

# Package ‘PHENTHAUproc’

January 29, 2024

**Title** Phenology Modelling of Thaumetopoea Processionea

**Version** 1.0.1

**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](https://doi.org/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 Resources and Life Sciences, Vienna, Austria.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Depends** R (>= 3.5)

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

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

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

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

**Maintainer** Lorenz Bachfischer <[lorenz.bachfischer@posteo.de](mailto:lorenz.bachfischer@posteo.de)>

**Repository** CRAN

**Date/Publication** 2024-01-29 13:10:14 UTC

## R topics documented:

convert_dwd_to_phenthau . . . . .	2
get_legend . . . . .	3
load_test . . . . .	3
local_daily . . . . .	4

local_hourly . . . . .	4
mortality . . . . .	5
parameter . . . . .	6
phenology . . . . .	7
phenthau . . . . .	9
plot_date . . . . .	11
plot_stages . . . . .	12
plot_station . . . . .	13
plot_station_step . . . . .	14
regional . . . . .	14
<b>Index</b>	<b>16</b>

---

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	<i>Get legend for PHENTHAUproc models</i>
------------	---

---

**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	<i>Load test data</i>
-----------	-----------------------

---

**Description**

Load test data

**Usage**

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

**Arguments**

type character, either day/hour/SpatRaster

**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	<i>Local daily Weather Station data from Freiburg</i>
-------------	---

---

**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/](https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/daily/kl/historical/)

---

local_hourly	<i>Local hourly Weather Station data from Freiburg</i>
--------------	--

---

**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/](https://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/hourly/air_temperature/historical/)

---

mortality	<i>Calculating starvation related mortality of Thaumetopoea processionea</i>
-----------	--

---

**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 none is specified returns a list of parameter (used inside phenthau function). Otherwise returns available parameter for given model, parametrisation & 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	returns start of phenological event with time serial number
...	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 occurrence 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
sr1 <- load_test()

# Calculating bud swelling for our raster example
budswelling <- phenology(sr1, "budswelling", "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")
```

**Arguments**

x	SpatRaster list/data.frame (tmean, tmax, tmin) - numeric - with time attribute/date column
params	list with parameter. If NULL parameter set is chosen by data input
def_hatch	definition of hatch - character - either "first" or "mean"

**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)

**Parametrisation**

Additional parametrizations are provided but have not yet been tested.

Use parameter() to return a data.frame with all possible parametrisation options or choose a model. The default is dependent on the data input and correspond to the model described by Halbig et al. 2024.

- "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
- "dailymeanmaxmin": PHENTHAUproc for daily mean, max and min temperature data adapted to DWD
- def\_hatch: how to calculate hatch. if "first" the first hatch prediction by any model is taken. If "mean" the mean day of all hatch models is taken.

## Output

Regional Output: phenthau for a SpatRasterList input returns 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 application not effective anymore

Local Output: phenthau for a data.frame input returns 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 is for local weather station data and creates a phenology wheel (ggplot2 & geom-textpath required!)
- plot\_station\_step is for local weather station data and creates a stepwise graph for the development stages

## Value

data.frame/list 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

```
sr1 <- load_test()
phen <- phenthau(sr1)
```

---

plot_date	<i>Plot SpatRaster with date serial number/time attribute</i>
-----------	---

---

## 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 <code>terra::plot</code> 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\(\)](#), [plot\\_station\(\)](#)

**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 is 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\(\)](#)

---

plot_station	<i>Plot local PHENTHAUproc in a circle plot</i>
--------------	---

---

**Description**

Plot local PHENTHAUproc in a circle plot

**Usage**

```
plot_station(x, main = "PHENTHAUproc")
```

**Arguments**

x	output of phenthau - dataframe - local PHENTHAUproc
main	title - character

**Value**

A plot showing local PHENTHAUproc results in a circle plot presentation.

**References**

Allan Cameron 2022: Allan Cameron (2022), Answer to the question "Is it possible to create a circular timeline plot with ggplot?" on Stackoverflow, (<https://stackoverflow.com/questions/71023419/is-it-possible-to-create-a-circular-timeline-plot-with-ggplot>)

**See Also**

Other Plot: [plot\\_date\(\)](#), [plot\\_stages\(\)](#), [plot\\_station\\_step\(\)](#)

**Examples**

```
# load daily test data from Freiburg
fr_df <- load_test("day")
fr <- phenthau(fr_df)

plot_station(fr)
```

---

plot_station_step	<i>Plot local PHENTHAUproc in a step plot</i>
-------------------	---

---

**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\(\)](#), [plot\\_station\(\)](#)

**Examples**

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

plot_station_step(fr)
```

---

regional	<i>Regional Weather data</i>
----------	------------------------------

---

**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/](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

## \* **Helper**

convert\_dwd\_to\_phenthau, 2  
load\_test, 3

## \* **Main**

get\_legend, 3  
mortality, 5  
parameter, 6  
phenology, 7  
phenthau, 9

## \* **Plot**

plot\_date, 11  
plot\_stages, 12  
plot\_station, 13  
plot\_station\_step, 14

## \* **datasets**

local\_daily, 4  
local\_hourly, 4  
regional, 14

call\_function, 2, 4  
check\_data\_with\_params, 2, 4  
check\_dimension\_and\_time, 2, 4  
convert\_dwd\_to\_phenthau, 2, 4

get\_date, 2, 4  
get\_formalArgs, 2, 4  
get\_legend, 3, 6–8, 11  
get\_time, 2, 4

load\_test, 2, 3  
local\_daily, 4  
local\_hourly, 4

mortality, 3, 5, 7, 8, 11

parameter, 3, 6, 6, 8, 11  
phenology, 3, 6, 7, 7, 11  
phenthau, 3, 6–8, 9  
plot\_date, 11, 12–14  
plot\_stages, 12, 12, 13, 14  
plot\_station, 12, 13, 14

plot\_station\_step, 12, 13, 14

regional, 14  
remove\_false, 2, 4

set\_attributes, 2, 4  
subset\_time, 2, 4

timename, 2, 4