Package: derappp (via r-universe)

July 3, 2026

Type Package

Title Data for Environmental Risk Assessment of Plant Protection Products

Version 0.9.2

Date 2026-07-02

Description Makes some public data for environmental risk assessment of plant protection products accessible in R. Includes definitions of substances in terms of well-defined chemical entities, if applicable. Also includes source references including page numbers, most often referring to secondary sources, as the original reports are usually unpublished. Uses standardised scientific names for biological species and includes physicochemical as well as ecotoxicological data. Data sources are mainly conclusions from the European pesticide risk assessment peer review process published by the European Food Safety Authority (EFSA) and Renewal Assessment Reports (RARs), also published by EFSA. Note that the use of the data included in this package for registration purposes may be restricted by regulatory data protection, also known as data exclusivity rules, as detailed for example in article 59 of Regulation (EC) No 1107/2009.

Depends R (>= 4.1.0), dm

Imports dplyr, tidyr, units, rlang, chents (>= 0.4.1), RefManageR, srppphist (>= 2.0.1)

Suggests devtools, knitr, rmarkdown, DiagrammeR (<= 1.0.11), DiagrammeRsvg, roxygen2 (>= 7.3.3.9000), quarto, pkgdown, testthat (>= 3.0.0), here, tibble, yaml, purrr, OpenFoodTox, lubridate, withr, rotl, readxl, jsonlite

BugReports https://github.com/agroscope-ch/derappp/issues

URL https://agroscope-ch.github.io/derappp/

Additional_repositories https://agroscope-ch.r-universe.dev, https://jranke.r-universe.dev

License GPL

LazyData true

LazyDataCompression gzip

Roxygen list(markdown = TRUE)

VignetteBuilder quarto

Encoding UTF-8

Language en-GB

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Config/pak/sysreqs
libicu-dev libpng-dev librsvg2-dev libxml2-dev libssl-dev python3 libudunits2-dev

Repository https://agroscope-ch.r-universe.dev

Date/Publication 2026-07-03 12:38:25 UTC

RemoteUrl https://github.com/agroscope-ch/derappp

RemoteRef HEAD

RemoteSha 6b1d80f273ac6d8fd3f4234603dc7bd449dcc362

Contents

`derappp` *Data for environmental risk assessment of plant protection products*

Description

Data for environmental risk assessment of plant protection products

Usage

```
derappp
```

Format

A number of tables collected in a dm::dm object

Examples

```
library(derappp) # This also loads the dm package
dm_draw(derappp)

# The list of chemical entities ("chents")
derappp$chents

# Some vapor pressures and water solubilities
library(units)
derappp$p0[1:2, ]
derappp$p0[1, ] |>
  left_join(derappp$sources, by = "sk") |>
  select(substance, p0, T, reference)

derappp$cwsat[1:3, ] |>
  left_join(derappp$sources, by = "sk") |>
  select(substance, cwsat, T, pH, reference) |>
  mutate(cwsat = set_units(cwsat, "g/L", mode = "standard"))

# Join names used in the Swiss register
derappp$chents |>
  left_join(derappp$substance_keys, by = c(chent = "substance")) |>
  filter(db == "srppp") |>
  left_join(srppphist::srppp_active_substances, by = c(key = "pk")) |>
  select(chent, iso, smiles, substance_de)

# Show some soil sorption data with units
derappp$soil_sorption |>
  filter(substance %in% c("Acetamiprid", "Captan", "Copper", "Cyprodinil")) |>
  select(substance, soil_pH, f_oc, Koc, Kfoc, n, sk, selected, reason) |>
  print(n = Inf)

# Show some soil degradation data with units
derappp$soil_degradation |>
  select(substance, DT50, kinetics, alpha, beta, k1, k2, g, tb, sk) |>
  print(n = 10)

# Some aquatic toxicity data with units
head(derappp$aquatic_toxicity) |>
  select(substance, formulation, derappp_species, duration, effect, level, sign, value)

# Species groupings and taxonomic IDs
derappp$species
```

`derappp_bib` *List of references*

Description

List of references

Usage

```
derappp_bib
```

Format

A list of RefManageR::BibEntry objects

Examples

```
derappp_bib[1:10]
```

`derappp_chents` *List of chemical entities with additional information*

Description

List of chemical entities with additional information

Usage

```
derappp_chents

## S3 method for class 'derappp_chents'
print(x, n = 3, ...)
```

Arguments

`x`	A list of chents::chent objects, with class `derappp_chents`
`n`	Number of entries to show
`...`	For compatibility with the generic

Format

A list of chents::chent objects, with class `derappp_chents`

Examples

```
# The chemical entities are stored as a list of chent objects, so we can
# access them by name
derappp_chents$Cyprodinil

derappp_chents$Myclobutanil

# The IUPAC name and source URL retrieved from the British Crop Protection
# Council (BCPC) are stored as fields in the
# `bcpc` element of the object, so we can easily access them
derappp_chents$Captan$bcpc[c("iupac_name", "source_url")]
```

```
# The PubChem information is stored as a list, so we can check which fields are available
names(derappp_chents$Captan$pubchem)

# For example, we can check the molecular formula
derappp_chents$Captan$pubchem$MolecularFormula

# We also have a print method for the complete object showing the first few
# items
print(derappp_chents, n = 2)
```

`loep` *Open an EFSA list of endpoints for a substance*

Description

For this to work as intended, you need to specify an environment variable `_derappp_sources_` that points to the directory where the EFSA journal PDF files and other sources are stored.

Usage

```
loep(string, open = interactive())
```

Arguments

`string`	A part of a substance name
`open`	Should the file be opened using utils::browseURL if present?

Details

If more than one substance matches the string given, the user can interactively select one of them. If more than one EFSA conclusion is linked to the substance, the user can again select one. If there is a separate list of endpoints (EFSA conclusions starting from generally have one) and it is available on the path, it is opened instead of the main EFSA conclusion.

Value

If successful, a path to a list of endpoints (invisible)

Examples

```
# The function only works if the environment variable _derappp_sources_ has
# the path to a directory with the EFSA journal pdf files.
if (Sys.getenv("_derappp_sources_") != "") {
  loep("Difluf")
  loep("Diflubenzuron")
}
```

`translate_substances` *Translate substance names between namespaces*

Description

This function is **experimental** and its interface may change in future versions without notice.

Usage

```
translate_substances(x, ..., from, to)

## S3 method for class 'character'
translate_substances(x, substances = x, ..., from, to)

## S3 method for class 'data.frame'
translate_substances(x, ..., from, to)
```

Arguments

`x`	Either a character vector (see substances) or a table containing a character vector If it is a table (a data.frame or a table derived from a data.frame such as a tibble), there must be a column named according to the "from" argument described below.
`...`	Currently not used
`from`	Namespace of the input, defaulting to "derappp"
`to`	Desired namespace of the output, defaulting to "derappp"
`substances`	A character vector of substance names or primary keys referring to one or more substance(s)

Value

A tibble with a column for each namespace, named with the namespaces involved.

Examples

```
# Check which namespaces currently exist
unique(derappp$substance_keys$db)

# Translate substance primary keys from the Swiss Register of Plant Protection Products
srppp_names <- c("1-Naphthylacetic acid", "Terbuthylazine", "Pyrethrine")
srppp_pk <- c("3", "1245", "323")

# Then do the actual translation with derappp::translate_substances()
translation <- translate_substances(srppp_pk, from = "srppp")
print(translation)
translate_substances(srppp_names, from = "substance_de")
```

```
# We get warned if a substance_de is not mapped and/or translation is incomplete
translate_substances(c(srppp_names, "Glyphosate", "Kupfer"), from = "substance_de")

# There is also a method for data frames preserving all columns
input <- data.frame(x = 1:3, srppp = srppp_pk)
translate_substances(input, from = "srppp")
input_de <- data.frame(x = 1:3, substance_de = srppp_names)
translate_substances(input_de, from = "substance_de")

# We can also translate back and get multiple matches for some names
translate_substances(translation$derappp, to = "srppp")

# Or translate to another namespaces
translate_substances(srppp_pk, from = "srppp", to = "NABO_SQ")

# An example with NABO Status Quo substances
translate_substances(c("Chlorothalonil R417888", "S-Metolachlor"),
  from = "NABO_SQ")

# If we translate to a namespace that does not have a mapping for the
# substance, we get NA
translate_substances(c("Chlorothalonil R417888", "S-Metolachlor", "Diazinone"),
  from = "NABO_SQ", to = "srppp")

# We only find exact matches (room for a future extension)
translate_substances(c("S-Metolachlor", "Glyphosat", "Glyphosate"),
  from = "derappp", to = "srppp")
```

---

wlt *Calculate a weighted laboratory toxicity (WLT)*

Description

This is an implementation of the method used for deriving WLT values in the Swiss National Pesticide Risk Indicator project (Korkaric et al. 2020, 2022 and 2023). In the original project, the method was applied manually. This method facilitates the derivation of updated WLT values for substances for which the relevant data has been integrated into the derappp package.

Usage

```
wlt(
  substance,
  medium = c("surface water"),
  smaller_than = c("warn", "keep", "ignore"),
  formulations = c("include", "include-unique", "exclude"),
  salt_water = c("include", "special", "exclude"),
  max_AF_salt = Inf
)
```

Arguments

substance	A substance name
medium	The medium for which the assessment is made
smaller_than	How to handle "smaller than" values (e.g. an endpoint specified as < 2 mg/L). Default is to warn if such an endpoint is in the endpoints specified as 'selected' in the input data, and belonging to the 'preferred' endpoints as determined based on endpoint level and exposure duration by this function.
formulations	Per default, all formulation data are included. When this argument is set to "include-unique", only data on formulations for which no comparable (based on species, duration and effect level), but ultimately based on expert judgement) endpoint is available are included. If set to "exclude", data from formulations are ignored. If set to "manual", formulation data to include can be specified using the argument "include".
salt_water	Per default, salt water species are included in the assessment for surface water. If this argument is set to "special", endpoints from salt water species receive a special treatment (see details). If set to "exclude", data from an internal list with species not living in freshwater are ignored.
max_AF_salt	Maximum factor to increase the assessment factor based on the comparison of salt water to freshwater species

Details

Currently, only surface water WLT values can be derived, and only the aquatic toxicity table in this package is supported as data source.

Default assessment factors (AF) are as in the EU regulation describing the Uniform Principles of Risk Assessment (EC 2011), and as detailed in the EFSA aquatic risk assessment guidance (EFSA 2013).

The special treatment of salt water species is based on the ratio between salt water and freshwater organisms of the lowest endpoints within each taxonomic group and separately for short-term and long-term data.

If this ratio is greater than 10, the AF of the freshwater species in the group is increased by one tenth of this ratio, up to an optional maximum factor.

WLT candidates are determined by dividing the preferred endpoint for each test by the appropriate (AF) which is equal to the TER trigger value defined in the Uniform Principles (EC 2011), also taking into account the details and additions (e.g. for macrophytes) as defined in the EFSA guidance (EFSA 2013).

References

EC (2011) COMMISSION REGULATION (EU) No 546/2011 Annex Part A 2.5.2.2.

EFSA (2013) Guidance on tiered risk assessment for plant protection products for aquatic organisms in edge-of-field surface waters, doi:10.2903/j.efsa.2013.3290

Korkaric M, Hanke I, Grossar D, Neuweiler R, Christ B, Wirth J, Hochstrasser M, Dubuis PH, Kuster T, Breitenmoser S, Egger B, Perren S, Schürch S, Aldrich A, Jeker L, Poiger T, Daniel O (2020) Datengrundlage und Kriterien für eine Einschränkung der PSM-Auswahl im ÖLN: Schutz

der Oberflächengewässer, der Bienen und des Grundwassers (Metaboliten), sowie agronomische Folgen der Einschränkungen. Agroscope Science, 106, 2020, 1-31. doi:10.34776/as106g

Korkaric M, Ammann L, Hanke I, Schneuwly J, Lehto M, Poiger T, de Baan L, Daniel O, Blom JF (2022) Neue Pflanzenschutzmittel-Risikoindikatoren für die Schweiz. Agrarforschung Schweiz 13, 1-10, doi:10.34776/afs131

Korkaric M, Lehto M, Poiger T, de Baan L, Mathis M, Ammann L, Hanke I, Balmer M, Blom JF (2023) Risikoindikatoren für Pflanzenschutzmittel: weiterführende Analysen zur Berechnung. Agroscope Science, 154, 1-48, doi:10.34776/as154g

Examples

```
f_wlt <- wlt("Flutolanil")
f_wlt$wlt
f_wlt$ratios_salt_nosalt
f_wlt$data

# Repeat the analysis without the special treatment for salt water organisms
f_wlt_special <- wlt("Flutolanil", salt_water = "special")
f_wlt_special$wlt
```

Index