Package 'PHENTHAUproc'

May 19, 2026

Title Phenology Modelling of Thaumetopoea Processionea

Version 1.1.2

Description Methods to calculate and present 'PHENTHAUproc', an early warning and decision support system for hazard assessment and control of oak processionary moth (OPM) using local and spatial temperature data. It was created by Halbig et al. 2024 (<doi:10.1016/j.foreco.2023.121525>) at FVA (<https://www.fva-bw.de/en/homepage/>) Forest Research Institute Baden-Wuerttemberg, Germany and at BOKU - University of Natural Ressources and Life Sciences, Vienna, Austria.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports grDevices, lubridate (>= 1.9), methods, rlang, stats, terra (>= 1.7), utils

Suggests dplyr, ggplot2 (>= 3.4), knitr, rmarkdown, testthat (>= 3.0.0), tidyterra (>= 0.4.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Lorenz Bachfischer [aut, cre], Department of Forest Protection, FVA [fnd]

Maintainer Lorenz Bachfischer <lorenz.bachfischer@posteo.de>

Repository CRAN

Date/Publication 2026-05-19 18:00:02 UTC

Contents

`convert_dwd_to_phenthau` *Convert hourly DWD temperature data to PHENTHAUproc input*

Description

Convert hourly DWD temperature data to PHENTHAUproc input

Usage

```
convert_dwd_to_phenthau(x)
```

Arguments

x	filepath to unzipped DWD temperature data (text file)

Value

A dataframe with date, hour and mean air temperature (tmean).

See Also

Other Helper: `call_function()`, `check_data_with_params()`, `check_dimension_and_time()`, `get_date()`, `get_formalArgs()`, `get_time()`, `load_test()`, `remove_false()`, `set_attributes()`, `subset_time()`, `timename()`

get_legend *Get legend for PHENTHAUproc models*

Description

Get legend for PHENTHAUproc models

Usage

```
get_legend(x = "stages")
```

Arguments

x	legend name - character - Available legends: "stages", "mortality", "ppa_biocide"

Value

dataframe with ID, category and colors

See Also

Other Main: `mortality()`, `parameter()`, `phenology()`, `phenthau()`

Examples

```
# return legend for development stages
get_legend("stages")
```

load_test *Load test data*

Description

Load test data

Usage

```
load_test(type = "SpatRaster")
```

Arguments

type	character, either day/hour/SpatRaster/SpatRaster_hour/budswelling

Value

data.frame/SpatRaster list with test data

See Also

Other Helper: `call_function()`, `check_data_with_params()`, `check_dimension_and_time()`, `convert_dwd_to_phenthau()`, `get_date()`, `get_formalArgs()`, `get_time()`, `remove_false()`, `set_attributes()`, `subset_time()`, `timename()`

`local_daily` *Local daily Weather Station data from Freiburg*

Description

A dataset containing daily mean, max and min temperatures for Freiburg from 2019-09-01 to 2020-09-30. Downloaded from opendata.dwd.de and preprocessed.

Format

A data frame with 396 rows and 4 variables

Details

The dataset can be loaded using load_test("day").

Stations_id: 01443 Stationsname: Freiburg

The variables are as follows:

- date in year-month-day - character
- tmean daily mean temperature in °C - numeric
- tmax daily max temperature in °C - numeric
- tmin daily min temperature in °C - numeric

Source

https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/

`local_hourly` *Local hourly Weather Station data from Freiburg*

Description

A dataset containing hourly temperatures for Freiburg from 2019-09-01 to 2022-12-31. Downloaded from opendata.dwd.de and preprocessed.

Format

A data frame with 29232 rows and 3 variables

Details

The dataset can be loaded using load_test("hour").

Stations_id: 01443 Stationsname: Freiburg

The variables are as follows:

- date in year-month-day - character
- tmean hourly mean temperature in °C - numeric

Source

https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/hourly/air_temperature/historical/

`mortality` *Calculating starvation related mortality of* Thaumetopoea processionea

---

Description

Calculating the starvation-related mortality rate of L1 larvae, dependent on degree days from the first hatch to feeding start (bud swelling of the host tree *Quercus robur*). All already available parameter settings can be shown with parameter().

Usage

```
mortality(
  x,
  par_budswelling = "quercus_robur_clone256_type1",
  par_hatch = c("custers", "wagenhoff", "meurisse"),
  def_hatch = "first",
  last = TRUE
)
```

Arguments

`x`	SpatRaster list (tmean, tmax, tmin) - numeric - with time attribute
`par_budswelling`	character - parametrisation for bud swelling
`par_hatch`	character - parametrisation for hatch
`def_hatch`	definition of hatch - character - either "first" or "mean"
`last`	If TRUE returns only the result for the last day. last row/layer for data.frame/SpatRaster

Value

SpatRaster with mortality in %.

See Also

Other Main: get_legend(), parameter(), phenology(), phenthau()

Examples

```
srl <- load_test()
mortality(srl)
```

`parameter` *Create parameter list*

Description

See all available models with parameter: parameter() model can be a single model or a model collection described in "Default settings". Return a data.frame with all model options: parameter()

Default Settings

The default parameter lists for different data input are: "dailymean": Regional PHENTHAUproc described in Halbig et al. 2024 for daily mean temperature data "hour": "Local PHENTHAUproc described in Halbig et al. 2024 for daily hourly temperature data "dailymeanminmax": PHENTHAUproc adapted to DWD Data for daily mean, min and max temperature data

Columns

model: model parametrisation: parametrisation method: method used to calculate effective temperatures ts_start: first day to calculate effective temperatures ts_end last day to calculate effective temperatures (Default 30. Sept) ts_prevyear: If True calculation of effective temperatures starts in previous year. (i.e. wagenhoff) ldt: lower development threshold cf_dependent: Is model cold/frost dependent cf_start: first day to calculate cold/frost days cf_end: last day to calculate cold/frost days cf_prevyear: If True calculation of cold/frost days starts in previous year. cf_temp: tmean" for cold days and "tmin" for frost days cf_limit: threshold for cold/frost days set: formula to calculate sum of effective temperatures a: parameter for set b: parameter for set

Usage

```
parameter(model = NULL, parametrisation = NULL, year = NULL, first = TRUE)
```

Arguments

`model`	type of model for phenthau function - character - Either single model or model collection
`parametrisation`	type of parametrisation - character
`year`	year of prognosis - numeric - Default: actual year
`first`	logical - If TRUE and parametrisation is missing first parametrisation in parameter() is used

Value

If no argument is specified returns a list of parameter (used inside phenthau function). Otherwise returns available parameter for given model, parametrisation and year

See Also

Other Main: `get_legend()`, `mortality()`, `phenology()`, `phenthau()`

Examples

```
# Default parameter list for daily mean, min and max temperature data:
parameter("dailymeanminmax")

# overview dataframe with all available parameter sets
parameter("all")

# all hatch model parameter
parameter("hatch")

# return parameter necessary for calculation
parameter("hatch", "custers", 2020)
```

---

phenology *Calculate phenological events*

Description

Using daily mean or min and max temperature data, the function calculates the temperature-dependent development stages of OPM or the bud stages (bud swelling and leaf unfolding) of its host tree *Quercus robur*.

The default settings correspond to the model described by Halbig et al. 2024. Additional parametrizations are provided but have not yet been tested.

Halbig et al. 2024 It follows 4 different steps:

- a) Calculating and summing up cold days or frost days. (Cold days are defined as days with a mean temperature below ldt (lower development threshold), while frost days are all days with a min temperature below ldt). Hatch dependent development stages need a hatch raster (hatch happened 1 or not 0) for each day
- b) Calculating degree days with the single sine method of Baskerville & Emin, 1969 or simple summing up tmean temperatures over ldt.
- c) Calculating the needed sum of effective temperatures for the development stage
- d) Comparing degree days with the needed sum of effective temperatures

Usage

```
phenology(
  x,
  model,
  parametrisation = NULL,
  year = NULL,
  hatch = NULL,
  return_date = TRUE,
  ...
)
```

Arguments

x	SpatRaster list/dataframe (tmean, tmax, tmin) - numeric - with time attribute/date column
model	name of model - character
parametrisation	name of parametrisation - character
year	year for prognosis - numeric
hatch	SpatRaster - logical - with time attribute TRUE/FALSE hatch/no_hatch
return_date	TRUE/FALSE defines output -> see value
...	parameter to change default values. (i.e. ldt = 3.5)

Value

If return_date is TRUE returns single layered SpatRaster with time serial number (first occurence of phenological event). If return_date is FALSE returns a one layer per day SpatRaster type logical with phenological event occurred/not TRUE/FALSE.

Author(s)

Bachfischer Lorenz, Department of Forest Protection FVA (2024) <lorenz.bachfischer@posteo.de>

References

Halbig et al. 2014: Halbig, P., Stelzer, A. S., Baier, P., Pennerstorfer, J., Delb, H., & Schopf, A. (2024). PHENTHAUproc–An early warning and decision support system for hazard assessment and control of oak processionary moth (Thaumetopoea processionea). Forest Ecology and Management, 552, 121525

Baskerville & Emin 1969: Baskerville, G. L., & Emin, P. (1969). Rapid estimation of heat accumulation from maximum and minimum temperatures. Ecology, 50(3), 514-517. (doi:10.2307/1933912)

Menzel 1997: Menzel, A. (1997). Phänologie von Waldbäumen unter sich ändernden Klimabedingungen: Auswertung der Beobachtungen in den internationalen phänologischen Gärten und Möglichkeiten der Modellierung von Phänodaten. Frank.

See Also

Other Main: get_legend(), mortality(), parameter(), phenthau()

Examples

```
## SpatRaster
srl <- load_test()

# Calculating bud swelling for our raster example
budswelling <- phenology(srl,
                         model = "budswelling",
                         parametrisation = "quercus_robur_clone256_type1",
                         year = 2020)
```

---

phenthau *Calculate PHENTHAUproc model*

Description

"phenthau" implements the early warning system PHENTHAUproc created by Halbig et al. 2024 in R.

Usage

```
phenthau(x, params = NULL, def_hatch = "first", budswelling = NULL)
```

Arguments

x	SpatRaster list/SpatRaster/data.frame - numeric - with time attribute/date column (see Details for Input requirements)
params	list with parameter (see Details for Input requirements)
def_hatch	"first" or "mean": PHENTHAUproc has three hatch models integrated (for details see: Custers 2003, Wagenhoff et al. 2014, Meurisse et al. 2012). With "first" (Default) the first hatch-model which predicts hatch, with "mean", the mean day of all hatch-models will be used for further calculations.
budswelling	SpatRaster/numeric with DOY (Day of year) - If budswelling is provided, internal calculation of budswelling will be replaced. For raster input provide a raster with same extend and crs as x.

Details

Overview

phenthau function combines multiple phenology models:

- bud swelling & leaf unfolding of *Quercus robur* or *Quercus petraea*(still in development)

- hatch of oak processionary moth using 3 different hatch models (Custers 2003, Wagenhoff et al. 2014 and Meurisse et al. 2012)
- development stages of OPM (0 egg - 8 adult)

Input requirements

For different input type different parameter sets are needed. If params is not provided it will be selected dependent on datatype and names(x) so follow the name convention!

daily raster input with tmean, tmin and tmax:

- a list of 3 named objects: "tmean", "tmin" and "tmax"
- each object is a SpatRaster with one layer per day
- terra::time for each SpatRaster has to be set to Date -> parameter("dailymeanminmax", year)

daily raster input with tmean:

- a SpatRaster
- one layer per day
- terra::time has to be set to Date
- params has to be set manually! -> parameter("dailymean", year)

hourly raster input:

- a SpatRaster
- one layer per hour
- terra::time has to be set to Date -> parameter("hour", year)

daily data.frame input with tmean, tmin and tmax:

- a data.frame with 4 columns: "date", "tmean", "tmin", "tmax"
- one row per day -> parameter("dailymeanminmax", year)

daily data.frame input with tmean:

- a data.frame with 2 columns: "date", "tmean"
- one row per day
- "date" column is.Date -> parameter("dailymean", year)

hourly data.frame input:

- a data.frame with 2 columns: "date", "hour", "tmean"
- 24 rows per day
- "date" column is.Date -> parameter("hour", year)

Parametrisation

Additional parametrization is provided but has not yet been tested.

Use parameter() to return a data.frame with all possible parametrization options or choose a model. The default is "dailymeanminmax" and not dependent on the data input anymore.

- "dailymean": Regional PHENTHAUproc how described in Halbig et al. 2024 for daily mean temperature data
- "hour": Local PHENTHAUproc how described in Halbig et al. 2024 for hourly temperature data
- "dailymeanminmax": PHENTHAUproc for daily mean, max and min temperature data adapted to open access DWD raster data

Output

Regional Output:
A list with all model calculations as SpatRaster objects:

- stages: SpatRaster with one layer per day - numeric - values from 0-8 (0 egg stage - 8 adult stage). -> use get_legend("stages") to show id/cover/colors
- custers/wagenhoff/meurisse - one layer per day - logical - TRUE/FALSE for hatch or no hatch
- budswelling: one layer per day - logical - TRUE/FALSE for budswelling or no budswelling
- leafunfolding: one layer per day - logical - TRUE/FALSE for leafunfolding or no leafunfolding -> plot the first day of a logical SpatRaster with plot_date()
- mortality: one layer - integer - mortality in %
- ppa_biocide: one layer per day - numeric - 0 application of plant protection agents (PPA) and biocides not yet effective, 1 application effective, 2 appliaction not effective anymore

Local Output:
A data.frame with two columns:

- model: name of model
- date: date of first appearing of event

Presentation

Regional Output:

- plot_stages is a wrapper around terra::plot to preset legend, names, colors and day to plot.
- plot_date is a wrapper around terra::plot to plot the date of first TRUE in multiple layered SpatRaster

Local Output:

- plot_station_step is for local weather station data and creates a stepwise graph for the development stages

Value

data.frame or list of SpatRaster with all PHENTHAUproc model outputs (see Details)

Author(s)

Bachfischer Lorenz, Department of Forest Protection FVA (2024) <lorenz.bachfischer@posteo.de>

References

Halbig et al. 2024: Halbig, P., Stelzer, A. S., Baier, P., Pennerstorfer, J., Delb, H., & Schopf, A. (2024). PHENTHAUproc–An early warning and decision support system for hazard assessment and control of oak processionary moth (Thaumetopoea processionea). Forest Ecology and Management, 552, 121525

Custers 2003: Custers, C. (2003). Climate change and trophic synchronisation. English Wageningen UR, Chairgroup Environmental Systems Analysis.

Wagenhoff et al. 2014: Wagenhoff, E., Wagenhoff, A., Blum, R., Veit, H., Zapf, D., & Delb, H. (2014). Does the prediction of the time of egg hatch of Thaumetopoea processionea (Lepidoptera: Notodontidae) using a frost day/temperature sum model provide evidence of an increasing temporal mismatch between the time of egg hatch and that of budburst of Quercus robur due to recent global warming?. European Journal of Entomology, 111(2).

Meurisse et al. 2012: Meurisse, N., Hoch, G., Schopf, A., Battisti, A., & Grégoire, J. C. (2012). Low temperature tolerance and starvation ability of the oak processionary moth: implications in a context of increasing epidemics. Agricultural and forest entomology, 14(3), 239-250.

See Also

Other Main: `get_legend()`, `mortality()`, `parameter()`, `phenology()`

Examples

```
srl <- load_test()
phen <- phenthau(srl)
```

`plot_date` *Plot SpatRaster with date serial number/time attribute*

Description

A wrapper around terra::plot to show the time serial number as a character date in the legend.

Usage

```
plot_date(x, breaks = NULL, ...)
```

Arguments

`x`	SpatRaster - numeric - value is serial number or SpatRaster - logical - with time attribute
`breaks`	number of breaks in the legend - numeric
`...`	arguments for terra::plot function, except (type, breaks, col or plg)

Value

A plot of a phenological event by day with legend.

See Also

Other Plot: plot_stages(), plot_station_step()

Examples

```
data <- load_test()
budswelling <- phenology(data, "budswelling", "quercus_robur_clone256_type1", 2020)
plot_date(budswelling)
```

`plot_stages` *Plot regional PHENTHAUproc - Stages*

Description

A wrapper to plot the development stages of OPM with assigned names and colors

Usage

```
plot_stages(x, time = NULL, ...)
```

Arguments

x	SpatRaster stages output of phenthau - numeric
time	day to plot - Date or character year-month_day i.e.("2020-05-01")
...	arguments passed along to terra::plot

Details

phenthau returns a list of SpatRasters. The stages object describes the development stages of oak processionary moth. This function plots the stages SpatRaster with the right names and colors. To get IDs, caregories and colors use get_legend("stages").

Value

A plot of the PHENTHAUproc stages of the last/chosen time with preset levels and colors.

See Also

Other Plot: plot_date(), plot_station_step()

plot_station_step *Plot local PHENTHAUproc in a step plot*

Description

Plots the development stages of OPM.

Usage

```
plot_station_step(x)
```

Arguments

x	output of phenthau - dataframe - local PHENTHAUproc

Value

A plot showing local PHENTHAUproc results in a step plot.

See Also

Other Plot: `plot_date()`, `plot_stages()`

Examples

```
fr_df <- load_test("day")
fr <- phenthau(fr_df)

plot_station_step(fr)
```

regional *Regional Weather data*

Description

A dataset containing daily mean, minimum and maximum temperatures in °C. The dataset is a 4*4 pixel cutout centered at FVA from the "Hyras" dataset available at the DWD open data center (https://opendata.dwd.de/climate_environment/CDC/grids_germany/daily/hyras_de/)

Format

A list of SpatRaster

Details

The dataset can be loaded using load_test("SpatRaster").

Spatial resolution: 5 km x 5 km Projection: ETRS89 / LCC Europe (EPSG:3034) Parameter: air temperature at 2 m

The dataset is a list with three SpatRaster as objects. The time attribute for all three SpatRaster is equal.

The list objects are:

- tmean: SpatRaster with tmean daily mean temperature in °C - numeric
- tmax: SpatRaster with tmax daily mean temperature in °C - numeric
- tmin: SpatRaster with tmin daily mean temperature in °C - numeric

Index