

Package ‘cercospoRa’

December 13, 2024

Title Process Based Epidemiological Model for Cercospora Leaf Spot of Sugar Beet

Version 0.0.1

Description Estimates sugar beet canopy closure with remotely sensed leaf area index and estimates when action might be needed to protect the crop from a Leaf Spot epidemic with a negative prognosis model based on published models.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0), remotes, stats, spelling

Config/testthat/edition 3

Imports data.table, lubridate, sf, terra, minpack.lm, circular

VignetteBuilder knitr

Depends R (>= 4.0.0)

LazyData true

URL <https://paul.melloy.com.au/cercospoRa/>,
<https://github.com/PaulMelloy/cercospoRa>,
<http://paul.melloy.com.au/cercospoRa/>

BugReports <https://github.com/PaulMelloy/cercospoRa/issues>

NeedsCompilation no

Author Paul Melloy [aut, cre, trl, cph]
(<<https://orcid.org/0000-0003-4253-7167>>, Translated equations from literature and developed package.),
Rene Heim [aut, cph] (<<https://orcid.org/0000-0002-0666-2588>>, Code review and development of leaf area index model),
Nathan Okole [aut, ctb, cph] (<<https://orcid.org/0000-0002-5375-2420>>, Code review and development of leaf area index model),
Facundo Ramón Ispizua Yamati [dct, ctb]

(<<https://orcid.org/0000-0001-5775-3554>>, Code review),
 Anne-Katrin Mahlein [ctb, cph]
 (<<https://orcid.org/0000-0003-1091-3690>>)

Maintainer Paul Melloy <paul@melloy.com.au>

Repository CRAN

Date/Publication 2024-12-13 09:20:02 UTC

Contents

calc_c_closure	2
calc_DIV	3
calc_epidemic_onset	4
calc_epidemic_onset_from_image	5
calc_r_x0	6
format_weather	7
latent_period	11
read_sb_growth_parameter	12
temperature_index	13
weathr	14
Index	15

calc_c_closure	<i>Calculate canopy closure date</i>
----------------	--------------------------------------

Description

Calculate canopy closure date

Usage

```
calc_c_closure(param_rxt, x1 = 1.3, k = 6)
```

Arguments

param_rxt	Output of the function <code>calc_r_x0</code> , which produces a list containing parameters that are necessary to calculate <code>c_closure</code> . These parameters are <code>r</code> , the growth rate, <code>x0</code> , the initial LAI value, and <code>t0</code> , the initial date and the associated dates
x1	LAI value at which 90% canopy closure is reached. It is fixed to 1.3 for sugar beet.
k	carrying capacity, which is the maximum LAI that can be attained. This value can be cultivar-dependent. The default is fixed to 6

Details

Calculates canopy closure dates from LAI and growth rate parameters

Value

c_closure date at which 90% canopy closure is reached at each location.

Examples

```
img_dir <- system.file("extdata", "uav_img", package = "cercospora")
epidemic_onset_param <-
  read_sb_growth_parameter(img_files = list.files(img_dir, pattern = "tif",
                                                full.names = TRUE),
                          img_dates = as.POSIXct(
                            c("2022-06-14", "2022-06-28"), tz = "UTC"),
                          target_res = 10)
param_rxt <- calc_r_x0(epidemic_onset_param,
                      min_r = 0.02,
                      max_r = 0.05,
                      k = 6)
c_closure <- calc_c_closure(param_rxt,
                           x1 = 1.3,
                           k=6 )
```

 calc_DIV

Calculate daily infection values

Description

This function calculates the daily infection values for *Cercospora beticola* on sugar beet. Functions were adapted from Wolf and Verreet (2005) and Wolf et al (2001)

Usage

```
calc_DIV(date_time, Tm, RH, rain, dat)
```

Arguments

date_time	POSIX_ct, date time the weather recording was taken
Tm	numeric, temperature, in Celsius' at time increment in date_time
RH	numeric, relative humidity (%) at time increment in date_time
rain	numeric, volume of rain in millimetres recorded between time recordings
dat	data.frame, containing column names "times","temp","rh","rain" with each of the respective arguments for input. provided as a convenience

Value

data.table, with probability of infection for each day, between 0 and 1 Undertaken with two methods by Wolf and Verreet (2005)

References

Wolf, P. F. J., and J. A. Verreet. "Factors Affecting the Onset of Cercospora Leaf Spot Epidemics in Sugar Beet and Establishment of Disease-Monitoring Thresholds." *Phytopathology* 95, no. 3 (March 2005): 269–74. <https://doi.org/10.1094/PHYTO-95-0269>.

Wolf, P. F. J., M. Heindl, and J. A. Verreet. "Influence of Sugar Beet Leaf Mass Development on Predisposition of the Crop to Cercospora Beticola (Sacc.)." *Journal of Plant Diseases and Protection* 108, no. 6 (2001): 578–92.

Examples

```
date_t <- Sys.time() + seq(0, 179 * 60 * 10, (60 * 10))
Tm <- rnorm(180,20,10)
RH <- runif(180,min = 40,90)
rain <- rbinom(180,1,0.1) * runif(180,0.1,20)

DIV1 <- calc_DIV(
  date_time = date_t,
  Tm = Tm,
  RH = RH,
  rain = rain
)
```

calc_epidemic_onset *Calculate epidemic onset*

Description

Calculate epidemic onset

Usage

```
calc_epidemic_onset(start, end, c_closure, weather, cultivar_sus = 5)
```

Arguments

start	POSIXct, start date in which to begin calculating the epidemic onset, if not specified, the first date in the weather data will be used.
end	POSIXct, end date, last date to complete calculating the epidemic onset, if not specified, the last date in the weather data will be used.
c_closure	POSIXct formatted date to start the model running the model This is usually at canopy closure (Wolf)
weather	data.table, formatted with format_weather
cultivar_sus	character, susceptibility of the cultivar in "R" resistant, "S" susceptible, "MR" moderately resistant etc.

Value

If the input weather is conducive for epidemic, the function returns a POSIX_ct date when epidemic commences. If no epidemic occurs, a numeric, proportion indicating the progress an epidemic is returned

Examples

```
wethr <- read.csv(system.file("extdata", "clean_weather.csv",
                             package = "cercospoRa"))
wethr <- format_weather(wethr,time_zone = "UTC")

calc_epidemic_onset(start = as.POSIXct("2022-04-25",tz = "UTC"),
                    end = as.POSIXct("2022-09-30",tz = "UTC"),
                    c_closure = as.POSIXct("2022-07-01",tz = "UTC"),
                    weather = wethr)
```

```
calc_epidemic_onset_from_image
```

Calculate epidemic onset from image

Description

Calculate epidemic onset from image

Usage

```
calc_epidemic_onset_from_image(
  start,
  end,
  c_closure,
  weather,
  cultivar_sus = 5
)
```

Arguments

start	POSIXct, start date in which to begin calculating the epidemic onset, if not specified, the first date in the weather data will be used.
end	POSIXct, end date, last date to complete calculating the epidemic onset, if not specified, the last date in the weather data will be used.
c_closure	map of canopy closure dates formatted as number of days since 1970-01-01. It defines the date to start running the model (Wolf)
weather	data.table, formatted with format_weather
cultivar_sus	character, susceptibility of the cultivar in "R" resistant, "S" susceptible, "MR" moderately resistant etc.

Value

If the input weather is conducive for epidemic, the function returns a POSIX_ct date when epidemic commences. If no epidemic occurs, a numeric, proportion indicating the progress an epidemic is returned

Examples

```
wethr <- read.csv(system.file("extdata", "clean_weather.csv",
                             package = "cercospoRa"))
wethr <- format_weather(wethr,time_zone = "UTC")

img_dir <- system.file("extdata", "uav_img",package = "cercospoRa")

epidemic_onset_param <-
  read_sb_growth_parameter(
    list.files(img_dir,pattern = "tif",
              full.names = TRUE),
    img_dates = as.POSIXct(c("2022-06-14","2022-06-28"),
                           tz = "UTC"),
    10)
param_rxt <- calc_r_x0(epidemic_onset_param,
                      min_r = 0.02,
                      max_r = 0.05,
                      k = 6)
c_closure <- calc_c_closure(param_rxt,
                           x1 = 1.3,
                           k=6 )
# this takes about 20 sec to run
epidemic_onset_map <- calc_epidemic_onset_from_image(start = as.POSIXct("2022-04-25",tz = "UTC"),
                                                    end = as.POSIXct("2022-09-30",tz = "UTC"),
                                                    c_closure = c_closure,
                                                    weather = wethr)

terra::plot(epidemic_onset_map)
```

 calc_r_x0

Calculate growth rate and LAI at t0

Description

Calculate growth rate and LAI at t0

Usage

```
calc_r_x0(param_r, min_r = 0.02, max_r = 0.05, k = 6)
```

Arguments

param_r	Output of the function read_sb_growth_parameter , which produces a list containing the LAI images and the associated dates
min_r	minimum growth rate for sugar beet. Default min_r is fixed to 0.02 to ensure that the growth rate at the inflection point of the sigmoid is at least 1 unit of LAI per month.
max_r	maximum growth rate for sugar beet. Default max_r is fixed to 0.05 to ensure that the growth rate at the inflection point of the sigmoid is at most 2.5 units of LAI per month.
k	carrying capacity, which is the maximum LAI that can be attained. This value can be cultivar-dependent. The default is fixed to 6

Details

Fits a non-linear model to remotely sensed LAI values and estimates the leaf area index (LAI) and growth rate r at the start of the time window t_0 .

Value

param_rxt: list containing parameters that are necessary to calculate c_closure. These parameters are r , the growth rate, x_0 , the initial LAI value, and t_0 , the initial date.

Examples

```
img_dir <- system.file("extdata", "uav_img", package = "cercospora")
epidemic_onset_param <-
  read_sb_growth_parameter(img_files = list.files(img_dir, pattern = "tif",
                                                full.names = TRUE),
                          img_dates = as.POSIXct(
                            c("2022-06-14", "2022-06-28"), tz = "UTC"),
                          target_res = 10)
param_rxt <- calc_r_x0(epidemic_onset_param,
                      min_r = 0.02,
                      max_r = 0.05,
                      k = 6)
```

format_weather

Format weather data into a standardised format.

Description

Formats raw weather data into an object suitable for use in the affiliated spore dispersal packages such as ascotracer and blackspot.sp. This standardised data format ensures that the supplied weather data meet the requirements for functions in the aforementioned packages. Input weather data expects a 'long' data format, where each line is an observation at a point in time.

Usage

```
format_weather(
  w,
  YYYY = NULL,
  MM = NULL,
  DD = NULL,
  hh = NULL,
  mm = NULL,
  ss = NULL,
  POSIXct_time = NULL,
  time_zone = NULL,
  temp,
  rain,
  rh,
  ws,
  wd,
  wd_sd,
  station,
  lon = NULL,
  lat = NULL,
  lonlat_file = NULL,
  data_check = TRUE
)
```

Arguments

w	a data.frame object of weather station data for formatting.
YYYY	Column name character or index in w that refers to the year when the weather was logged.
MM	Column name character or index in w that refers to the month (numerical) when the weather was logged.
DD	Column name character or index in w that refers to the day of month when the weather was logged.
hh	Column name character or index in w that refers to the hour (24 hour) when the weather was logged.
mm	Column name character or index in w that refers to the minute when the weather was logged.
ss	Column name character or index in w that refers to the second when the weather was logged.
POSIXct_time	Column name character or index in w which contains a POSIXct formatted time. This can be used instead of arguments YYYY, MM, DD, hh, mm.
time_zone	Time zone (Olsen time zone format) character where the weather station is located. May be in a column or supplied as a character string. Optional, see also r . See details.
temp	Column name character or index in x that refers to temperature in degrees Celsius.

rain	Column name character or index in w that refers to rainfall in millimetres.
rh	Column name character or index in w that refers to relative humidity as a percentage.
ws	Column name character or index in w that refers to wind speed in km / h.
wd	Column name character or index in w that refers to wind direction in degrees.
wd_sd	Column name character or index in w that refers to wind speed columns standard deviation. This is only applicable if weather data is already summarised to hourly increments. See details.
station	Column name character or index in w that refers to the weather station name or identifier. See details.
lon	Column name character or index in w that refers to weather station's longitude. See details.
lat	Column name character or index in w that refers to weather station's latitude. See details.
lonlat_file	A file path (character) to a CSV which included station name/id and longitude and latitude coordinates if they are not supplied in the data. Optional, see also lon and lat.
data_check	If TRUE, it checks for NA values in all 'rain', 'temp', 'rh', 'wd' and 'ws' data and if any values which are unlikely. Use a character vector of variable names, (with any of 'rain', 'temp', 'rh', 'wd' or 'ws') to check data from specific variables. If FALSE it ignores all variables and could cause subsequent models using this data to fail.

Details

time_zone The time-zone in which the time was recorded. All weather stations in w must fall within the same time-zone. If the required stations are located in differing time zones, `format_weather()` should be run separately on each object, then data can be combined after formatting.

wd_sd If weather data is provided in hourly increments, a column with the standard deviation of the wind direction over the hour is required to be provided. If the weather data are sub-hourly, the standard deviation will be calculated and returned automatically.

lon, lat and lonlat_file If w provides longitude and latitude values for station locations, these may be specified in the lon and lat columns. If the coordinates are not relevant to the study location NA can be specified and the function will drop these column variables. If these data are not included, (NULL) a separate file may be provided that contains the longitude, latitude and matching station name to provide station locations in the final `epiphy.weather` object that is created by specifying the file path to a CSV file using `lonlat_file`.

Value

A `epiphy.weather` object (an extension of **data.table**) containing the supplied weather aggregated to each hour in a suitable format for use with disease models. Depending on the input weather, classes will be given to the output object to indicate which models it meets the data requirements for. Some of the columns returned are as follows:

times: Time in POSIXct format with "UTC" time-zone
rain: Rainfall in mm
temp: Temperature in degrees Celsius
ws: Wind speed in km / h
wd: Wind direction in compass degrees
wd_sd: Wind direction standard deviation in compass degrees
lon: Station longitude in decimal degrees
lat: Station latitude in decimal degrees
station: Unique station identifying name
YYYY: Year
MM: Month
DD: Day
hh: Hour
mm: Minute

Examples

```

# load the weather data to be formatted
weather_station_data <-
  read.csv(system.file("extdata",
                      "scaddan_weather.csv",
                      package = "cercoRa"))
weather_station_data$Local.Time <-
  as.POSIXct(weather_station_data$Local.Time, format = "%Y-%m-%d %H:%M:%S",
            tz = "UTC")

weather <- format_weather(
  w = weather_station_data,
  POSIXct_time = "Local.Time",
  ws = "meanWindSpeeds",
  wd_sd = "stdDevWindDirections",
  rain = "Rainfall",
  temp = "Temperature",
  wd = "meanWindDirections",
  lon = "Station.Longitude",
  lat = "Station.Latitude",
  station = "StationID",
  time_zone = "UTC"
)

# Reformat saved weather

# Create file path and save data
file_path_name <- paste(tempdir(), "weather_saved.csv", sep = "\\")
write.csv(weather, file = file_path_name,
          row.names = FALSE)

# Read data back in to
weather2 <- read.csv(file_path_name, stringsAsFactors = FALSE)
  
```

```
# reformat the data to have appropriate column classes and data class
weather2 <- format_weather(weather2,
                           time_zone = "UTC")
unlink(file_path_name) # remove temporary weather file
```

latent_period	<i>Calculate Cercospora Latent Period</i>
---------------	---

Description

Calculate Cercospora Latent Period

Usage

```
latent_period(Tm, ref = "wolf")
```

Arguments

Tm	numeric, Average temperature in degrees Celsius for a hour of the day
ref	character, method for calculating latent period. Default is "wolf" also available "jarroudi". See references for where formulas were used

Details

Calculates the latent period for *Cercospora beticola* infections on sugar beet. Note the published formula in Wolf and Verreet (2005) contains an error in the exponent. $e(0.35 \times (T_m - 10))$ should be $e(-0.35 \times (T_m - 10))$. See issue #22 on Github for additional information.

Value

numeric, Latent period for a given hour. ref = "jarroudi" returns latent period in days.

References

Wolf, P. F. J., and J. A. Verreet. "Factors Affecting the Onset of Cercospora Leaf Spot Epidemics in Sugar Beet and Establishment of Disease-Monitoring Thresholds." *Phytopathology* 95, no. 3 (March 2005): 269–74. <https://doi.org/10.1094/PHYTO-95-0269>.

El Jarroudi, Moussa, Fadia Chairi, Louis Kouadio, Kathleen Antoons, Abdoul-Hamid Mohamed Sallah, and Xavier Fettweis. "Weather-Based Predictive Modeling of Cercospora Beticola Infection Events in Sugar Beet in Belgium." *Journal of Fungi* 7, no. 9 (September 18, 2021): 777. <https://doi.org/10.3390/jof7090777>.

temperature_index	<i>Get temperature index</i>
-------------------	------------------------------

Description

Temperature index is a proportional representation of the latent period. Temperatures at or above the optimum temperature for the disease cycle will yield 1 and the lower the temperature the infinitely longer the latent period.

Usage

```
temperature_index(Tm, opt_Tm = 21)
```

Arguments

Tm	numeric, temperature for any given hour
opt_Tm	numeric, the lowest temperature optimum at which all temperatures above will have the same (fastest) latent period.

Details

For equations and original documentation, see Wolf and Verreet (2005) and Wolf et al. (2001)

Value

numeric, proportion representing the speed of the latent period in relation to the temperature optimum

References

Wolf, P. F. J., and J. A. Verreet. "Factors Affecting the Onset of Cercospora Leaf Spot Epidemics in Sugar Beet and Establishment of Disease-Monitoring Thresholds." *Phytopathology* 95, no. 3 (March 2005): 269–74. <https://doi.org/10.1094/PHYTO-95-0269>.

Wolf, P. F. J., F.-J. Weis, and J.-A. Verreet. "Threshold Values as Indicators of Fungicide Treatments for the Control of Leaf Blotching Caused by *Cercospora Beticola* (Sacc.) in Sugar Beets." *Journal of Plant Diseases and Protection* 108, no. 3 (2001): 244–57.

weathr

Weather station data

Description

A unformatted weather station dataset from Germany

Author(s)

Institute for Sugar Beet Research - IFZ

Source

Climavi One + Climavi Soil, Agvolution GmbH, Göttingen, Germany A data frame with 8,016 rows and 15 columns: Weather station that accompanied field trial described in cercospora manuscript

Station Identifier

Datum Date

Stunde hour

Minute minute

T-005 Temperature in degrees Celcius at 5cm

T-200 Temperature in degrees Celcius at 200cm

T-020 Temperature in degrees Celcius at 20cm

F-200 Percentage Relative humidity at 200cm

GS200 Global radiation (W/m²) at 200cm

WR200 Wind direction degrees at 200cm

N100 Precipitation (mm) at 100cm

WG200 Wind Speed (m/s) at 200cm ...

Index

`calc_c_closure`, [2](#)
`calc_DIV`, [3](#)
`calc_epidemic_onset`, [4](#)
`calc_epidemic_onset_from_image`, [5](#)
`calc_r_x0`, [2](#), [6](#)

`data.frame`, [8](#)

`format_weather`, [4](#), [5](#), [7](#)

`latent_period`, [11](#)

`read_sb_growth_parameter`, [7](#), [12](#)

`temperature_index`, [13](#)

`weathr`, [14](#)