

Package: epicrop (via r-universe)

May 29, 2026

Type Package

Title SEIR Simulation Modelling of Crop Diseases

Version 1.0.0.9000

Description Generic simulation modelling of crop diseases using a Susceptible-Exposed-Infectious-Removed ('SEIR') model. This type of model was first described in botanical epidemiology by Zadoks (1971) <[doi:10.1094/Phyto-61-600](https://doi.org/10.1094/Phyto-61-600)> and implemented in the 'EPIRICE' model by Savary et al. (2012) <[doi:10.1016/j.cropro.2011.11.009](https://doi.org/10.1016/j.cropro.2011.11.009)> detailing the model's development, use and results from modelling global unmanaged epidemics of rice diseases for five major rice diseases, bacterial blight, brown spot, leaf blast, sheath blight and tungro. Specific functions are provided to simulate all of these diseases with a generic 'SEIR' model function that is suitable for parameterising for use with other pathosystems as demonstrated in Savary et al. 2012 and Kim et al. 2015 <[doi:10.1016/j.agrformet.2015.01.011](https://doi.org/10.1016/j.agrformet.2015.01.011)> along with two wheat diseases, brown (leaf) rust and Septoria tritici blotch as published in Savary et al. 2015 <[doi:10.1007/s10658-015-0650-7](https://doi.org/10.1007/s10658-015-0650-7)>. Functions for fetching and formatting weather data from the NASA POWER database are also provided to simplify the process of running simulations.

License GPL (>= 3)

URL <https://codeberg.org/adamhsparks/epicrop>,
<https://adamhsparks.codeberg.page/epicrop>

BugReports <https://codeberg.org/adamhsparks/epicrop/issues>

Depends R (>= 4.1.0)

Imports cli, data.table, digest, memoise, nasapower (>= 4.0), Rcpp,
rlang, stats

Suggests cropgrowdays, ggplot2, ggrepel, ggspatial, knitr, rmarkdown,
roxyglobals, spelling, testthat, vcr (>= 0.6.0), withr

LinkingTo Rcpp

VignetteBuilder knitr
ByteCompile TRUE
Config/Needs/build moodymudskipper/devtag
Config/roxygen2/version 8.0.0
Config/roxyglobals/filename globals.R
Config/roxyglobals/unique FALSE
Config/testthat/edition 3
Config/testthat/parallel true
Encoding UTF-8
Language en-US
Roxygen list(markdown = TRUE, roclets = c("`collate", "`namespace",
 "`rd", "`roxyglobals::global_roclet", "`devtag::dev_roclet"))
X-schema.org-applicationCategory Tools
X-schema.org-keywords rice, wheat, botanical-epidemiology, disease,
 model, seir, agricultural-modelling, agricultural-modeling,
 crop-protection, agricultural-research, model, modelling,
 modeling, seir-model, plant-disease
Config/pak/sysreqs libssl-dev libx11-dev
Repository <https://adamhspark.r-universe.dev>
Date/Publication 2026-05-29 04:47:48 UTC
RemoteUrl <https://codeberg.org/adamhspark/epicrop>
RemoteRef HEAD
RemoteSha 1dd18d5ec0906f0a961acecc1cd500af3aa2b3b7

Contents

audpc_dt	3
build_epicrop_emergence	3
disease_models	4
fetch_epicrop_weather_list	7
format_wth	9
get_audpc	11
get_wth	12
run_epicrop_model	13
seir	14
Index	20

audpc_dt	<i>Extract AUDPC Summary as data.table</i>
----------	--

Description

Extracts Area Under Disease Progress Curve (AUDPC) values along with location and emergence date information from an `epicrop.sim` object.

Usage

```
audpc_dt(x)
```

Arguments

`x` An `epicrop.sim` object from `seir()` or disease model functions.

Value

A `data.table::data.table()` with columns:

- `lat`: Latitude (if available)
- `lon`: Longitude (if available)
- `AUDPC`: Area under disease progress curve
- `emergence`: Emergence date (as character)

<code>build_epicrop_emergence</code>	<i>Build emergence dates from year × month_day combinations</i>
--------------------------------------	---

Description

Construct a vector of emergence dates by combining a set of years with a set of month-day strings (e.g., "-06-01"), returning a sorted `IDate` vector tagged with class `epicrop.emerge` for downstream use with `run_epicrop_model()`.

Usage

```
build_epicrop_emergence(years, month_day)
```

Arguments

`years` Integer vector of years (e.g., 2001:2003).

`month_day` Character vector of month-day strings in the form "-MM-DD" (e.g., `c("-06-01", "-06-14")`). If a string is supplied without a leading dash (e.g., "06-01"), callers should normalize it in advance; this helper expects the leading - as part of the format "YYYY-MM-DD".

Value

A sorted vector of [data.table::IDate](#) emergence dates with class `epicrop.emerge` (*i.e.*, `inherits(x, "epicrop.emerge")` will be TRUE).

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

See Also

[run_epicrop_model\(\)](#), [fetch_epicrop_weather_list\(\)](#)

Examples

```
# Basic usage: combine years with multiple month-day patterns
years <- 2001:2002
month_day <- c("-06-01", "-06-14", "-06-30")

em <- build_epicrop_emergence(years, month_day)
em

# Check class and structure
inherits(em, "epicrop.emerge") # TRUE
head(em)                       # First few dates (IDate)
length(em)                      # 6 (2 years × 3 month-days)

# Typical downstream: pass to run_epicrop_model(...)
# run_epicrop_model(bacterial_blight, em, wth_list, output = "audpc")
```

disease_models

Rice and Wheat Disease Models

Description

Helper functions with pre-loaded parameter tables for optimal performance. Custom parameter tables can be provided to override defaults.

Usage

```
bacterial_blight(
  wth,
  emergence,
  RcA = NULL,
  RcT = NULL,
  RRS = NULL,
  RRG = NULL,
  ...
```

```
)  
  
brown_spot(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)  
  
leaf_blast(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)  
  
sheath_blight(  
  wth,  
  emergence,  
  RcA = NULL,  
  RcT = NULL,  
  RRS = NULL,  
  RRG = NULL,  
  ...  
)  
  
tungro(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)  
  
modified_kim_leaf_blast(  
  wth,  
  emergence,  
  RcA = NULL,  
  RcT = NULL,  
  RRS = NULL,  
  RRG = NULL,  
  ...  
)  
  
modified_kim_sheath_blight(  
  wth,  
  emergence,  
  RcA = NULL,  
  RcT = NULL,  
  RRS = NULL,  
  RRG = NULL,  
  ...  
)  
  
leaf_rust(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)  
  
s_tritici_blotch(  
  wth,  
  emergence,  
  RcA = NULL,  
  RcT = NULL,  
  RRS = NULL,  
  RRG = NULL,  
  ...  
)
```

```

)
bb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
bs(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
lb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
sb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
tg(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
mk_lb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
mk_sb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
lr(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
brown_rust(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)
stb(wth, emergence, RcA = NULL, RcT = NULL, RRS = NULL, RRG = NULL, ...)

```

Arguments

`wth` a data.frame of class `epicrop.wth` from either `get_wth()` or `format_wth()` containing weather on a daily time-step with the following field names:

Field Name	Value
<i>YYYYMMDD</i>	Date as Year Month Day (ISO8601)
<i>DOY</i>	Consecutive day of year, commonly called "Julian date"
<i>TEMP</i>	Mean daily temperature (degrees C)
<i>RHUM</i>	Mean daily relative humidity (%)
<i>RAIN</i>	Mean daily rainfall (mm)
<i>TMIN</i>	Optional Minimum daily temperature (degrees C), see TMIN/TMAX

Details

TMAX | **Optional** Maximum daily temperature (degrees C), see TMIN/TMAX
LAT | **Optional** latitude of weather observation, see LAT/LON Details
LON | **Optional** longitude of weather observation, see LAT/LON Details

`emergence` Expected date of plant emergence (or transplanting for rice) entered in YYYY-MM-DD format (character).

`RcA` Optional custom age modifier table (2-column matrix).

`RcT` Optional custom temperature modifier table (2-column matrix).

`RRS` Optional custom senescence rate (scalar, vector, or function).

`RRG` Optional custom growth rate (scalar, vector, or table).

`...` Other parameters passed to `seir()` allowing the user to override the default values from published papers.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

References

- Savary, S., Nelson, A., Willocquet, L., Pangga, I., and Aunario, J. (2012). Modeling and mapping potential epidemics of rice diseases globally. *Crop Protection*, 34: 6-17. doi:10.1016/j.cropro.2011.11.009
- Kim, K. H., Cho, J., Lee, Y. H., & Lee, W. S. (2015). Predicting potential epidemics of rice leaf blast and sheath blight in South Korea. *Agricultural and Forest Meteorology*, 203: 191-207. doi:10.1016/j.agrformet.2015.01.011
- Savary, S., Stetkiewicz, S., Brun, F., & Willocquet, L. (2015). Modelling and mapping potential epidemics of wheat diseases. *European Journal of Plant Pathology*, 142: 771-790. doi:10.1007/s1065801506507

fetch_epicrop_weather_list

Fetch weather once (single or multiple locations) and split by calendar year

Description

Retrieve one or more continuous blocks of weather for one or more locations from given start dates and durations, then split each result into a named list of per-year tables. Each (location, start, duration) result is an epicrop.wth.list. The top-level return is an epicrop.wth.bundle when multiple requests are processed, otherwise a single epicrop.wth.list.

Usage

```
fetch_epicrop_weather_list(
  lonlat,
  start_date,
  duration,
  years = NULL,
  mode = "pairwise",
  location_names = NULL
)
```

Arguments

- | | |
|------------|--|
| lonlat | Numeric length-2 vector c(lon, lat), OR multiple locations as: numeric vector of length 2n, a 2-column matrix/data.frame, or a list of numeric length-2 vectors. |
| start_date | A starting date (character "YYYY-MM-DD", Date , or data.table::IDate) or a month-day string like "06-30" (leading "-" optional) used together with years. |
| duration | Integer(s). Number of days to retrieve from start_date (inclusive). Single value or vector (recycled). |

years	Optional integer vector of years. Required if start_date contains any month-day values without a year.
mode	Character scalar: "pairwise" (default) pairs locations and dates by position (recycling length-1), or "cross" forms all combinations (locations × dates).
location_names	Optional character names for locations (length = number of unique locations).

Details

Flexible inputs:

- lonlat: a single numeric vector, `c(lon, lat)`; or multiple locations as a numeric vector of length `2n`, or a 2-column matrix/data.frame, or a list of `list(c(lon, lat))`.
- start_date: a single "YYYY-MM-DD" (or Date/IDate), or a vector of the same; or a month-day string like "06-30" or "6-30" (leading "-" optional) provided together with years.
- duration: single integer or a vector (recycled); interpreted as inclusive days to fetch (`end = start + duration - 1`).
- years: only used when start_date lacks a year. We compute a **cover window** per distinct month-day: `start = min(paste(years, md))`, `end = max(paste(years, md)) + (duration - 1)`.

Value

- If only one (location, start, duration) is requested: an `epicrop.wth.list`.
- Otherwise: an `epicrop.wth.bundle` (a named list of `epicrop.wth.list`).

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

Examples

```
# Single location, month-day start date (any of "6/30", "-06-30", "06-30")
w1 <- fetch_epicrop_weather_list(
  lonlat    = c(121.255669, 14.16742),
  start_date = "6/30",
  duration   = 120L,
  years     = 2000:2001
)

# ISO start date
w2 <- fetch_epicrop_weather_list(
  lonlat    = c(121.255669, 14.16742),
  start_date = "2000-06-30",
  duration   = 120L
)

# Multiple locations × multiple dates (cross mode), mixed inputs
locations <- rbind(
  "IRRI_ZES" = c(121.255669, 14.16742),
  "Metro_Manila" = c(120.985, 14.6042)
```

```

)
w3 <- fetch_epicrop_weather_list(
  lonlat    = locations,
  start_date = c("2000-06-30", "6-30"),
  duration  = 120L,
  years     = 2000:2001,
  mode      = "cross"
)

# Multiple start dates for same location across years (efficient API calls)
w4 <- fetch_epicrop_weather_list(
  lonlat    = c(121.255669, 14.16742),
  start_date = c("-06-01", "-06-14", "-06-30"),
  duration  = 90L,
  years     = 2001:2003
)
# This makes only 3 API calls (one per year), fetching from earliest to
# latest start date + duration

```

format_wth

Format Weather Data for Use in seir() Simulations

Description

Formats an existing `data.frame()` of weather data for use in **epicrop**. Only the data necessary for use in the model is returned and the columns are renamed to values expected by the model. The data are assumed to be on a daily time-step. For converting sub-daily weather data to daily, see `cropprowdays::daily_mean()`.

Usage

```
format_wth(wth, date, temp, rhum, rain, tmin, tmax, lat, lon)
```

Arguments

wth	A <code>data.frame()</code> object containing the weather data for formatting.
date	The column name indicating the field containing the date values.
temp	The column name indicating the field containing the average daily temperature values.
rhum	The column name indicating the field containing the average daily relative humidity.
rain	The column name indicating the field containing the total daily rainfall.
tmin	The column name indicating the field containing the minimum daily temperature (optional).
tmax	The column name indicating the field containing the maximum daily temperature.

lat	The column name indicating the field containing the latitude of the observation data (optional).
lon	The column name indicating the field containing the longitude of the observation data (optional).

Details

Required values:

Weather data must be a daily time-step and contain the following values,

- full date including year, month and day,
- average daily temperature,
- average daily relative humidity,
- total daily rainfall

Optional values:

The following values are optional and the model can be run by default without them.

- average minimum daily temperature (only used if `simple_wetness == FALSE`)
- average maximum daily temperature (only used if `simple_wetness == FALSE`)
- Latitude (only used if `simple_wetness == FALSE` and for mapping outputs)
- Longitude (only used for mapping outputs)

Field Name	Value
<i>YYYYMMDD</i>	Date as Year Month Day (ISO8601)
<i>DOY</i>	Consecutive day of year, commonly called "Julian date"
<i>TEMP</i>	Mean daily temperature (°C)
<i>TMIN</i>	Daily minimum temperature (°C)
<i>TMAX</i>	Daily maximum temperature (°C)
<i>RHUM</i>	Mean daily relative humidity (%)
<i>RAIN</i>	Mean daily rainfall (mm)
<i>LAT</i>	Latitude of area of interest
<i>LON</i>	Longitude of area of interest.

Value

An `epicrop.wth` object, a data frame compatible with a `data.table::data.table()`, suitable for use in `epicrop`.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

Examples

```
# Get weather data from the POWER API, using the `get_wth()` function,
# this is for example purposes only as it is NOT necessary to reformat
# the data provided by `get_wth()`.
```

```
library(nasapower)
w <- get_power(
  lonlat = c(121.25562, 14.6674),
  dates = c("2000-06-30", "2000-12-31"),
  community = "AG",
  pars = c(
    "T2M",
    "T2M_MIN",
    "T2M_MAX",
    "RH2M",
    "PRECTOTCORR"
  ),
  temporal_api = "DAILY"
)

format_wth(
  wth = w,
  date = "YYYYMMDD",
  temp = "T2M",
  rhum = "RH2M",
  rain = "PRECTOTCORR",
  tmin = "T2M_MIN",
  tmax = "T2M_MAX",
  lat = "LAT",
  lon = "LON"
)
```

get_audpc

Get AUDPC from epicrop.sim Object

Description

Get the AUDPC value from an **epicrop** model run object's attributes for use in analysis, mapping, or reporting.

Usage

```
get_audpc(x)
```

Arguments

x An `epicrop.sim` object

Value

Numeric AUDPC value

 get_wth

 Get Weather Data for Modelling Disease

Description

This function is a wrapper for the `nasapower::get_power()` with predefined parameters suitable for use in **epicrop**.

Usage

```
get_wth(lonlat, dates, duration = NULL)
```

Arguments

lonlat	A numeric vector of geographic coordinates for a cell or region entered as lon (x), lat (y) coordinates.
dates	A character vector of start and end dates in that order.
duration	A numeric value indicating how many days a single growing season being simulated is to be. If this is used, only the start date will be used and the end date will be ignored if supplied. This must match the duration parameter value passed along to <code>seir()</code> or any of the predict family of functions.

Value

A `epicrop.wth`, a data frame based on a `data.table::data.table()`, object suitable for use in **epicrop** containing weather data, dates and geolocation information (LAT/LON values) with the following columns:

Field Name	Value
<i>YYYYMMDD</i>	Date as Year Month Day (ISO8601)
<i>DOY</i>	Consecutive day of year, commonly called "Julian date"
<i>TEMP</i>	Mean daily temperature (°C)
<i>TMIN</i>	Daily minimum temperature (°C)
<i>TMAX</i>	Daily maximum temperature (°C)
<i>RHUM</i>	Mean daily relative humidity (%)
<i>RAIN</i>	Mean daily rainfall (mm)
<i>LAT</i>	Latitude of area of interest
<i>LON</i>	Longitude of area of interest.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

Examples

```
# get weather for IRRI Zeigler Experiment Station in wet season 2000 from the
# default NASA POWER data.
power <- get_wth(
  lonlat = c(121.25562, 14.6774),
  dates = c("2000-06-30", "2000-12-31")
)

# get 120 days of weather for IRRI Zeigler Experiment Station in wet season
# 2000 by specifying the duration but not the end-date and specifying to use
# POWER data.
power <- get_wth(
  lonlat = c(121.25562, 14.6774),
  dates = "2000-06-30",
  duration = 120
)
```

run_epicrop_model	<i>Run an epicrop disease model across emergence dates and/or locations</i>
-------------------	---

Description

A unified helper that executes any **epicrop** model (e.g., [bacterial_blight\(\)](#), [brown_spot\(\)](#), [leaf_blast\(\)](#), [sheath_blight\(\)](#), [tungro\(\)](#), [modified_kim_leaf_blast\(\)](#), [modified_kim_sheath_blight\(\)](#), [leaf_rust\(\)](#), or [s_tritici_blotch\(\)](#)) over a set of emergence dates, using a list of per-year weather data.tables. Handles within-year and cross-year windows without globally binding weather.

Usage

```
run_epicrop_model(
  model_fun,
  emergence_dates,
  wth_list,
  window_days = 120L,
  output = "audpc",
  ...
)
```

Arguments

model_fun	Function. An epicrop model function (e.g., bacterial_blight()).
emergence_dates	Vector of dates (character "YYYY-MM-DD", Date, or IDate).

wth_list	Named list of weather <code>data.table::data.tables</code> with one per year; names(wth_list) must be year strings (e.g., "2001"). Each table must have a YYYYMMDD column (Date/IDate/ISO/"YYYYMMDD").
window_days	Integer. Days after emergence to include (default 120L). The slicer returns exactly window_days rows when data are available.
output	Character. "audpc" (compact summary using <code>get_audpc()</code>) or "full" (return full model outputs per date, with emergence added). Defaults to "audpc".
...	Additional arguments forwarded to <code>model_fun</code> .

Value

A `data.table::data.table`. When `output = "audpc"`: columns emergence (IDate) and AUDPC (numeric). When `output = "full"`: full model output columns plus emergence, and an AUDPC column preserved per emergence date.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com>

See Also

`fetch_epicrop_weather_list`, `build_epicrop_emergence`

Examples

```
years <- 2001:2003
month_day <- c("-06-01", "-06-14", "-06-30")
lonlat <- c(121.255669, 14.16742)
duration <- 120L
emergence <- build_epicrop_emergence(years, month_day)

wth_list <- fetch_epicrop_weather_list(years,
                                       lonlat,
                                       start_date = month_day,
                                       duration = duration)
run_epicrop_model(bacterial_blight, emergence, wth_list)
```

seir

*Susceptible-Exposed-Infectious-Removed (SEIR) Model Framework
with Lesion Expansion*

Description

This function is originally used by specific disease models in ‘EPIRICE’ and ‘EPIWHEAT’ to model disease intensity of several rice diseases and two wheat diseases respectively. Given proper values it can be used with other pathosystems as well.

Usage

```

seir(
  wth,
  emergence,
  onset,
  duration,
  H0,
  I0,
  RcA,
  RcT,
  RcOpt,
  p,
  i,
  Sx,
  a = 1,
  rhlim = 90,
  rainlim = 5,
  RRS = NULL,
  RRG = NULL,
  RcW = NULL,
  RRLEX = NULL,
  simple_wetness = TRUE,
  weighted_wetness = FALSE,
  age_driver = c("day", "dvs"),
  thermal_time = NULL,
  infection_window = NULL
)

```

Arguments

`wth` a data.frame of class `epicrop.wth` from either `get_wth()` or `format_wth()` containing weather on a daily time-step with the following field names:

Field Name	Value
<i>YYYYMMDD</i>	Date as Year Month Day (ISO8601)
<i>DOY</i>	Consecutive day of year, commonly called "Julian date"
<i>TEMP</i>	Mean daily temperature (degrees C)
<i>RHUM</i>	Mean daily relative humidity (%)
<i>RAIN</i>	Mean daily rainfall (mm)
<i>TMIN</i>	Optional Minimum daily temperature (degrees C), see TMIN/TMAX

Details

TMAX | **Optional** Maximum daily temperature (degrees C), see TMIN/TMAX
 Details *LAT* | **Optional** latitude of weather observation, see LAT/LON Details
LON | **Optional** longitude of weather observation, see LAT/LON Details

`emergence` Expected date of plant emergence (or transplanting for rice) entered in YYYY-MM-DD format (character).

onset	expected number of days until the onset of disease after emergence date (day, integer). For this modern seir() variant onset is 1-indexed (1 = emergence day). When comparing with legacy SEIR(), translate as legacy_onset = onset - 1.
duration	simulation duration <i>i. e.</i> , growing season length (day, integer). Described in Table 1 of Savary <i>et al.</i> 2012.
H0	initial number of plant's healthy sites (integer). Described in Table 1 of Savary <i>et al.</i> 2012.
I0	initial number of infective sites (scalar integer). Described in Table 1 of Savary <i>et al.</i> 2012. I0 is injected as new infections (become infectious after p days).
RcA	modifier for Rc (the basic infection rate corrected for removals) for crop age as a numeric two-column matrix with the first column being the crop age in days (0 to duration-1) or DVS (0 to 2) and the second being the modifier (bounded by 0 and 1). Described in Tables 1 and 2 of Savary <i>et al.</i> 2012.
RcT	modifier for Rc (the basic infection rate corrected for removals) for temperature as a numeric two-column matrix with the first column being temperature in degrees C and the second the modifier (bounded by 0 and 1). Described in Tables 1 and 2 of Savary <i>et al.</i> 2012.
RcOpt	potential basic infection rate corrected for removals (numeric). Described in Table 1 of Savary <i>et al.</i> 2012. This value can be modified to reflect crop cultivar resistance ratings as well, see Kim <i>et al.</i> 2015 for more details.
p	duration of latent period (day, integer or function of temperature). Described in Table 1 of Savary <i>et al.</i> 2012.
i	duration of infectious period (day, integer or function of temperature). Described in Table 1 of Savary <i>et al.</i> 2012.
Sx	maximum number of sites (integer). Described in Table 1 of Savary <i>et al.</i> 2012.
a	aggregation coefficient, values are from 1 to >1 (numeric or function of DVS). Described in Table 1 of Savary <i>et al.</i> 2012 and Table 1 of Savary <i>et al.</i> 2015. See further details in a - Aggregation section. Defaults to 1.
rhlim	relative humidity value threshold to decide whether leaves are wet or not (numeric). Described in Table 1 of Savary <i>et al.</i> 2012. Savary <i>et al.</i> 2012 used 90. Defaults to 90.
rainlim	rainfall amount (mm) threshold to decide whether leaves are wet or not (numeric). Described in Table 1 of Savary <i>et al.</i> 2012. Savary <i>et al.</i> 2012 used 5. Defaults to 5.
RRS	relative rate of physiological senescence (numeric, vector, or function of DVS). Described in Table 1 of Savary <i>et al.</i> 2012.
RRG	relative rate of growth (numeric, vector, or function of time). Described in Table 1 of Savary <i>et al.</i> 2012.
RcW	Optional modifier for Rc to calculate leaf wetness effect from 1 if wet to 0 if dry. This is a numeric two-column matrix with the first column being hours (0 to 24) and the second the modifier (bounded by 0 and 1). Used when simple_wetness = FALSE (hourly/derived wetness). If NULL (default), a linear relationship is used where 0 hours wet = 0 and 24 hours wet = 1. Ignored if simple_wetness = TRUE.

RRLEX	Rate of lesion expansion (scalar, vector, or function). Allows healthy sites to be directly converted to infectious sites without passing through latent stage. Set to 0 (default) for standard SEIR behavior.
simple_wetness	Logical. If TRUE (default), uses binary wetness calculation (0 or 1) based on rhlim and rainlim thresholds. If FALSE, uses hourly wetness duration calculation (requires TMIN, TMAX, LAT, DOY in wth) and applies RcW curve.
weighted_wetness	Logical. If TRUE, compute weighted hourly wetness (0 to 24); if FALSE, count hours where hourly RH \geq rhlim (0 to 24). Only used when simple_wetness = FALSE. Defaults to FALSE.
age_driver	Character. Use "day" (default, 0-indexed days from emergence) or "dvs" (development stage 0-2) for age-based modifiers in RcA. When "dvs", thermal_time parameter is required. Defaults to "day".
thermal_time	List with base_temp (degrees C) and gdd_to_maturity (degrees C days). Only required when age_driver = "dvs". Determines how growing degree days are accumulated and converted to development stage (DVS).
infection_window	Optional list defining DVS-based infection window: <ul style="list-style-type: none"> • dvs_start: DVS when susceptibility begins • dvs_end: DVS when susceptibility ends • base_temp: Base temperature for GDD (optional, uses thermal_time if present) • gdd_to_maturity: GDD to maturity (optional, uses thermal_time if present)

Value

A `data.table::data.table()` object containing the following columns:

simday Day of simulation (1-indexed, where 1 is the emergence date)

dates Date of simulation (Date class)

sites Total number of healthy sites present on day "x"

latent Number of latent sites present on day "x"

infectious Number of infectious sites present on day "x"

removed Number of removed sites present on day "x"

senesced Number of senesced sites present on day "x"

rateinf Rate of infection (new infections on day "x")

rlex Rate of lesion expansion (healthy sites converted to infectious via direct expansion on day "x")

rtransfer Rate of transfer from latent to infectious sites on day "x"

rremoved Rate of removal from infectious to removed sites on day "x"

rgrowth Rate of growth of healthy sites on day "x"

rsenesced Rate of senescence of healthy sites on day "x"

diseased Number of diseased (latent + infectious + removed) sites on day "x"

intensity Proportion of diseased sites (latent + infectious) per total living sites (excludes removed and senesced) on day "x"

lat Latitude value if provided by the wth object

lon Longitude value if provided by the wth object

The returned object also has an attribute AUDPC (accessed via `attr(result, "AUDPC")`) containing the area under the disease progress curve for the entire simulation, calculated using the trapezoidal rule.

Output Structure

The output is organized into three categories:

State Variables (stocks at end of each day):

- sites, latent, infectious, removed, senesced

Flow Variables (rates/changes during each day):

- rateinf, rlex, rtransfer, rremoved, rgrowth, rsenesced

Derived Variables (calculated from states):

- diseased, intensity

Mass Balance: At any day: sites + latent + infectious + removed + senesced should equal initial sites (H0) plus cumulative growth minus cumulative senescence (accounting for numerical precision).

a – Aggregation Details

When *a* is set to 1 the assumption is that that there is no disease aggregation with new infections occurring at random among the healthy sites. When *a* is greater than 1 there is aggregation in the disease occurrence, the pathogen is unable to access the entire population of healthy sites, which results in disease aggregation. Refer to Savary *et al.* (2012) for greater detail.

TMIN/TMAX Details

If `simple_wetness` is set to FALSE, the function will use the TMIN and TMAX columns in the wth object to calculate the leaf wetness value for 24 hours of the day using the `rhlim` and `rainlim` values and then uses `RcW` to calculate a value between 0 and 1 for the whole day. When `simple_wetness` is set to TRUE, the function only sets the leaf wetness to 0 or 1 for the day based on the `rhlim` and `rainlim` values.

LAT/LON Details

If the wth object provides LAT and LON columns, these will be included in the output for mapping purposes. Both values must be present. These columns are provided by default when using `get_wth()`.

Author(s)

Adam H. Sparks, <adamhsparks@gmail.com> from original by Robert J. Hijmans, Rene Pangga and Jorrel Aunario.

References

Sparks, A. H., P. D. Esker, M. Bates, W. Dall' Acqua, Z. Guo, V. Segovia, S.D. Silwal, S. Tolos, and K. A. Garrett, 2008. Ecology and Epidemiology in R: Disease Progress over Time. *The Plant Health Instructor*. doi:10.1094/PHIA2008012902.

Madden, L. V., G. Hughes, and F. van den Bosch. 2007. *The Study of Plant Disease Epidemics*. American Phytopathological Society, St. Paul, MN. doi:10.1094/9780890545058.

Savary, S., Nelson, A., Willocquet, L., Pangga, I., and Aunario, J. Modeling and mapping potential epidemics of rice diseases globally. *Crop Protection*, Volume 34, 2012, Pages 6-17, ISSN 0261-2194 doi:10.1016/j.cropro.2011.11.009.

See Also

Disease-specific helper functions:

- `bacterial_blight()`,
- `brown_spot()`,
- `leaf_blast()`,
- `sheath_blight()`,
- `tungro()`,
- `modified_kim_leaf_blast()`,
- `modified_kim_sheath_blight()`,
- `leaf_rust()`, and
- `s_tritici_blotch()`.

Index

audpc_dt, 3

bacterial_blight (disease_models), 4
bacterial_blight(), 13, 19
bb (disease_models), 4
brown_rust (disease_models), 4
brown_spot (disease_models), 4
brown_spot(), 13, 19
bs (disease_models), 4
build_epicrop_emergence, 3

cropgrowdays::daily_mean(), 9

data.frame(), 9
data.table::data.table, 14
data.table::data.table(), 3, 10, 12, 17
data.table::IDate, 4, 7
Date, 7
disease_models, 4

fetch_epicrop_weather_list, 7
fetch_epicrop_weather_list(), 4
format_wth, 9
format_wth(), 6, 15

get_audpc, 11
get_audpc(), 14
get_wth, 12
get_wth(), 6, 15, 18

lb (disease_models), 4
leaf_blast (disease_models), 4
leaf_blast(), 13, 19
leaf_rust (disease_models), 4
leaf_rust(), 13, 19
lr (disease_models), 4

mk_lb (disease_models), 4
mk_sb (disease_models), 4
modified_kim_leaf_blast
 (disease_models), 4
modified_kim_leaf_blast(), 13, 19
modified_kim_sheath_blight
 (disease_models), 4
modified_kim_sheath_blight(), 13, 19

nasapower::get_power(), 12

run_epicrop_model, 13
run_epicrop_model(), 3, 4

s_tritici_blotch (disease_models), 4
s_tritici_blotch(), 13, 19
sb (disease_models), 4
seir, 14
seir(), 3, 12
sheath_blight (disease_models), 4
sheath_blight(), 13, 19
stb (disease_models), 4

tg (disease_models), 4
tungro (disease_models), 4
tungro(), 13, 19