

# Package ‘PesticideLoadIndicator’

January 20, 2025

**Title** Computes Danish Pesticide Load Indicator

**Version** 1.3.1

**Description** Computes the Danish Pesticide Load Indicator as described in Kudsk et al. (2018) <[doi:10.1016/j.landusepol.2017.11.010](https://doi.org/10.1016/j.landusepol.2017.11.010)> and Moehring et al. (2019) <[doi:10.1016/j.scitotenv.2019.04.040](https://doi.org/10.1016/j.scitotenv.2019.04.040)>. Additionally offers the possibility to directly link pesticide use data to pesticide properties given access to the Pesticide properties database (Lewis et al., 2016) <[doi:10.1080/10807039.2015.1133242](https://doi.org/10.1080/10807039.2015.1133242)>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**Suggests** testthat, knitr, rmarkdown

**Imports** readxl, stringr, magrittr, rlang, dplyr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Niklas Moehring [aut, cre] (<<https://orcid.org/0000-0003-0292-4402>>),  
Leonie Vidensky [aut],  
Robert Finger [aut],  
Per Kudsk [aut],  
Lise Nistrup Jørgensen [aut],  
Jens Erik Ørum [aut],  
Uwe Schmitt [ctb]

**Maintainer** Niklas Moehring <[niklas.moehring@mtec.ethz.ch](mailto:niklas.moehring@mtec.ethz.ch)>

**Repository** CRAN

**Date/Publication** 2022-10-24 16:27:57 UTC

## Contents

check_products_column_names . . . . .	2
check_substance_column_names . . . . .	2
compute_pesticide_load_indicator . . . . .	3
compute_risk_score . . . . .	5

default.load.factors . . . . .	5
match.ppdb . . . . .	6
products.load . . . . .	6
products.path . . . . .	7
required_columns_products . . . . .	7
substances.load . . . . .	8
substances.path . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

check\_products\_column\_names  
*Check if column names of applied pesticide products dataframe are valid*

---

### Description

checks for valid column names and stops execution if problems are detected

### Usage

check\_products\_column\_names(products)

### Arguments

products            Dataframe with raw pesticide application data.

### Value

No return value

---

check\_substance\_column\_names  
*Check if column names of substances dataframe are valid*

---

### Description

checks for valid column names and stops execution if problems are detected

### Usage

check\_substance\_column\_names(substances)

### Arguments

substances            Dataframe describing active ingredients of the applied pesticide products.

**Value**

No return value

---

compute\_pesticide\_load\_indicator

*Compute Pesticide Load Indicator with user supplied information on pesticide properties*

---

**Description**

Compute Pesticide Load Indicator with user supplied information on pesticide properties

**Usage**

```
compute_pesticide_load_indicator(substances, products)
```

**Arguments**

substances	Dataframe describing active ingredients of the applied pesticide products, including their ecotoxicity, fate and human health properties.
products	Dataframe with raw pesticide application data.

**Value**

Dataframe with pesticide indicators for each pesticide application indicated in the products dataframe. Computes Pesticide Load Indicator (L) and its subindicators: The Human Health Load (HL), Ecotoxicity Load (TL) and Fate Load (FL). If standard dosages are provided the Standard Treatment Index (STI) and the Pesticide Load Index (LI=STI\*L) are also computed.

**Examples**

```
## Not run:
# A) Compute Pesticide Load indicator for a complete database

# load the dataframe containing the pesticide use data.
products_user <- products.load()
# load the (user-supplied) dataframe with detailed information on used pesticides.
substances_user <- substances.load()

# Compute the Pesticide Load Indicator and its sub-indicators using the user supplied data.
indicators_user <- compute_pesticide_load_indicator(substances = substances_user,
products= products_user)

# B) Compute Pesticide Load Indicator starting from basic data on products used
# Add properties of pesticides with match_ppdb()

# Step1: load the dataframe containing the basic pesticide use data.
products_basic <- products.load()[,c("product", "crop", "standard.dosage", "formula")]
```

```

# Step2: Add information on the year in which the product is used.
# (not necessary if all data is before 2013 - then just insert a dummy year > 2013)
products_basic$Year <- c(2009,2010,2011,2012)

# Step3: load the (user-supplied) dataframe with basic information on used pesticides
substances_basic <- substances.load()[,c("substance","product","concentration")]

# Step4: Add the CAS number of each active ingredient to link to the Pesticide Properties database.
substances_basic$CAS.number <- c("94361-06-5","141517-21-7","111988-49-9","467-69-6",
                                "1918-00-9","94-74-6","21087-64-9","142459-58-3")

# Step5: Add the Load factors as defined in the Danish Pesticide Load indicator.
# These values are supplied in the package.
# Alternatively supply own values for the Load factor.
Load.factors <- c("Load.Factor.SCI","Load.Factor.BCF","Load.Factor.SoilDT50",
                 "Load.Factor.Birds","Load.Factor.Mammals","Load.Factor.Fish",
                 "Load.Factor.Aquatic.Invertebrates","Load.Factor.Algae","Load.Factor.Aquatic.Plants",
                 "Load.Factor.Earthworms","Load.Factor.Bees","Load.Factor.Fish.Chronic",
                 "Load.Factor.Aquatic.Invertebrates.Chronic","Load.Factor.Earthworms.Chronic")
for (i in 1:length(Load.factors)){
  substances_basic[,Load.factors[i]]<- rep(times=dim(substances_basic)[[1]],
                                           substances.load()[1,Load.factors[i]])
}

# Step6: Add pesticide properties from the PPDB using the match_ppdb() function
# Indicate the folder containing the "General","Fate", "Human" and "Ecotox" tables of the PPPDB.
# Excel files (under the exact same name, e.g. Human.xlsx) are required.
# Attention, a licensed access to the PPPDB (Lewis et al., 2016) is required.
# Note that the "Fate" table should include a column indicating the "SCI.GROW" values.
folder <- "path"
matched_data <- match_ppdb (substances=substances_basic, products=products_basic,
                           folder=folder, healthrisk_off=TRUE)

products_complete<- matched_data[[1]]
substances_complete<- matched_data[[2]]

# Step7(optional): change reference values in the substances_complete data_frame if required.

# Step8: Supply the sum of risk scores based on the product label to compute the Human Health Load.
# Add the reference value for the Human Health Load
products_complete$sum.risk.score <- c(150,25,20,130)
products_complete$reference.sum.risk.scores <- 350

# Step9: Compute the Pesticide Load Indicator and its sub-indicators
indicators_user <- compute_pesticide_load_indicator(substances = substances_complete,
products= products_complete)
# Note that from version 1.3.1 on, the optional Pesticide Load Index
# is computed by first standardizing the Pesticide Load with the standard dosage.

## End(Not run)

```

---

compute_risk_score	<i>Calculates the sum of risk scores from a list of H-phrases and expands the table</i>
--------------------	---

---

**Description**

Calculates the sum of risk scores from a list of H-phrases and expands the table

**Usage**

```
compute_risk_score(table, var_name)
```

**Arguments**

table	Dataframe with H-phrases on product level.
var_name	Name of the variable that contains the information (string) on H-phrases. For example "H317; H411" or "H410, H411".

**Value**

Adds a variable indicating the sum of risk scores needed to compute the Pesticide Load Indicator. Check national pesticide databases for information on product labels of pesticides (information on H-phrases of each product).

---

default.load.factors	<i>Default load factors</i>
----------------------	-----------------------------

---

**Description**

Default load factors

**Usage**

```
default.load.factors
```

**Format**

An object of class list of length 14.

---

match.ppdb	<i>Expend tables with information on ecotoxicity, fate (and human health) properties from PPDB</i>
------------	--

---

**Description**

Expend tables with information on ecotoxicity, fate (and human health) properties from PPDB

**Usage**

```
match.ppdb(substances, products, folder, healthrisk_off = TRUE)
```

**Arguments**

substances	Dataframe describing active ingredients of the applied pesticide products and their CAS number.
products	Dataframe with raw pesticide application data.
folder	Folder with exported xlsx files from PPDB containing information on active ingredient properties.
healthrisk_off	Compute the Human Health risk sum score from the PPDb (default off).

**Value**

Adds Ecotoxicity and Fate properties of active substances needed to compute the Pesticide Load Indicator to user-provided substance and product data frames. Properties are based on information from the Pesticide Properties Database (PPDB), which has to be provided by the user in Excel format (license required). Note that the function can optionally also retrieve the sum of risk scores for Human Health from the PPDB, based on active ingredient-level risk phrases in the PPDB. This is not recommended. Best practice is to compute the sum of risk scores based on risk phrases of the respective pesticide product (see Kudsk et al., 2018 for weighing of respective risk phrases). Product label information cannot be retrieved from the PPDB as labels might be country-specific. Check national pesticide databases for this information. Note that you have to add the reference value for sum.risk.scores, as follows, if you select healthrisk\_off=TRUE: `products$reference.sum.risk.scores <- 300`

---

products.load	<i>load included products.xlsx file</i>
---------------	---

---

**Description**

load included products.xlsx file

**Usage**

```
products.load()
```

**Value**

products.xlsx file as data.frame

---

products.path                    *path to includedexamples products.xlsx file*

---

**Description**

path to includedexamples products.xlsx file

**Usage**

products.path()

**Value**

path to products.xlsx file

---

required\_columns\_products

*Computing the Pesticide Load Indicator for pesticide application data*  
*The provided functions will compute the Pesticide Load Indicator (PLI) as described in Kudsk et al. (2018) for pesticide application data provided by the user. Computing the PLI requires information on applied pesticides in a table format, as well as information on fate, ecotoxicity and human health properties of applied pesticide products, as provided in the Pesticide Properties Database (PPDB) of the University of Hertfordshire. See below for a detailed description. The PLI can either be computed using user supplied information on pesticide properties or by automatically including the information based on the PPDB. Access to the PPDb requires a license - see <http://sitem.herts.ac.uk/aeru/ppdb/>.*

---

**Description**

Computing the Pesticide Load Indicator for pesticide application data The provided functions will compute the Pesticide Load Indicator (PLI) as described in Kudsk et al. (2018) for pesticide application data provided by the user. Computing the PLI requires information on applied pesticides in a table format, as well as information on fate, ecotoxicity and human health properties of applied pesticide products, as provided in the Pesticide Properties Database (PPDB) of the University of Hertfordshire. See below for a detailed description. The PLI can either be computed using user supplied information on pesticide properties or by automatically including the information based on the PPDB. Access to the PPDB requires a license - see <http://sitem.herts.ac.uk/aeru/ppdb/>.

**Usage**

required\_columns\_products

**Format**

An object of class character of length 5.

---

substances.load      *load included substances.xlsx file*

---

**Description**

load included substances.xlsx file

**Usage**

substances.load()

**Value**

substances.xlsx file as data.frame

---

substances.path      *path to included examples substances.xlsx file*

---

**Description**

path to included examples substances.xlsx file

**Usage**

substances.path()

**Value**

path to substances.xlsx file



# Index

## \* datasets

- default.load.factors, [5](#)
- required\_columns\_products, [7](#)

- check\_products\_column\_names, [2](#)
- check\_substance\_column\_names, [2](#)
- compute\_pesticide\_load\_indicator, [3](#)
- compute\_risk\_score, [5](#)

- default.load.factors, [5](#)

- match.ppdb, [6](#)

- products.load, [6](#)
- products.path, [7](#)

- required\_columns\_products, [7](#)

- substances.load, [8](#)
- substances.path, [8](#)