Package 'dendrometeR'

July 22, 2025

Type Package

Title Analyzing Dendrometer Data

Version 1.1.1

Maintainer Marko Smiljanic <marko.smiljanic313@gmail.com>

Description Various functions to import, verify, process and plot high-resolution dendrometer data using daily and stem-cycle approaches as described in Deslauriers et al, 2007 <doi:10.1016/j.dendro.2007.05.003>. For more details about the package please see: Van der Maaten et al. 2016 <doi:10.1016/j.dendro.2016.06.001>.

License GPL (>= 2)

URL https://github.com/smiljanicm/dendrometeR

BugReports https://github.com/smiljanicm/dendrometeR/issues

Depends R (>= 4.0.0), forecast, graphics, grDevices, pspline, stats, zoo

Imports methods

Suggests knitr

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Olivier Bouriaud [aut], Ernst van der Maaten [aut], Marieke van der Maaten-Theunissen [aut], Marko Smiljanic [aut, cre]

Repository CRAN

Date/Publication 2025-02-21 13:40:02 UTC

Contents

`climate_seg` *Segmenting climate and environmental data*

Description

The function calculates means or sums, or extracts minimum or maximum values of environmental parameters for stem-cyclic phases as defined using `cycle_stats`.

Usage

```
climate_seg(env.data, dm.stats, value = c("mean", "min",
            "max", "sum"))
```

Arguments

`env.data`	a `data.frame` with with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and a certain climate parameter (e.g., temperature or precipitation) in columns.
`dm.stats`	a `list` as produced by `cycle_stats`.
`value`	a `character` string of `"mean"`, `"min"`, `"max"` or `"sum"`, specifying whether means (e.g., for temperature) or sums (e.g., for precipitation) should be calculated, or minimum or maximum values should be extracted. Defaults to `"mean"`. Argument matching is performed.

Details

The function segments environmental parameters according to the stem-cyclic phases as defined using `cycle_stats`. Means, sums, and minimum and maximum values can be calculated or extracted.

`env.data` should cover at least the same period as the dendrometer data used to define the cyclic phases, and should have the same (or a higher) temporal resolution.

Value

The function returns a `data.frame` with segmented environmental data. The `data.frame` contains the following columns:

`dmID`	dendrometer ID.
`cycle`	cycle number.
`phase`	cyclic phase (1: contraction, 2: expansion, 3: stem-radius increment, 4: full cycle).
`begin`	timestamp indicating the beginning of each phase.
`end`	timestamp indicating the end of each phase.
`...`	columns with segmented environmental data (mean, min, max or sum).

Examples

```
data(dmED)
dm.gpf <- fill_gaps(dmED)
dm.phase <- phase_def(dm.gpf)
dm.stats <- cycle_stats(dm.gpf, dm.phase)
data(envED)
clim.phase <- climate_seg(envED, dm.stats, value = "mean")
```

`cycle_stats` *Define stem cycles and calculate statistics for all cyclic phases*

Description

The function defines stem cycles from output of `phase_def` and calculates statistics for complete cycles as well as for the phases of contraction, expansion and stem-radius increment.

Usage

```
cycle_stats(dm.gpf, dm.phase, sensor = 1, smooth.param = 1)
```

Arguments

dm.gpf	a data.frame with either gap-free or gap-filled dendrometer series as produced by fill_gaps.
dm.phase	a data.frame with numbers indicating the different stem-cyclic phases. Output of phase_def.
sensor	a numeric specifying the sensor to be used in the function. Defaults to 1 (first column in both data.frames).
smooth.param	a numeric specifying the degree of smoothing. Defaults to 1 (no smoothing).

Details

The function uses the output of phase_def to define stem cycles and to calculate statistics for all cyclic phases. These statistics include the timing and duration of each phase, as well as information on stem-size changes. The function works for single dendrometer series, which are defined by the argument sensor.

The function includes a smoothing option (argument smooth.param) particularly for noisy datasets in which outliers may under- or overestimate the minimum and maximum stem size within phases and stem cycles. By default, no smoothing is performed.

Value

The function returns a list with:

- a data.frame named cycleStats containing the following summary statistics:

dmID	dendrometer ID.
cycle	cycle number.
phase	cyclic phase (1: contraction, 2: expansion, 3: stem-radius increment, 4: full cycle).
begin	timestamp indicating the beginning of each phase.
end	timestamp indicating the end of each phase.
duration_h	phase duration in hours.
duration_m	phase duration in minutes.
magnitude	magnitude of stem-size changes in each phase.
min	minimum stem size within each phase.
max	maximum stem size within each phase.

- a data.frame named cycle.df containing, for all individual records, the following columns:

dmID	dendrometer ID.
cycle	cycle number.
phase	cyclic phase (1: contraction, 2: expansion, 3: stem-radius increment, 4: full cycle).

Author(s)

Olivier Bouriaud, Ernst van der Maaten and Marieke van der Maaten-Theunissen.

Examples

```
data(dmCD)
dm.phase <- phase_def(dmCD)
dm.stats <- cycle_stats(dmCD, dm.phase)
```

---

daily_stats *Calculate daily statistics for dendrometer and environmental data*

Description

The function calculates various daily statistics for dendrometer and environmental data. It either returns multiple statistics for individual sensors, or a single statistic for multiple sensors.

Usage

```
daily_stats(dm.data, sensor = 1, value = c("mean", "min",
            "max", "sum"), smooth.param = 1)
```

Arguments

dm.data	a data.frame with a timestamp (%Y-%m-%d %H:%M:%S format) as row names, and dendrometer series in columns. Output as created using code from the Import dendrometer data vignette, or gap-filled dendrometer series as produced by fill_gaps. Environmental data can be specified as well, and should be formatted as dendrometer data.
sensor	a numeric or character string specifying the sensor(s) to be used in the function. Defaults to 1 (first column of data.frame). If "ALL" is specified, a single value will be calculated or extracted for all series in the data.frame.
value	a character string of "mean", "min", "max" or "sum", specifying the daily statistic to be calculated or extracted. Optional argument for sensor = "ALL", defaults to "mean". Argument matching is performed.
smooth.param	a numeric specifying the degree of smoothing. Defaults to 1 (no smoothing). In case smoothing is applied, series should be gap-free or gap-filled.

Details

The function calculates various daily statistics for dendrometer and environmental data. For sensor is numeric, the function returns multiple statistics for a single sensor. For sensor = "ALL", the function returns a single statistic (i.e. "mean", "min", "max" or "sum") for all columns of the data.frame, whereby "sum" is particularly relevant for environmental parameters like precipitation.

The function includes a smoothing option (argument `smooth.param`) particularly for noisy datasets in which outliers may under- or overestimate minimum and maximum stem sizes within days. By default, no smoothing is performed. Smoothing requires gap-free series.

Value

The function returns:

- for `sensor` is `numeric`, a `data.frame` containing the following columns:

`dmID`	dendrometer ID.
`date`	timestamp in `%Y-%m-%d` format.
`DOY`	day of year.
`min`	minimum daily stem size.
`mean`	mean daily stem size.
`max`	maximum daily stem size.
`amplitude`	amplitude of daily stem-size changes (i.e. max - min).
`time_min`	timestamp indicating the timing of the minimum.
`time_max`	timestamp indicating the timing of the maximum.

- for `sensor` is `"ALL"`:

a `data.frame` with a timestamp (`%Y-%m-%d`) as row names, and processed dendrometer or environmental data in columns (i.e. mean, minimum, maximum or sum).

Author(s)

Olivier Bouriaud, Ernst van der Maaten and Marieke van der Maaten-Theunissen.

Examples

```
data(dmCD)
dm.daily <- daily_stats(dmCD, sensor = 1)

data(dmED)
dm.daily <- daily_stats(dmED, sensor = "ALL", value = "max")
```

dendro.resolution *Check the resolution of the data*

Description

The function provides the resolution of the dendrometer data.

Usage

```
dendro.resolution(dm.data, unts = c("secs", "mins", "hours", "days"))
```

Arguments

dm.data	a data.frame with a timestamp (%Y-%m-%d %H:%M:%S format) as row names, and dendrometer series in columns. Output as created using code from the Import dendrometer data vignette.
unts	a character string of "secs", "mins", "hours", "days", specifiying the units in which the resolution should be calculated. Defaults to "secs". Argument matching is performed.

Value

The function returns the resolution of the data in the desired unit.

Author(s)

Marko Smiljanic

Examples

```
data(dmCD, dmHS, dmED)
dendro.resolution(dmCD, unts = "hours")
dendro.resolution(dmHS, unts = "hours")
dendro.resolution(dmED, unts = "mins")
```

dmCD *Pre-processed dendrometer data from Camp Daniel, Canada*

Description

This dataset presents a pre-processed version of dmCDraw, in which different time variables were converted to a timestamp using code provided in the Import dendrometer data vignette.

Usage

```
data(dmCD)
```

Format

A `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and the dendrometer series in the first column.

`dmCDraw` *Raw dendrometer data from Camp Daniel, Canada*

Description

This dataset presents hourly dendrometer series for a black spruce (*Picea mariana* (Mill.) BSP) tree from Camp Daniel, Canada, for the year 2008.

Usage

```
data(dmCDraw)
```

Format

A `data.frame` with a dendrometer series and various time variables.

Source

Sergio Rossi

`dmED` *Pre-processed dendrometer data from Eldena, Germany*

Description

This dataset presents a pre-processed version of `dmEDraw`, using code provided in the `Import dendrometer data` vignette.

Usage

```
data(dmED)
```

Format

A `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and the two dendrometer series in the first and second column.

dmEDraw *Raw dendrometer data from Eldena, Germany*

Description

This dataset presents half-hourly dendrometer series for two European beech (*Fagus sylvatica* L.) trees from the monitoring plot Eldena, Germany, for the year 2015.

Usage

```
data(dmEDraw)
```

Format

A `data.frame` with dendrometer series and a timestamp.

Source

Martin Wilmking

dmHS *Pre-processed dendrometer data from Hinnensee, Germany*

Description

This dataset presents a pre-processed version of `dmHSraw`, using code provided in the `Import dendrometer data` vignette.

Usage

```
data(dmHS)
```

Format

A `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and the dendrometer series in the first column.

dmHSraw *Raw dendrometer data from Hinnensee, Germany*

Description

This dataset presents half-hourly dendrometer series for a European beech (*Fagus sylvatica* L.) tree from the monitoring plot Hinnensee, Germany, for the year 2012.

Usage

```
data(dmHSraw)
```

Format

A `data.frame` with a dendrometer series and various time variables.

Source

Sonia Simard

envED *Environmental data from Eldena, Germany*

Description

This dataset presents some temperature data from the monitoring plot Eldena, Germany, for the year 2015.

Usage

```
data(envED)
```

Format

A `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and air and soil temperature parameters in columns.

Source

Martin Wilmking

`fill_gaps` *Fill gaps in dendrometer series*

Description

The function fills gaps in a `data.frame` with dendrometer series using an ARMA model (cf. Deslauriers et al. 2011), and is designed for single growing seasons. The function is able to fill gaps of short duration (i.e. several hours), but cannot sensibly handle long gaps.

Usage

```
fill_gaps(dm.data, Hz = 0.01, season = FALSE)
```

Arguments

`dm.data`	a `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and dendrometer series in columns. Output as created using code from the `Import dendrometer data` vignette.
`Hz`	a `numeric` specifying the parameter for smoothing with ARMA gap-filling. A higher value means rougher smoothing. Defaults to 0.01.
`season`	a `logical` indicating whether `auto.arima` should check seasonal models; can be very slow. Defaults to FALSE, i.e. search restricted to non-seasonal models.

Details

The function uses `auto.arima` to fill missing records. The non-seasonal part of the model is specified by the three integer components: the AR order p, the degree of differencing d, and the MA order q. For the seasonal part of the model, the period parameter is set equal to the number of daily measurements observed in the dendrometer data. The output of the ARMA model is smoothed using `smooth.Pspline`. The smoothing parameter Hz can be adjusted; defaults to 0.01.

The function is designed for single growing seasons, amongst others because ARMA-based gap-filling routines will then perform best (i.e. ARMA parameters might be distinct for individual growing seasons). To allow the usage of `fill_gaps` for datasets from the Southern Hemisphere, the input data may contain two consecutive calendar years.

Value

The function returns a `data.frame` with gap-filled dendrometer series.

Author(s)

Olivier Bouriaud, Ernst van der Maaten, Marieke van der Maaten-Theunissen and Marko Smiljanic.

References

Deslauriers, A., Rossi, S., Turcotte, A., Morin, H. and Krause, C. (2011) A three-step procedure in SAS to analyze the time series from automatic dendrometers. *Dendrochronologia* 29: 151-161.

Examples

```
data(dmCD)

# creating some artificial gaps (for demonstration purposes):
dmCD[c(873:877,985:990),1] <- NA

# slow, as also seasonal models are checked, but best possible gap-filling:
dm.gpf <- fill_gaps(dmCD, Hz = 0.01, season = TRUE)
```

`fill_plot` *Plot gap-filled dendrometer series*

Description

The function creates a plot with gap-filled and original dendrometer series.

Usage

```
fill_plot(dm.data, dm.gpf, sensor = 1, year = NULL,
          period = NULL)
```

Arguments

`dm.data`	a `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and dendrometer series in columns. Output as created using code from the `Import dendrometer data` vignette.
`dm.gpf`	a `data.frame` with gap-filled dendrometer series as produced by `fill_gaps`.
`sensor`	a `numeric` specifying the sensor to be plotted (by column number). Defaults to 1 (first dendrometer series in both `data.frames`).
`year`	a `numeric` specifying the year(s) to be plotted. Defaults to the first year in the dataset. Two consecutive years (e.g., for a growing season at the Southern Hemisphere) can be defined with `year = c(year1, year2)`.
`period`	a `numeric` indicating the period to be plotted, specified using day of year values (begin and end). Defaults to the complete data period.

Details

The function creates a plot showing the gap-filling results for a single dendrometer series over a specified time window. Although the function is intended to plot short time periods (within a growing season), it can plot two calendar years at maximum (e.g., 2014-2015), thereby allowing the visualization of a complete growing season at the Southern Hemisphere as well.

Value

Plot.

Author(s)

Olivier Bouriaud, Ernst van der Maaten and Marieke van der Maaten-Theunissen.

Examples

```
data(dmCD)
## creating some artificial gaps (for demonstration purposes):
dmCD[c(873:877,985:990),1] <- NA
dm.gpf <- fill_gaps(dmCD, Hz = 0.01)
fill_plot(dmCD, dm.gpf, period = c(137,144))
```

---

is.dendro *Check input data*

Description

The function checks whether the input data is in the required format, as described in the `Import dendrometer data` vignette.

Usage

```
is.dendro(dm.data)
```

Arguments

`dm.data`	a `data.frame` with a timestamp (`%Y-%m-%d %H:%M:%S` format) as row names, and dendrometer series in columns. Output as created using code from the `Import dendrometer data` vignette.

Value

The function returns `TRUE` if the input data is valid and `FALSE` otherwise. In the latter case, specific error messages are given as well.

Author(s)

Ernst van der Maaten, Marieke van der Maaten-Theunissen and Marko Smiljanic.

Examples

```
data(dmCD, dmHS, dmED)
is.dendro(dmCD)
is.dendro(dmHS)
is.dendro(dmED)
```

phase_def *Define stem-cyclic phases*

Description

The function identifies and assigns each timestamp to one of the three distinct phases of contraction, expansion and stem-radius increment (Deslauriers et al. 2011) for dendrometer series from a `data.frame` with gap-free dendrometer data.

Usage

```
phase_def(dm.gpf, resolution = dendro.resolution(dm.gpf),
          shapeSensitivity = 0.6, minmaxDist = 0.2, minmaxSD = 2,
          radialIncrease = "max")
```

Arguments

`dm.gpf`	a `data.frame` with either gap-free or gap-filled dendrometer series as produced by `fill_gaps`.
`resolution`	a `numeric` specifying the resolution of the dendrometer data in seconds. Defaults to the resolution of `dm.gpf` as calculated using `dendro.resolution`.
`shapeSensitivity`	a `numeric` specifying a time window, defined as proportion of a single day. Within this time window possible extrema points (i.e. minimum and maximum) in dendrometer measurements are searched for. Defaults to 0.6 (60 percent of a day). See details for further explanation.
`minmaxDist`	a `numeric` specifying the minimum temporal distance between consecutive minimum and maximum points (i.e. in the x direction). Defaults to 0.2 (20 percent of a day).
`minmaxSD`	a `numeric` specifying the minimum difference between consecutive minimum and maximum points expressed as a number of standard deviations (i.e. in the y direction). Defaults to 2.
`radialIncrease`	a `character` string of `"max"`, `"min"`, `"mid"`, specifying when the stem-radius increment phase should start, with `"max"` as the most, and `"min"` as the least conservative approach; `"mid"` is in between. See details for further explanation.

Details

The function defines the stem-cyclic phases of contraction, expansion, and stem-radius increment, as described in Deslauriers et al. (2011). The function is a more robust version of the original SAS routine, as its architecture allows to handle noisy data as well.

First, the function searches for minimum and maximum points within a daily time window as specified by `shapeSensitivity`. Then, the original dendrometer series are offset by `(1 - shapeSensitivity) / 2` in both directions to assure whether the identified extrema are indeed the extrema of cyclic phases. A comparison between the original and offset series allows to select all appropriate minimum and maximum values.

The arguments `minmaxDist` and `minmaxSD` specify the temporal distance and the minimum difference between consecutive minimum and maximum points (i.e. in x and y direction), respectively. The argument `radialIncrease` determines from which moment on data points should be assigned to the stem-radius increment phase: when points are continuously above the previous maximum (`"max"`), when a single data point is above the previous maximum (`"min"`), or right in between `"min"` and `"max"` (`"mid"`).

Value

The function returns a `data.frame` with numbers indicating the different stem-cyclic phases: (1) contraction, (2) expansion, (3) stem-radius increment for each timestamp.

Author(s)

Marko Smiljanic

References

Deslauriers, A., Rossi, S., Turcotte, A., Morin, H. and Krause, C. (2011) A three-step procedure in SAS to analyze the time series from automatic dendrometers. *Dendrochronologia* 29: 151-161.

Examples

```
data(dmCD)
dm.phase <- phase_def(dmCD)
```

---

phase_plot *Plot stem-cyclic phases*

Description

The function creates a plot showing the three distinct phases of contraction, expansion and stem-radius increment (Deslauriers et al. 2011) for dendrometer series from a `data.frame` as produced by `phase_def`.

Usage

```
phase_plot(dm.gpf, dm.phase, sensor = NULL, period = NULL, colPhases = NULL, ...)
```

Arguments

`dm.gpf`	a `data.frame` with gap-filled dendrometer series as produced by `fill_gaps`.
`dm.phase`	a `data.frame` with numbers indicating the different stem-cyclic phases. Output of `phase_def`.

sensor	a numeric specifying the sensor to be plotted (by column number). Alternatively, sensor can be a character with column names. Concatenations and sequences are allowed for plotting phase definitions of multiple sensors at once. Defaults to all sensors in dm.gpf and dm.phase.
period	a numeric indicating the period to be plotted, specified using day of year values (begin and end). Defaults to the complete data period. Alternatively, period can be a character of two time stamps, indicating the begin and end date of the period to be plotted.
colPhases	a vector of length 3, specifying custom colors to be used for the three stem-cyclic phases. Defaults to the first three colors from the current palette.
...	additional graphical parameters (see par).

Details

The function plots phases of contraction, expansion and stem-radius increment along (one or more) dendrometer series. If more series are plotted (default), colors for the different lines can be defined using the col argument for graphical devices (see par). Note: if there are not enough custom colors, the function will repeat the last one used. If no colors are defined, the current palette will be used.

The time axis will be automatically labeled depending upon the length of the dendrometer series. If period is specified using a numeric, DOY values are displayed on the x-axis. In case a character of two time stamps is provided, axis labeling will be as follows: if series are longer than 120 days, years and months will be shown. If the length is between 30 and 120 days, months and days, and below 30 days, months, days and hours are displayed.

Value

Plot showing stem-cyclic phases on dendrometer series.

Author(s)

Marko Smiljanic

References

Deslauriers, A., Rossi, S., Turcotte, A., Morin, H. and Krause, C. (2011) A three-step procedure in SAS to analyze the time series from automatic dendrometers. *Dendrochronologia* 29: 151-161.

Examples

```
data(dmCD)
dm.phase <- phase_def(dmCD)
phase_plot(dmCD, dm.phase)

# zoom in on the dendrometer series
phase_plot(dmCD, dm.phase, period = c(133, 142))

# customization options
phase_plot(dmCD, dm.phase, period = c("2008-05-12", "2008-05-22"),
           colPhases = c("green", "cyan", "orange"),
```

```
            pch = 4, main = "Dendrometer", ylab = "Values")

# specific sensors may be selected as follows:


data(dmED)
dm.gpf <- fill_gaps(dmED)
dm.phase <- phase_def(dm.gpf)
phase_plot(dm.gpf, dm.phase, sensor = 1)
phase_plot(dm.gpf, dm.phase, sensor = c(2,1))
phase_plot(dm.gpf, dm.phase, sensor = "Beech03")
phase_plot(dm.gpf, dm.phase, sensor = c("Beech03", "Beech04"))
```

Index