb_0.4.0          sessioninfo_1.2.2   cachem_1.0.8
## [49] parallel_4.3.1      rvest_1.0.3         BiocManager_1.30.22
## [52] vctrs_0.6.4         devtools_2.4.5      webshot_0.5.5
## [55] jsonlite_1.8.7      bookdown_0.36       callr_3.7.3
```

```
## [58] hms_1.1.3           bit64_4.0.5         systemfonts_1.0.5
## [61] jquerylib_0.1.4     glue_1.6.2          ps_1.7.5
## [64] stringi_1.7.12      gtable_0.3.4        later_1.3.1
## [67] EnvStats_2.8.1      munsell_0.5.0       pillar_1.9.0
## [70] htmltools_0.5.6.1   R6_2.5.1            Rdpack_2.5
## [73] rprojroot_2.0.3     vroom_1.6.4         evaluate_0.22
## [76] shiny_1.7.5.1       highr_0.10          rbibutils_2.2.15
## [79] memoise_2.0.1       httpuv_1.6.12       bslib_0.5.1
## [82] Rcpp_1.0.11         svglite_2.1.2       xfun_0.40
## [85] fs_1.6.3            usethis_2.2.2       pkgconfig_2.0.3
```

References

1. LibreOffice - free office suite. https://www.libreoffice.org/.