Package 'epifitter'

May 8, 2026

Type Package

Title Analysis and Simulation of Plant Disease Progress Curves

Version 1.0.0

Description Tools for analysis, visualization, and simulation of plant disease progress curves. Includes functions to calculate area-under-the-curve summaries, fit and compare exponential, monomolecular, logistic, and Gompertz models using linear or nonlinear regression, work with single or multiple epidemics, and produce 'ggplot2'-based visualizations. Also includes an experimental powdery mildew dataset for reproducible teaching and research workflows. See Madden, Hughes, and van den Bosch (2007) <doi:10.1094/9780890545058> for background on the epidemiological methods.

Depends R (>= 4.1)

Imports DescTools, cowplot, deSolve, dplyr, ggplot2, magrittr, minpack.lm, stats, tibble, tidyr

Suggests knitr, rmarkdown, lemon, testthat (>= 3.0.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Date 2026-04-11

URL https://github.com/AlvesKS/epifitter, https://alvesks.github.io/epifitter/

BugReports https://github.com/AlvesKS/epifitter/issues

X-schema.org-applicationCategory Tools

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author Kaique dos S. Alves [aut, cre] (ORCID:
<https://orcid.org/0000-0001-9187-0252>),
Emerson M. Del Ponte [aut] (ORCID:
<https://orcid.org/0000-0003-4398-409X>),
Adam H. Sparks [aut] (ORCID: <https://orcid.org/0000-0002-0061-8359>)

Maintainer Kaique dos S. Alves <kaiquedsalves@gmail.com>

Repository CRAN

Date/Publication 2026-04-12 18:00:02 UTC

Contents

AUDPC *Area under the disease progress curve*

Description

Calculate the area under a disease progress curve using the trapezoidal method.

Usage

```
AUDPC(
  time,
  y,
  y_proportion = TRUE,
  type = "absolute",
  aggregate = c("mean", "median", "none")
)
```

Arguments

time	A numeric vector of assessment times.
y	A numeric vector of disease intensity values.
y_proportion	Logical. Are the ‘y‘ values expressed as proportions?
type	Either ‘"absolute"‘ or ‘"relative"‘.
aggregate	How to handle multiple observations at the same time point. The default, ‘"mean"‘, averages replicated observations before calculating area. ‘"median"‘ uses the median and ‘"none"‘ requires unique time values.

Value

A numeric scalar with the AUDPC value.

References

Madden, L. V., Hughes, G., and van den Bosch, F. (2007). *The Study of Plant Disease Epidemics*. American Phytopathological Society.

Examples

```
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.5, n = 1)
AUDPC(time = epi$time, y = epi$y, y_proportion = TRUE)
```

AUDPC_2_points *Estimate AUDPC from two observations*

Description

Estimate the area under the disease progress curve from only the initial and final observations under a logistic epidemic assumption.

Usage

```
AUDPC_2_points(time, y0, yT)
```

Arguments

time	Time elapsed between the two assessments.
y0	Initial disease intensity as a proportion.
yT	Final disease intensity as a proportion.

Value

A numeric scalar with the estimated AUDPC.

References

Jeger, M. J., and Viljanen-Rollinson, S. L. H. (2001). The use of the area under the disease-progress curve (AUDPC) to assess quantitative disease resistance in crop cultivars. *Theoretical and Applied Genetics*, 102, 32-40.

Examples

```
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.5, n = 1)
AUDPC_2_points(time = epi$time[7], y0 = epi$y[1], yT = epi$y[7])
```

AUDPS *Area under the disease progress stairs*

Description

Calculate the area under the disease progress stairs, an alternative to AUDPC that gives more balanced weight to the first and last observations.

Usage

```
AUDPS(
  time,
  y,
  y_proportion = TRUE,
  type = "absolute",
  aggregate = c("mean", "median", "none")
)
```

Arguments

time	A numeric vector of assessment times.
y	A numeric vector of disease intensity values.
y_proportion	Logical. Are the 'y' values expressed as proportions?
type	Either '"absolute"' or '"relative"'.
aggregate	How to handle multiple observations at the same time point. The default, '"mean"', averages replicated observations before calculating area. '"median"' uses the median and '"none"' requires unique time values.

Value

A numeric scalar with the AUDPS value.

References

Simko, I., and Piepho, H.-P. (2012). The area under the disease progress stairs: Calculation, advantage, and application. *Phytopathology*, 102, 381-389.

Examples

```
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.5, n = 1)
AUDPS(time = epi$time, y = epi$y, y_proportion = TRUE)
```

---

expo_fun *Exponential model differential equation*

Description

Internal helper used by the simulation functions to solve the exponential epidemic model with 'deSolve::ode()'.

Usage

```
expo_fun(t, y, par)
```

Arguments

t	Time.
y	State variable.
par	Model parameters.

Value

A list containing the rate of change.

fit_lin *Fit epidemic models using linearization*

Description

Fit exponential, monomolecular, logistic, and Gompertz models to disease progress data using linearized forms of each model.

Usage

```
fit_lin(time, y)
```

Arguments

time	Numeric vector of assessment times.
y	Numeric vector of disease intensity values.

Value

A list with fit statistics, parameter estimates, and prediction data.

Examples

```
set.seed(1)
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.2, n = 4)
fit_lin(time = epi$time, y = epi$random_y)
```

fit_multi *Fit models to multiple disease progress curves*

Description

Apply 'fit_lin()', 'fit_nlin()', or 'fit_nlin2()' to multiple disease progress curves stored in a data frame.

Usage

```
fit_multi(
  time_col,
  intensity_col,
  data,
  strata_cols = NULL,
  starting_par = list(y0 = 0.01, r = 0.03, K = 0.8),
  maxiter = 500,
  nlin = FALSE,
  estimate_K = FALSE
)
```

Arguments

time_col	Character name specifying the time column.
intensity_col	Character name specifying the disease intensity column.
data	A data frame containing the variables for model fitting.
strata_cols	Character vector specifying grouping columns. Use 'NULL' to fit all rows as a single epidemic. Defaults to 'NULL'.
starting_par	Named list of starting values for model parameters.
maxiter	Maximum number of iterations for nonlinear fitting. Must be a positive number.
nlin	Logical. Should nonlinear fitting be used?
estimate_K	Logical. Should the asymptote 'K' be estimated?

Value

A list with grouped parameter estimates and prediction data.

Examples

```
set.seed(1)
epi1 <- sim_gompertz(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.2, n = 2)
epi2 <- sim_gompertz(N = 30, y0 = 0.01, dt = 5, r = 0.2, alpha = 0.2, n = 2)
data <- dplyr::bind_rows(epi1, epi2, .id = "curve")
fit_multi(time_col = "time", intensity_col = "random_y", data = data, strata_cols = "curve")
fit_multi(time_col = "time", intensity_col = "random_y", data = data)
```

---

fit_nlin *Fit epidemic models using nonlinear regression*

Description

Fit exponential, monomolecular, logistic, and Gompertz models to disease progress data using nonlinear regression.

Usage

```
fit_nlin(time, y, starting_par = list(y0 = 0.01, r = 0.03), maxiter = 50)
```

Arguments

time	Numeric vector of assessment times.
y	Numeric vector of disease intensity values.
starting_par	Named list with starting values for 'y0' and 'r'. When omitted or partially specified, 'epifitter' supplies data-driven fallback values.
maxiter	Maximum number of iterations. Must be a positive number.

Value

A list with fit statistics, parameter estimates, and prediction data.

Examples

```
set.seed(1)
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.2, n = 4)
fit_nlin(time = epi$time, y = epi$random_y, starting_par = list(y0 = 0.01, r = 0.03))
```

`fit_nlin2` *Fit epidemic models and estimate the asymptote*

Description

Fit monomolecular, logistic, and Gompertz epidemic models using nonlinear regression while also estimating the maximum disease intensity parameter 'K'.

Usage

```
fit_nlin2(
  time,
  y,
  starting_par = list(y0 = 0.01, r = 0.03, K = 0.8),
  maxiter = 50
)
```

Arguments

`time`	Numeric vector of assessment times.
`y`	Numeric vector of disease intensity values.
`starting_par`	Named list with starting values for 'y0', 'r', and 'K'. When omitted or partially specified, 'epifitter' supplies data-driven fallback values.
`maxiter`	Maximum number of iterations. Must be a positive number.

Value

A list with fit statistics, parameter estimates, and prediction data.

Examples

```
set.seed(1)
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.5, n = 4)
fit_nlin2(
  time = epi$time,
  y = epi$random_y * 0.8,
  starting_par = list(y0 = 0.01, r = 0.1, K = 0.8),
```

```
  maxiter = 1024
)
```

`gompi_fun` *Gompertz model differential equation*

Description

Internal helper used by the simulation functions to solve the Gompertz epidemic model with ‘deSolve::ode()‘.

Usage

```
gompi_fun(t, y, par)
```

Arguments

`t`	Time.
`y`	State variable.
`par`	Model parameters.

Value

A list containing the rate of change.

`logi_fun` *Logistic model differential equation*

Description

Internal helper used by the simulation functions to solve the logistic epidemic model with ‘deSolve::ode()‘.

Usage

```
logi_fun(t, y, par)
```

Arguments

`t`	Time.
`y`	State variable.
`par`	Model parameters.

Value

A list containing the rate of change.

mono_fun *Monomolecular model differential equation*

Description

Internal helper used by the simulation functions to solve the monomolecular epidemic model with ‘deSolve::ode()‘.

Usage

```
mono_fun(t, y, par)
```

Arguments

t	Time.
y	State variable.
par	Model parameters.

Value

A list containing the rate of change.

plot_fit *Plot fitted epidemic models*

Description

Create a faceted ‘ggplot2‘ panel showing observed and fitted values for the selected epidemic models.

Usage

```
plot_fit(
  object,
  point_size = 1.2,
  line_size = 1,
  models = c("Exponential", "Monomolecular", "Logistic", "Gompertz")
)
```

Arguments

object	A fitted object returned by ‘fit_lin()‘, ‘fit_nlin()‘, or ‘fit_nlin2()‘.
point_size	Point size for observed values.
line_size	Line width for fitted curves.
models	Character vector with the models to display.

Value

A 'ggplot2' object.

Examples

```
epi <- sim_logistic(N = 30, y0 = 0.01, dt = 5, r = 0.3, alpha = 0.2, n = 4)
fit <- fit_lin(time = epi$time, y = epi$random_y)
plot_fit(fit)
```

PowderyMildew *Powdery mildew disease progress curves in organic tomato*

Description

Experimental disease progress curve data for powdery mildew under different irrigation systems and soil moisture levels in organic tomato.

Format

A data frame with 240 rows and 5 variables:

irrigation_type Irrigation system.

moisture Soil moisture level.

block Experimental block.

time Assessment time.

sev Disease severity as a proportion.

References

Lage, D. A. C., Marouelli, W. A., and Cafe-Filho, A. C. (2019). Management of powdery mildew and behaviour of late blight under different irrigation configurations in organic tomato. *Crop Protection*, 125, 104886.

Examples

```
data("PowderyMildew")
str(PowderyMildew)
```

print.fit_lin *Print fitted model summaries*

Description

Print method for objects returned by [fit_lin()] and compatible fitters.

Usage

```
## S3 method for class 'fit_lin'
print(x, ...)
```

Arguments

x	An object produced by [fit_lin()] or [fit_nlin()].
...	Further arguments passed to [print()].

print.fit_nlin2 *Print fitted model summaries with asymptote estimates*

Description

Print method for objects returned by [fit_nlin2()].

Usage

```
## S3 method for class 'fit_nlin2'
print(x, ...)
```

Arguments

x	An object produced by [fit_nlin2()].
...	Further arguments passed to [print()].

`sim_exponential` *Simulate an exponential disease progress curve*

Description

Simulate disease progress data under the exponential epidemic model, with optional replicated observations.

Usage

```
sim_exponential(N = 10, dt = 1, y0 = 0.01, r, n, alpha = 0.2)
```

Arguments

`N`	Total epidemic duration.
`dt`	Time interval between assessments.
`y0`	Initial disease intensity.
`r`	Apparent infection rate.
`n`	Number of replicated curves.
`alpha`	Noise level applied to replicated observations.

Value

A data frame with simulated disease progress values and replicated noisy observations.

Examples

```
sim_exponential(N = 30, dt = 5, y0 = 0.01, r = 0.05, n = 4)
```

`sim_gompertz` *Simulate a Gompertz disease progress curve*

Description

Simulate disease progress data under the Gompertz epidemic model, with optional replicated observations.

Usage

```
sim_gompertz(N = 10, dt = 1, y0 = 0.01, r, K = 1, n, alpha = 0.2)
```

Arguments

N	Total epidemic duration.
dt	Time interval between assessments.
y0	Initial disease intensity.
r	Apparent infection rate.
K	Maximum disease intensity.
n	Number of replicated curves.
alpha	Noise level applied to replicated observations.

Value

A data frame with simulated disease progress values and replicated noisy observations.

Examples

```
sim_gompertz(N = 30, dt = 5, y0 = 0.01, r = 0.05, K = 1, n = 4)
```

---

sim_logistic *Simulate a logistic disease progress curve*

Description

Simulate disease progress data under the logistic epidemic model, with optional replicated observations.

Usage

```
sim_logistic(N = 10, dt = 1, y0 = 0.01, r, K = 1, n, alpha = 0.2)
```

Arguments

N	Total epidemic duration.
dt	Time interval between assessments.
y0	Initial disease intensity.
r	Apparent infection rate.
K	Maximum disease intensity.
n	Number of replicated curves.
alpha	Noise level applied to replicated observations.

Value

A data frame with simulated disease progress values and replicated noisy observations.

Examples

```
sim_logistic(N = 30, dt = 5, y0 = 0.01, r = 0.05, K = 1, n = 4)
```

`sim_monomolecular` *Simulate a monomolecular disease progress curve*

Description

Simulate disease progress data under the monomolecular epidemic model, with optional replicated observations.

Usage

```
sim_monomolecular(N = 10, dt = 1, y0 = 0.01, r, K = 1, n, alpha = 0.2)
```

Arguments

`N`	Total epidemic duration.
`dt`	Time interval between assessments.
`y0`	Initial disease intensity.
`r`	Apparent infection rate.
`K`	Maximum disease intensity.
`n`	Number of replicated curves.
`alpha`	Noise level applied to replicated observations.

Value

A data frame with simulated disease progress values and replicated noisy observations.

Examples

```
sim_monomolecular(N = 30, dt = 5, y0 = 0.01, r = 0.05, K = 1, n = 4)
```

Index