Package 'phenocamr'

March 22, 2020

Title Facilitates 'PhenoCam' Data Access and Time Series Post-Processing

Version 1.1.4

Description Programmatic interface to the 'PhenoCam' web services (<http://phenocam.sr.unh.edu>). Allows for easy downloading of 'PhenoCam' data directly to your R workspace or your computer and provides post-processing routines for consistent and easy timeseries outlier detection, smoothing and estimation of phenological transition dates. Methods for this package are described in detail in Hufkens et. al (2018) <doi:10.1111/2041-210X.12970>.

Depends R (>= 3.4.0),

Imports jsonlite, changepoint, httr, zoo, utils, graphics, stats, memoise, daymetr, MODISTools

Suggests shiny, covr, testthat, knitr, rmarkdown, shinydashboard, leaflet, plotly, DT

License AGPL-3

LazyData true

ByteCompile true

RoxygenNote 7.0.0

VignetteBuilder knitr

NeedsCompilation no

Author Hufkens Koen [aut, cre]

Maintainer Hufkens Koen <koen.hufkens@gmail.com>

Repository CRAN

Date/Publication 2020-03-22 17:40:05 UTC

R topics documented:

`contract_phenocam` *Contracts the file from 1-day to a 3-day time step*

Description

Reverts the 'expand_phenocam()' function in order to save space and generate files as outlined in the cited data paper. This routine is used as a post-production measure.

Usage

```
contract_phenocam(
  data,
  internal = TRUE,
  no_padding = FALSE,
  out_dir = tempdir()
)
```

Arguments

`data`	a phenocam data file with a 1 or 3 day time step
`internal`	return a data structure if given a file on disk (`TRUE` / `FALSE` = default)
`no_padding`	allow for padding to REMAIN or not (`TRUE` / `FALSE` = default)
`out_dir`	output directory where to store data (default = tempdir())

Value

A contracted PhenoCam 3-day time series to its original 3-day time step (if provided at a 1-day interval), also removes padding introduced by processing for 1-day data.

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# Overwrites the original file, increasing
# it's file size.
expand_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

# Contracts the file to it's original size, skipping
# two days.
contract_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

## End(Not run)
```

---

daylength *Calculates day length (in hours) and the solar elevation*

Description

This routine uses Forsythe et al. 1995.

Usage

```
daylength(doy, latitude)
```

Arguments

doy	a vector with doy values 1 - 365(6)
latitude	a given latitude

Value

nested list with daylength (daylength) and solar elevation (solar_elev) elements

Examples

```
## Not run:
# calcualte the hours of sunlight and solar elevation on day of year 1
# and latitude 51
ephem <- daylength(1, 51)
print(ephem)

## End(Not run)
```

`detect_outliers` *Detect outliers in PhenoCam time series*

Description

The function fills in the existing column to hold outlier flags, and either overwrites the original file or outputs a data structure.

Usage

```
detect_outliers(
  data,
  iterations = 20,
  sigma = 2,
  grvi = FALSE,
  snowflag = FALSE,
  plot = FALSE,
  internal = TRUE,
  out_dir = tempdir()
)
```

Arguments

`data`	PhenoCam data structure or filename
`iterations`	number of itterations in order to detect outliers ()
`sigma`	number of deviations to exclude outliers at
`grvi`	reverse the direction of the screening intervals to accomodate for GRVI outliers
`snowflag`	use manual snow flag labels as outliers
`plot`	visualize the process, mostly for debugging (`TRUE` / `FALSE` = default)
`internal`	return a data structure if given a file on disk (`TRUE` / `FALSE` = default) to accomodate for GRVI outliers
`out_dir`	output directory where to store data

Examples

```
## Not run:
# download demo data (do not detect outliers)
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3",
                  outlier_detection = FALSE)

# detect outliers in the downloaded file
detect_outliers(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
```

```
## End(Not run)
```

---

download_phenocam *Downloads PhenoCam time series*

Description

This is a wrapper around most of all the other functions. It downloads a time series and extract relevant phenological transition dates or phenophases.

Usage

```
download_phenocam(
  site = "harvard$",
  veg_type = NULL,
  frequency = "3",
  roi_id = NULL,
  outlier_detection = TRUE,
  smooth = TRUE,
  contract = FALSE,
  daymet = FALSE,
  trim_daymet = TRUE,
  trim = NULL,
  phenophase = FALSE,
  out_dir = tempdir(),
  internal = FALSE
)
```

Arguments

site	the site name, as mentioned on the PhenoCam web page expressed as a regular expression ("harvard$" == exact match)
veg_type	vegetation type (DB, EN, ... default = ALL)
frequency	frequency of the time series product (1, 3, "roistats")
roi_id	the id of the ROI to download (default = ALL)
outlier_detection	TRUE or FALSE, detect outliers
smooth	smooth data (logical, default is TRUE)
contract	contract 3-day data (logical, default is TRUE)
daymet	TRUE or FALSE, merges the daymet data
trim_daymet	TRUE or FALSE, trims data to match PhenoCam data
trim	year (numeric) to which to constrain the output (default = NULL)
phenophase	logical, calculate transition dates (default = FALSE)
out_dir	output directory where to store downloaded data (default = tempdir())
internal	allow for the data element to be returned to the workspace

Value

Downloaded files in out_dir of requested time series products, as well as derived phenophase estimates based upon these time series.

Examples

```
## Not run:
# download the first ROI time series for the Harvard PhenoCam site
# at an aggregation frequency of 3-days.
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# read phenocam data into phenocamr data structure
df <- read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))


## End(Not run)
```

`expand_phenocam` *Expand a PhenoCam time series from 3-day to a 1-day time step*

Description

Necessary step to guarantee consistent data processing between 1 and 3-day data products. Should rarely be used independent of 'download_phenocam()'.

Usage

```
expand_phenocam(data, truncate = NULL, internal = TRUE, out_dir = tempdir())
```

Arguments

`data`	a PhenoCam file
`truncate`	year (numerical), limit the time series to a particular year (default = NULL)
`internal`	return a data structure if given a file on disk (`TRUE` / `FALSE` = default)
`out_dir`	output directory where to store data (default = tempdir())

Value

Expanded PhenoCam data structure or file, including 90 day padding if requested.

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# Overwrites the original file, increasing
# it's file size.
expand_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

# Contracts the file to it's original size, skipping
# two days.
contract_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

## End(Not run)
```

---

grvi *Calculate green-red vegetation index (GRVI)*

Description

The GRVI is defined as the normalized ratio between the red and green channel of a RGB image or digital number triplet. However, the blue channel can be used as well using a weighting factor. As such a paramter vector is provided so different channels / DN can be weighted separately.

Usage

```
grvi(data, par = c(1, 1, 1), internal = TRUE, out_dir = tempdir())
```

Arguments

data	a PhenoCam data file or data frame (when using a file provide a full path if not in the current working directory)
par	grvi parameters (digital number weights)
internal	return a data structure if given a file on disk (TRUE / FALSE = default)
out_dir	output directory where to store data

Value

Inserts a GRVI data column into the provided PhenoCam data structure or file.

Examples

```
## Not run:
# with defaults, outputting a data frame
# with smoothed values, overwriting the original

# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# calculate and append the GRVI for a file (overwrites the original)
grvi(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

# as all functions this also works on a PhenoCam data structure
df <- read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
df <- grvi(df, par = c(1, 1, 0))

## End(Not run)
```

---

list_rois *List all site regions-of-interst (ROIs)*

Description

The ROI list can be helpful in determining which time series to download using 'download_phenocam()'.

Usage

```
list_rois(out_dir = tempdir(), internal = TRUE)
```

Arguments

out_dir	output directory (default = tempdir())
internal	TRUE or FALSE (default = TRUE)

Value

A data frame with ROIs for all available cameras

Examples

```
## Not run:
# download the site meta-data
df <- list_rois()

## End(Not run)
```

`list_sites` *List all site meta-data*

Description

The site list can be helpful in determining which time series to download using 'download_phenocam()'. The site list also includes meta-data concerning plant functional types, general climatological conditions such as mean annual temperature or geographic location.

Usage

```
list_sites(out_dir = tempdir(), internal = TRUE)
```

Arguments

`out_dir`	output directory (default = tempdir())
`internal`	TRUE or FALSE (default = TRUE)

Value

A data frame with meta-data for all available sites.

Examples

```
## Not run:
# download the site meta-data
df <- list_sites()

## End(Not run)
```

`merge_daymet` *Merge Daymet data with a PhenoCam time series*

Description

Combine PhenoCam time series with matching climatological variables from Daymet.

Usage

```
merge_daymet(data, trim = FALSE, internal = TRUE, out_dir = tempdir())
```

Arguments

data	a PhenoCam data file or data structure
trim	logical, trim the daymet data to the length of the PhenoCam time series or include the whole Daymet time series (1980-current). (default = FALSE)
internal	return a data structure if given a file on disk (TRUE / FALSE = default)
out_dir	output directory where to store data (default = tempdir())

Value

A PhenoCam data structure or file which combines PhenoCam time series data with Daymet based climate values (columns will be added).

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# merge data with daymet data
merge_daymet(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

## End(Not run)
```

---

merge_modis *Merge ORNL MODIS data with a PhenoCam time series*

Description

Combine PhenoCam time series with MODIS data for matching dates.

Usage

```
merge_modis(
  data,
  product,
  band,
  trim = FALSE,
  internal = TRUE,
  out_dir = tempdir()
)
```

Arguments

data	a PhenoCam data file or data structure
product	which MODIS product to query (character vector)
band	which MODIS band(s) to include (character vector)
trim	logical, trim the MODIS data to the length of the PhenoCam time series or include the whole Daymet time series (1980-current). (default = FALSE)
internal	return a data structure if given a file on disk (TRUE / FALSE = default)
out_dir	output directory where to store data (default = tempdir())

Value

A PhenoCam data structure or file which combines PhenoCam time series data with MODIS values (columns will be added). Data is queried from the ORNL MODIS subsets service using the 'MODISTools' package, please consult either sources on product and band names.

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# merge data with daymet data
df <-  merge_modis(file.path(tempdir(),
"harvard_DB_1000_3day.csv"),
product = "MOD13Q1",
band = "250m_16_days_NDVI")

## End(Not run)
```

---

normalize_ts *Normalize PhenoCam time series*

Description

Normalize PhenoCam data between 0-1 to to standardize further processing, independent of the relative amplitude of the time series (works on vectors not data frames). For internal use only.

Usage

```
normalize_ts(df, percentile = 90)
```

Arguments

df	a PhenoCam data frame
percentile	percentile value to interprete

Value

A normalized PhenoCam time series.

Examples

```
# Internal function only, should not be used stand-alone.
# As such no documentation is provided.
```

---

optimal_span *Calculates the optimal span for a loess spline*

Description

The optimal span is calculated based upon the bayesian information criterion (BIC).

Usage

```
optimal_span(
  y,
  x = NULL,
  weights = NULL,
  step = 0.01,
  label = NULL,
  plot = FALSE
)
```

Arguments

y	a vector with measurement values to smooth
x	a vector with dates / time steps
weights	optional values to weigh the loess fit with
step	span increment size
label	title to be used when plotting function output
plot	plot visual output of the optimization routine

Value

Returns an optimal span to smooth a provided vector using the ‘loess()‘ smoother.

Examples

```
## Not run:
# Internal function only, should not be used stand-alone.
l <- sin(seq(1,10,0.01))
l <- l + runif(length(l))
optimal_span(l, plot = TRUE)

## End(Not run)
```

`phenocam_explorer` *Starts the phenocamr shiny interface*

Description

The GUI allows you to interactively download data and visualize time series.

Usage

```
phenocam_explorer()
```

Examples

```
## Not run:
# Starts the PhenoCam explorer GUI in a browser
phenocam_explorer()

## End(Not run)
```

`phenophases` *Calculates phenophases*

Description

This routine combines a forward and backward run of transition_dates function to calculate the phenophases in both rising and falling parts of a PhenoCam time series.

Usage

```
phenophases(data, mat, internal = TRUE, out_dir = tempdir(), ...)
```

Arguments

data	a PhenoCam data file (or data frame)
mat	mean annual temperature
internal	return PhenoCam data file or data frame
out_dir	output directory
...	pass parameters to the transition_dates() function

Value

Estimates of transition dates for both rising and falling parts of a PhenoCam time series. All time series are evaluated (gcc_90, gcc_75, etc). The function returns a nested list with UNIX time based values including uncertainties on these estimates and their associated thresholds. When written to disk UNIX dates are converted to YYYY-MM-DD. The nested list has named locations rising and falling, or location 1 and 2 in the list respectivelly.

Examples

```
## Not run:
# downloads a time series
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# read in data as data frame and calculate phenophases
df <- read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
my_dates <- phenophases(df, internal = TRUE)

# print results
print(my_dates)

## End(Not run)
```

---

process_phenocam *Function to post-process PhenoCam time series*

Description

Wrapper around other more basic funtions, in order to generate phenocam data products.

Usage

```
process_phenocam(
  file,
  outlier_detection = TRUE,
  smooth = TRUE,
```

```
  contract = FALSE,
  expand = TRUE,
  truncate,
  phenophase = TRUE,
  snow_flag = FALSE,
  penalty = 0.5,
  out_dir = tempdir(),
  internal = FALSE,
  ...
)
```

Arguments

file	1 or 3-day PhenoCam time series file path
outlier_detection	TRUE or FALSE, detect outliers
smooth	smooth data (logical, default is TRUE)
contract	contract 3-day data upon output (logical, default is TRUE)
expand	expand 3-day data upon input (logical, default is TRUE)
truncate	year (numeric) to which to constrain the output
phenophase	logical, calculate transition dates (default = FALSE)
snow_flag	integrate snow flags?
penalty	how sensitive is the change point algorithm, lower is more sensitve (< 1, default = 0.5)
out_dir	output directory where to store downloaded data (default = tempdir())
internal	allow for the data element to be returned to the workspace
...	additional parameters to be forwarded to the phenophases() function, used internally in the routine

Value

Downloaded files in out_dir of requested time series products, as well as derived phenophase estimates based upon these time series.

Examples

```
## Not run:
# download the first ROI time series for the Harvard PhenoCam site
# at an aggregation frequency of 3-days.
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# read phenocam data into phenocamr data structure
df <- process_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
```

```
## End(Not run)
```

---

read_phenocam *Read PhenoCam time series data*

Description

Reads PhenoCam data into a nested list, preserving header data and critical file name information.

Usage

```
read_phenocam(filename)
```

Arguments

filename	a PhenoCam data file

Value

A nested data structure including site meta-data, the full header and the data as a 'data.frame()'.

Examples

```
## Not run:
# download demo data (do not smooth)
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3",
                  smooth = FALSE)

# read the phenocamo data file
df = read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

# print data structure
print(summary(df))

# write the phenocamo data file
write_phenocam(df, out_dir = tempdir())

## End(Not run)
```

smooth_ts *Smooth a PhenoCam time series*

Description

Smooths time series iteratively using a Akaike information criterion (AIC) to find an optimal smoothing parameter and curve.

Usage

```
smooth_ts(
  data,
 metrics = c("gcc_mean", "gcc_50", "gcc_75", "gcc_90", "rcc_mean", "rcc_50", "rcc_75",
    "rcc_90"),
  force = TRUE,
  internal = TRUE,
  out_dir = tempdir()
)
```

Arguments

data	a PhenoCam data file or data structure
metrics	which metrics to process, normally all default ones
force	TRUE / FALSE, force reprocessing?
internal	return a data structure if given a file on disk (TRUE / FALSE = default)
out_dir	output directory where to store data

Value

An PhenoCam data structure or file with optimally smoothed time series objects added to the original file. Smoothing is required for 'phenophase()' and 'transition_dates()' functions.

Examples

```
## Not run:
# with defaults, outputting a data frame
# with smoothed values, overwriting the original

# download demo data (do not smooth)
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3",
                  smooth = FALSE)

# smooth the downloaded file (and overwrite the original)
```

```
smooth_ts(file.path(tempdir(),"harvard_DB_1000_3day.csv"))

# the function also works on a PhenoCam data frame
df <- read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
df <- smooth_ts(df)

## End(Not run)
```

`transition_dates` *Calculates transition dates for a PhenoCam time series*

Description

Segments of a PhenoCam time series and calculates threshold based transition dates for all segments. This function is rarely called stand alone and 'phenophases()' should be preferred when evaluating PhenoCam time series.

Usage

```
transition_dates(
  data,
  lower_thresh = 0.1,
  middle_thresh = 0.25,
  upper_thresh = 0.5,
  percentile = 90,
  penalty = 0.5,
  seg_length = 14,
  reverse = FALSE,
  plot = FALSE
)
```

Arguments

`data`	a PhenoCam data file or data structure
`lower_thresh`	the minimum threshold used (default = 0.1)
`middle_thresh`	the middle threshold used (default = 0.25)
`upper_thresh`	the maximum threshold used (default = 0.5)
`percentile`	time series percentiles to process (mean, 50, 75, 90)
`penalty`	how sensitive is the algorithm, lower is more sensitve (< 1)
`seg_length`	minimum length of a segment to be evaluated
`reverse`	flip the direction of the processing
`plot`	plot for debugging purposes

Value

Transition date estimates in UNIX time, including uncertainties and the threshold values estimated for each section of a time series.

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# read the data and calculate transition dates
df <- read_phenocam(file.path(tempdir(),"harvard_DB_1000_3day.csv"))
my_dates <- transition_dates(df,
                             lower_thresh = 0.1,
                             middle_thresh = 0.25,
                             upper_thresh = 0.5,
                             percentile = 90,
                             reverse = FALSE,
                             plot = FALSE)

## End(Not run)
```

`truncate_phenocam` *Truncate a PhenoCam time series*

Description

The 'expand_phenocam()' function provides a similar functionality and is prefered. This function remains as it might serve a purpose to some. Might be deprecated in the future.

Usage

```
truncate_phenocam(data, year = 2015, internal = TRUE, out_dir = tempdir())
```

Arguments

`data`	a PhenoCam file or data frame
`year`	the last valid year, discard the rest
`internal`	return a data structure if given a file on disk (`TRUE` / `FALSE` = default)
`out_dir`	output directory where to store data (default = tempdir())

Value

A truncated PhenoCam data structure or file, with data limited to the year specified.

Examples

```
## Not run:
# download demo data
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3")

# overwrites the original file, increasing
# decreasing the file size, with given year as maximum.
truncate_phenocam(file.paste(tempdir(),"harvard_DB_1000_3day.csv"),
                  year = 2015)

## End(Not run)
```

`write_phenocam` *Write a phenocamr data structure to file*

Description

Writes a nested data structure of class phenocamr to file, reconstructing the original data structure from included headers and data components.

Usage

```
write_phenocam(df = NULL, out_dir = tempdir())
```

Arguments

`df`	a nested data structure of class phenocamr
`out_dir`	output directory where to store data

Value

writes PhenoCam data structure to file, retains proper header info and inserts a processing time stamp.

Examples

```
## Not run:
# download demo data (do not smooth)
download_phenocam(site = "harvard$",
                  veg_type = "DB",
                  roi_id = "1000",
                  frequency = "3",
                  smooth = FALSE)
```

```
# read the phenocamo data file
df = read_phenocam(file.paste(tempdir(),"harvard_DB_1000_3day.csv"))

# print data structure
print(summary(df))

# write the phenocamo data file
write_phenocam(df, out_dir = tempdir())

## End(Not run)
```

Index