Package: srppp (via r-universe)

August 13, 2024

Type Package

Title The Swiss Register of Plant Protection Products as an R Package

Version 0.99.3

Date 2024-08-13

Description Functions to generate data objects from XML versions of the Swiss Register of Plant Protection Products 'SRPPP'. An online version of the register can be accessed at <https://www.psm.admin.ch/de/produkte>. There is no guarantee of correspondence of the data read in using this package with that online version, or with the original registration documents. Also, the Federal Food Safety and Veterinary Office, coordinating the authorisation of plant protection products in Switzerland, does not answer requests regarding this package.

Depends R (>= 4.1.0), dm

Imports xml2, tibble, stringr, dplyr, tidyr, cli

Suggests knitr, rmarkdown, here, DiagrammeR, testthat (>= 3.0.0), parallel, waldo

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

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

License GPL (>= 3)

LazyData yes

LazyDataCompression xz

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

Config/testthat/edition 3

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

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

RemoteRef HEAD

RemoteSha e76660437e5dd8e65468043143f0453d881dfa5f

Contents

`alternative_products` *Find alternative products for all products containing certain active substances*

Description

This function searches for uses of a given list of active substances and reports either a table of uses with the number of available alternative products for each use, a detailed table of the alternative product uses, a table of uses without alternatives, or a list containing these three tables.

Usage

```
alternative_products(
  srppp,
  active_ingredients,
  details = FALSE,
  missing = FALSE,
  list = FALSE,
  lang = c("de", "fr", "it")
)
```

Arguments

srppp	A srppp_dm object.
active_ingredients	Character vector of active ingredient names that will be matched against the column 'substances_de' in the srppp table 'substances'.
details	Should a table of alternative uses with 'wNbr' 'use_nr' be returned?
missing	If this is set to TRUE, uses without alternative product registrations are listed.
list	If TRUE, a list of three tables is returned, a table of uses without alternative products ("Lückenindikationen"), a table of the number of alternative products for each use, if any, and a detailed table of all the alternative uses. This argument overrides the arguments 'details' and 'missing'.
lang	The language used for the active ingredient names and the returned tables.

Details

A use is defined as a combination of an application area, a crop ('culture') and a pathogen ('pest').

Examples

```
## Not run:
sr <- srppp_dm()

actives_de <- c("Lambda-Cyhalothrin", "Deltamethrin")

alternative_products(sr, actives_de)
alternative_products(sr, actives_de, missing = TRUE)
alternative_products(sr, actives_de, details = TRUE)
alternative_products(sr, actives_de, list = TRUE)

# Example in Italian
actives_it <- c("Lambda-Cialotrina", "Deltametrina")
alternative_products(sr, actives_it, lang = "it")

## End(Not run)
```

---

application_rate_g_per_ha

Calculate application rates for active ingredients

Description

An application rate in g active substance/ha is calculated from information on dosage (product concentration in the application solution), application volume, or directly from the product application rate. This is complicated by the fact that a rate ("expenditure" in the XML file) with units l/ha can refer to the application solution or to the liquid product.

Usage

```
application_rate_g_per_ha(
  product_uses,
  aggregation = c("max", "mean", "min"),
  dosage_units = c("percent_ww", "percent_vv", "state_of_matter"),
  skip_l_per_ha_without_g_per_L = TRUE,
  fix_l_per_ha = TRUE
)
```

Arguments

`product_uses`	A tibble containing the columns 'pNbr', 'use_nr', 'application_area_de', 'min_dosage', 'max_dosage', 'min_rate', 'max_rate', from the 'uses' table in a srppp_dm object, as well as the columns 'percent' and 'g_per_L' from the 'ingredients' table in a srppp_dm object.
`aggregation`	How to represent a range if present, e.g. "max" (default) or "mean".
`dosage_units`	If no units are given, or units are "%", then the applied amount in g/ha is calculated using a reference application volume and the dosage. As the dosage units are not explicitly given, we can specify our assumptions about these using this argument (currently not implemented, i.e. specifying the argument has no effect).
`skip_l_per_ha_without_g_per_L`	Per default, uses where the use rate has units of l/ha are skipped, if there is not product concentration in g/L. This was also done in the 2023 indicator project.
`fix_l_per_ha`	During the review of the 2023 indicator project calculations, a number of cases were identified where the unit l/ha specifies a water volume, and not a product volume. If TRUE (default), these cases are corrected, if FALSE, these cases are discarded.

Details

In some cases (currently one), external information was found, indicating that the "expenditure" is an application volume l_per_ha_is_water_volume.

Value

A tibble containing one additional column 'rate_g_per_ha'

Note

A reference application volume is used if there is no 'expenditure'. It is selected only based on the product application area. This is not correct if hops ('Hopfen') is the culture, as it has a unique reference application volume of 3000 L/ha.

Applications to hops were excluded for calculating mean use rates in the indicator project (Korkaric 2023), arguing that it is not grown in large areas in Switzerland.

Examples

```
## Not run:
library(srppp)
library(dplyr, warn.conflicts = FALSE)
library(dm, warn.conflicts = FALSE)
sr <- srppp_dm()

product_uses_with_ingredients <- sr$substances |>
  filter(substance_de %in% c("Halauxifen-methyl", "Kupfer (als Kalkpr\u00E4parat)")) |>
  left_join(sr$ingredients, by = "pk") |>
  left_join(sr$uses, by = "pNbr") |>
  left_join(sr$products, by = "pNbr") |>
  select(pNbr, name, use_nr,
    min_dosage, max_dosage, min_rate, max_rate, units_de,
    application_area_de,
    substance_de, percent, g_per_L)

application_rate_g_per_ha(product_uses_with_ingredients) |>
  filter(name %in% c("Cerelex", "Pixxaro EC", "Bordeaux S")) |>
  select(ai = substance_de, app_area = application_area_de,
    min_d = min_dosage,  max_d = max_dosage,
    min_r = min_rate, max_r = max_rate,
    units_de, rate = rate_g_per_ha) |>
  print(n = Inf)

## End(Not run)
```

`l_per_ha_is_water_volume` *Use definitions where the rate in l/ha refers to the volume of the spraying solution*

Description

Use definitions where the rate in l/ha refers to the volume of the spraying solution

Usage

```
l_per_ha_is_water_volume
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 4 columns.

See Also

application_rate_g_per_ha

Examples

```
library(srppp)
l_per_ha_is_water_volume
```

`srppp_dm` *Create a dm object from an XML version of the Swiss Register of Plant Protection Products*

Description

While reading in the data, the information obtained from the XML file is left unchanged, with the exceptions listed in the section 'Details'. An overview of the contents of the most important tables in the resulting data object is given in `vignette("srppp")`.

Usage

```
srppp_dm(from = srppp_xml_url, remove_duplicates = TRUE)

## S3 method for class 'srppp_dm'
print(x, ...)
```

Arguments

`from`	A specification of the way to retrieve the XML
`remove_duplicates`	Should duplicates based on wNbrs be removed?
`x`	A srppp_dm object
`...`	Not used

Details

Corrections made to the data:

- In the following case, the product composition is corrected while reading in the data: The active substance content of Dormex (W-3066) is not 667 g/L, but 520 g/L This was confirmed by a visit to the Wädenswil archive by Johannes Ranke and Daniel Baumgartner, 2024-03-27.

Removal of redundant information:

- Information on products that has been duplicated across several products sharing the same P-Number has been associated directly with this P-Number, in order to avoid duplications. While reading in the XML file, it is checked that the resulting deduplication does not remove any data.
- In very few cases of historical XML files, there are two `<Product>` sections sharing the same W-Number. In these cases, one of these has apparently been included in error and an informed decision is taken while reading in the data which one of these sections is discarded. The details of this procedure can be found in the source code of the function `srppp_xml_get_products`.

Amendments to the data:

In the table of obligations, the following information on mitigation measures is extracted from the ones relevant for the environment (SPe 3).

- "sw_drift_dist": Unsprayed buffer towards surface waters to mitigate spray drift in meters
- "sw_runoff_dist": Vegetated buffer towards surface waters to mitigate runoff in meters
- "sw_runoff_points": Required runoff mitigation points to mitigate runoff
- "biotope_drift_dist": Unsprayed buffer towards biotopes (as defined in articles 18a and 18b of the Federal Act on the Protection of Nature and Cultural Heritage) to mitigate spray drift in meters

Value

A dm::dm object with tables linked by foreign keys pointing to primary keys, i.e. with referential integrity.

Examples

```
library(dplyr, warn.conflicts = FALSE)
library(dm, warn.conflicts = FALSE)

sr <- srppp_dm()
dm_examine_constraints(sr)
## Not run:
dm_draw(sr)


## End(Not run)

# Show ingredients for products named 'Boxer'
sr$products |>
  filter(name == "Boxer") |>
  left_join(sr$ingredients, by = "pNbr") |>
  left_join(sr$substances, by = "pk") |>
  select(wNbr, name, pNbr, isSalePermission, substance_de, g_per_L)

# Show authorised uses of the original product
boxer_uses <- sr$products |>
  filter(name == "Boxer", !isSalePermission) |>
  left_join(sr$uses, by = "pNbr") |>
  select(pNbr, use_nr,
    min_dosage, max_dosage, min_rate, max_rate, units_de,
    waiting_period, time_units_de, application_area_de)
print(boxer_uses)

# Show crop for use number 1
boxer_uses |>
  filter(use_nr == 1) |>
  left_join(sr$cultures, join_by(pNbr, use_nr)) |>
  select(use_nr, culture_de)

# Show target pests for use number 1
```

```
boxer_uses |>
  filter(use_nr == 1) |>
  left_join(sr$pests, join_by(pNbr, use_nr)) |>
  select(use_nr, pest_de)

# Show obligations for use number 1
boxer_uses |>
  filter(use_nr == 1) |>
  left_join(sr$obligations, join_by(pNbr, use_nr)) |>
  select(use_nr, sw_runoff_points, obligation_de) |>
  knitr::kable() |>
  print()

# Show application comments for use number 1
boxer_uses |>
  filter(use_nr == 1) |>
  left_join(sr$application_comments, join_by(pNbr, use_nr)) |>
  select(use_nr, application_comment_de)
```

`srppp_xml_clean_product_names` *Clean product names*

Description

Clean product names

Usage

```
srppp_xml_clean_product_names(names)
```

Arguments

`names`	The product names that should be cleaned from comments

`srppp_xml_define_use_numbers` *Define use identification numbers in an SRPPP read in from an XML file*

Description

Define use identification numbers in an SRPPP read in from an XML file

Usage

```
srppp_xml_define_use_numbers(srppp_xml = srppp_xml_get())
```

Arguments

srppp_xml	An object as returned by 'srppp_xml_get'

Value

An srppp_xml object with use_nr added as an attribute of 'Indication' nodes.

Examples

```
srppp_xml_define_use_numbers()
```

---

srppp_xml_get *Read an XML version of the Swiss Register of Plant Protection Products*

Description

Read an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get(from, ...)

## S3 method for class '`NULL`'
srppp_xml_get(from, ...)

## S3 method for class 'character'
srppp_xml_get(from, ...)

srppp_xml_get_from_path(path, from)
```

Arguments

from	A specification of the way to retrieve the XML
...	Unused argument introduced to facilitate future extensions
path	A path to a zipped SRPPP XML file

Value

An object inheriting from 'srppp_xml', 'xml_document', 'xml_node'

Examples

```
# The current SRPPP as available from the FOAG website
srppp_cur <- srppp_xml_get()
# The current SRPPP as available from the FOAG website
srppp_cur <- srppp_xml_get(srppp_xml_url)
```

`srppp_xml_get_ingredients` *Get ingredients for all products described in an XML version of the Swiss Register of Plant Protection Products*

Description

Get ingredients for all products described in an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get_ingredients(srppp_xml = srppp_xml_get())
```

Arguments

`srppp_xml`	An object as returned by 'srppp_xml_get'

Examples

```
srppp_xml_get_ingredients()
```

`srppp_xml_get_parallel_imports` *Get Parallel Imports from an XML version of the Swiss Register of Plant Protection Products*

Description

Get Parallel Imports from an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get_parallel_imports(srppp_xml = srppp_xml_get())
```

Arguments

`srppp_xml`	An object as returned by 'srppp_xml_get'

Value

A tibble::tibble with a row for each parallel import section in the XML file.

Examples

```
# Get current list of parallel_imports
srppp_xml_get_parallel_imports()
```

srppp_xml_get_products *Get Products from an XML version of the Swiss Register of Plant Protection Products*

Description

Get Products from an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get_products(
  srppp_xml = srppp_xml_get(),
  verbose = TRUE,
  remove_duplicates = TRUE
)
```

Arguments

srppp_xml	An object as returned by 'srppp_xml_get'
verbose	Should we give some feedback?
remove_duplicates	Should duplicates based on wNbrs be removed? If set to 'TRUE', one of the two entries with identical wNbrs is removed, based on an investigation of background information carried out by the package authors. In all cases except for one, one of the product sections with duplicate wNbrs has information about an expiry of the registration, and the other doesn't. In these cases the registration without expiry is kept, and the expiring registration is discarded. In the remaining case (wNbr 5945), the second entry is selected, as it contains more indications which were apparently intended to be published as well.

Value

A tibble::tibble with a row for each product section in the XML file. An attribute 'duplicated_wNbrs' is also returned, containing duplicated W-Numbers, if applicable, or NULL.

Examples

```
# Get current list of products
srppp_xml_get_products()
```

srppp_xml_get_substances

Get substances from an XML version of the Swiss Register of Plant Protection Products

Description

Get substances from an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get_substances(srppp_xml = srppp_xml_get())
```

Arguments

srppp_xml	An object as returned by 'srppp_xml_get'

Examples

```
srppp_xml_get_substances()
```

srppp_xml_get_uses

Get uses ('indications') for all products described in an XML version of the Swiss Register of Plant Protection Products

Description

Get uses ('indications') for all products described in an XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_get_uses(srppp_xml = srppp_xml_get())
```

Arguments

srppp_xml	An object as returned by srppp_xml_get with use numbers defined by srppp_xml_define_use_numbers

Examples

```
srppp_xml <- srppp_xml_get()
srppp_xml <- srppp_xml_define_use_numbers(srppp_xml)
srppp_xml_get_uses(srppp_xml)
```

srppp_xml_url *URL of the XML version of the Swiss Register of Plant Protection Products*

Description

URL of the XML version of the Swiss Register of Plant Protection Products

Usage

```
srppp_xml_url
```

Format

length one character string

Examples

```
print(srppp_xml_url)
```

units_convertible_to_g_per_ha *Product application rate units convertible to grams active substance per hectare*

Description

Product application rate units convertible to grams active substance per hectare

Usage

```
units_convertible_to_g_per_ha
```

Format

An object of class `character` of length 7.

See Also

application_rate_g_per_ha

Examples

```
library(srppp)
library(dplyr)
# These are the convertible units
units_convertible_to_g_per_ha
```

Index