

Package ‘ushr’

May 8, 2026

Type Package

Title Understanding Suppression of HIV

Version 0.2.3

Date 2020-04-21

Maintainer Sinead E. Morris <sinead.morris@columbia.edu>

Description Analyzes longitudinal data of HIV decline in patients on antiretroviral therapy using the canonical biphasic exponential decay model (pioneered, for example, by work in Perelson et al. (1997) <doi:10.1038/387188a0>; and Wu and Ding (1999) <doi:10.1111/j.0006-341X.1999.00410.x>). Model fitting and parameter estimation are performed, with additional options to calculate the time to viral suppression. Plotting and summary tools are also provided for fast assessment of model results.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends dplyr (>= 0.8.0.1), tidyr (>= 0.8.3), ggplot2 (>= 3.1.1), stats

Suggests GGally (>= 1.4.0), knitr (>= 1.22), rmarkdown (>= 1.12), testthat (>= 2.2.0)

VignetteBuilder knitr

URL <https://github.com/SineadMorris/ushr>

NeedsCompilation no

Author Sinead E. Morris [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-8626-1698>>),
Luise Dziobek-Garrett [ctb],
Andrew Yates [ctb]

Repository CRAN

Date/Publication 2020-04-21 18:20:03 UTC

Contents

actg315raw	3
add_noise	4
biphasic_root	4
filter_data	5
filter_dataTTS	6
fit_model	7
fit_model_triphasic	8
get_biphasic	9
get_CI	10
get_CItable	10
get_curve	11
get_error	12
get_error_triphasic	12
get_nonparametricTTS	13
get_optim_fit	14
get_optim_fit_triphasic	15
get_parametricTTS	15
get_params	16
get_plottheme	17
get_singlephase	17
get_transformed_params	18
get_triphasic	18
get_TTS	19
plot_data	20
plot_model	21
plot_pairs	22
plot_TTS	22
remove_vl0	23
simulate_data	23
simulate_time	24
simulate_time_fixed	25
simulate_vl	25
single_root	26
summarize_model	26
switch_params	27
switch_simulated_params	27
transformVL	28
triphasic_root	28
tri_switch_params	29
ushr	29
ushr_triphasic	31

Index

34

`actg315raw`*Data from ACTG315 trial of HIV viral load in adults undergoing ART*

Description

Data from the ACTG315 clinical trial of HIV-infected adults undergoing ART. Data are included for 46 individuals, with HIV viral load measurements observed on specific days up to 28 weeks after treatment initiation, and converted to log₁₀ RNA copies/ml. The RNA assay detection threshold was 100 copies/ml. Additional columns include patient identifiers and CD4 T cell counts.

Usage

```
data(actg315raw)
```

Format

A data frame with 361 rows and 5 columns:

Obs.No Row number

Patid Numerical patient identifier

Day Time of each observation, in days since treatment initiation

log10.RNA. HIV viral load measurements, in log₁₀ RNA copies/ml

CD4 CD4 T cell counts, in cells/mm³

Source

[Hulin Wu, Data Sets](#)

References

Lederman et al (1998) JID 178(1), 70–79; Connick et al (2000) JID 181(1), 358–363; Wu and Ding (1999) Biometrics 55(2), 410–418.

Examples

```
library(dplyr)
data(actg315raw)

actg315 <- actg315raw %>%
  mutate(vl = 10^log10.RNA.) %>%
  select(id = Patid, time = Day, vl)

print(head(actg315))

plot_data(actg315, detection_threshold = 100)
```

add_noise *Add noise to viral load observations*

Description

This function adds noise to vl measurements for each subject.

Usage

```
add_noise(vl, sd_noise)
```

Arguments

vl numeric vector of viral load measurements.
sd_noise numeric value indicating the standard deviation level to be used when adding noise to the simulated data (on the log10 scale).

biphasic_root *Biphasic root function*

Description

This function defines the root equation for the biphasic model, i.e. $V(t) - \text{suppression_threshold} = 0$.

Usage

```
biphasic_root(timevec, params, suppression_threshold)
```

Arguments

timevec numeric vector of the times, t, at which V(t) should be calculated
params named vector of all parameters needed to compute the biphasic model, V(t)
suppression_threshold suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.

filter_data	<i>Prepare input data</i>
-------------	---------------------------

Description

This function prepares the raw input data for model fitting.

Usage

```
filter_data(data, detection_threshold = 20, censortime = 365,
            censor_value = 10, decline_buffer = 500, initial_buffer = 3,
            n_min_single = 3, threshold_buffer = 10, nsuppression = 1)
```

Arguments

data	raw data set. Must be a data frame with the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector with the viral load measurements for each subject; 'time' - numeric vector of the times at which each measurement was taken.
detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Measurements below this value will be assumed to represent undetectable viral load levels. Default value is 20.
censortime	numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
censor_value	positive numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
decline_buffer	numeric value indicating the value assigned to measurements below the detection threshold. Must be less than or equal to the detection threshold.
initial_buffer	numeric (integer) value indicating the maximum number of initial observations from which the beginning of each trajectory will be chosen. Default value is 3.
n_min_single	numeric value indicating the minimum number of data points required to be included in the analysis. Defaults to 3. It is highly advised not to go below this threshold.
threshold_buffer	numerical value indicating the range above the detection threshold which represents potential skewing of model fits. Subjects with their last two data points within this range will have the last point removed. Default value is 10.
nsuppression	numerical value (1 or 2) indicating whether suppression is defined as having one observation below the detection threshold, or two sustained observations. Default value is 1.

Details

Steps include: 1. Setting values below the detection threshold to half the detection threshold (following standard practice). 2. Filtering out subjects who do not suppress viral load below the detection threshold by a certain time. 3. Filtering out subjects who do not have a decreasing sequence of viral load (within some buffer range). 4. Filtering out subjects who do not have enough data for model fitting. 5. Removing the last data point of subjects with the last two points very close to the detection threshold. This prevents skewing of the model fit. Further details can be found in the Vignette.

Value

data frame of individuals whose viral load trajectories meet the criteria for model fitting. Includes columns for 'id', 'vl', and 'time'.

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

filter_data(simulated_data)
```

<code>filter_dataTTS</code>	<i>Prepare input data for non-parametric TTS calculations.</i>
-----------------------------	--

Description

This function prepares the raw input data for TTS interpolation. Individuals whose data do not meet specific inclusion criteria are removed (see Vignette for more details).

Usage

```
filter_dataTTS(data, suppression_threshold = 20, uppertime = 365,
  censor_value = 10, decline_buffer = 500, initial_buffer = 3)
```

Arguments

<code>data</code>	raw data set. Must be a data frame with the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector stating the viral load measurements for each subject; 'time' - numeric vector stating the time at which each measurement was taken.
<code>suppression_threshold</code>	numeric value indicating the suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.

uppertime	the maximum time point to include in the analysis. Subjects who do not suppress viral load below the suppression threshold within this time will be discarded from model fitting. Units are assumed to be the same as the 'time' column. Default value is 365.
sensor_value	positive numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
decline_buffer	the maximum allowable deviation of values away from a strictly decreasing sequence in viral load. This allows for e.g. measurement noise and small fluctuations in viral load. Default value is 500.
initial_buffer	numeric (integer) value indicating the maximum number of initial observations from which the beginning of each trajectory will be chosen. Default value is 3.

Details

Steps include: 1. Setting values below the suppression threshold to half the suppression threshold (following standard practice). 2. Filtering out subjects who do not suppress viral load below the suppression threshold by a certain time. 3. Filtering out subjects who do not have a decreasing sequence of viral load (within some buffer range).

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

filter_dataTTS(data = simulated_data)
```

fit_model

Fit model and obtain parameter estimates

Description

This function fits either the biphasic or single phase model to the processed data and extracts the best-fit parameters.

Usage

```
fit_model(data, id_vector, param_names, initial_params, free_param_index,
          n_min_biphasic, model_list, whichcurve = get_biphasic,
          forward_param_transform_fn, inv_param_transform_fn, searchmethod)
```

Arguments

<code>data</code>	dataframe with columns for each subject's identifier ('id'), viral load measurements ('vl'), and timing of sampling ('time')
<code>id_vector</code>	vector of identifiers corresponding to the subjects to be fitted.
<code>param_names</code>	names of parameter vector.
<code>initial_params</code>	named vector of the initial parameter guess.
<code>free_param_index</code>	logical vector indicating whether the parameters A, delta, B, gamma are to be recovered. This should be <code>c(TRUE, TRUE, TRUE, TRUE)</code> for the biphasic model and <code>c(FALSE, FALSE, TRUE, TRUE)</code> for the single phase model.
<code>n_min_biphasic</code>	the minimum number of data points required to fit the biphasic model. Defaults to 6. It is highly advised not to go below this threshold.
<code>model_list</code>	character indicating which model is to be fit. Can be either 'four' for the biphasic model, or 'two' for the single phase model. Defaults to 'four'.
<code>whichcurve</code>	indicates which model prediction function to use. Should be <code>get_biphasic</code> for the biphasic model or <code>get_singlephase</code> for the singlephase model. Defaults to <code>get_biphasic</code> .
<code>forward_param_transform_fn</code>	list of transformation functions to be used when fitting the model in optim. Defaults to log transformations for all parameters (to allow unconstrained optimization).
<code>inv_param_transform_fn</code>	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.
<code>searchmethod</code>	optimization algorithm to be used in optim. Defaults to Nelder-Mead.

`fit_model_triphasic` *Fit model and obtain parameter estimates*

Description

This function fits the triphasic model to the processed data and extracts the best-fit parameters.

Usage

```
fit_model_triphasic(data, id_vector, param_names, initial_params,
  free_param_index, n_min_triphasic, forward_param_transform_fn,
  inv_param_transform_fn, searchmethod)
```

Arguments

data	dataframe with columns for each subject's identifier ('id'), viral load measurements ('vl'), and timing of sampling ('time')
id_vector	vector of identifiers corresponding to the subjects to be fitted.
param_names	names of parameter vector.
initial_params	named vector of the initial parameter guess.
free_param_index	logical vector indicating whether the parameters A, delta, A_b, delta_b, B, gamma are to be recovered. This should be c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE) for the triphasic model.
n_min_triphasic	the minimum number of data points required to fit the triphasic model.
forward_param_transform_fn	list of transformation functions to be used when fitting the model in optim. Defaults to log transformations for all parameters (to allow unconstrained optimization).
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.
searchmethod	optimization algorithm to be used in optim. Defaults to Nelder-Mead.

 get_biphasic

Compute the biphasic model curve

Description

This function calculates the biphasic model, $V(t)$, for a vector of input times, t

Usage

```
get_biphasic(params, timevec)
```

Arguments

params	named numeric vector of all parameters needed to compute the biphasic model, $V(t)$
timevec	numeric vector of the times, t , at which $V(t)$ should be calculated

Value

numeric vector of viral load predictions, $V(t)$, for each time point in 'timevec'

Examples

```
get_biphasic(params = c(A = 10000, delta = 0.68, B = 1000, gamma = 0.03),
             timevec = seq(1, 100, length.out = 100))
```

get_CI	<i>Calculate parameter confidence intervals</i>
--------	---

Description

This function calculates parameter 95

Usage

```
get_CI(fit)
```

Arguments

fit	the output of optim i.e. the fitted model for a particular subject
-----	--

get_CItable	<i>Make parameter summary table</i>
-------------	-------------------------------------

Description

This function collate confidence intervals and parameter estimates from all subjects (fitted with the same model) into a nice table.

Usage

```
get_CItable(CIlist, param_names, free_param_index, fitted)
```

Arguments

CIlist	a list of confidence intervals and parameter estimates obtained from fitting either the single or biphasic model to each eligible subject.
param_names	character vector of the parameter names. This should be c("A", "delta", "B", "gamma") for the biphasic model or c("B", "gamma") for the single phase model.
free_param_index	logical vector indicating whether the parameters A, delta, B, gamma are to be included. This should be c(TRUE, TRUE, TRUE, TRUE) for the biphasic model and c(FALSE, FALSE, TRUE, TRUE) for the single phase model.
fitted	data frame with an 'id' column of the unique identifiers for each subject represented in CIlist. Identifiers should be ordered according to their appearance in CIlist.

get_curve	<i>Compute the model for a given subject's data and best-fit parameters</i>
-----------	---

Description

This function calculates the biphasic or single phase model given a subject's data and best-fit parameters

Usage

```
get_curve(data, best_param, param_names, whichcurve = get_biphasic)
```

Arguments

data	data frame with columns for the subject's identifier ('id') and timing of sampling ('time')
best_param	named numeric vector of best fit parameters obtained from fitting the biphasic or single phase model to the subjects data
param_names	character vector containing the names of the parameters in 'best_param'
whichcurve	character indicating which model function should be used. Use 'get_biphasic' for the biphasic model, or 'get_singlephase' for the single phase model. Defaults to 'get_biphasic'.

Value

data frame with columns for the sampling times ('time'), fitted viral load predictions ('fit'), and the corresponding subject identifier ('id')

Examples

```
nobs <- 7
example_param <- c(A = 10000, delta = 0.03, B = 1000, gamma = 0.68)

vldata <- get_biphasic(params = example_param, timevec = seq(5, 100, length.out = nobs))

subjectdata <- data.frame(id = 123, time = seq(5, 100, length.out = nobs),
                          vl = 10^(log10(vldata) + rnorm(nobs, 0, 0.2)))

get_curve(data = subjectdata, best_param = example_param, param_names = names(example_param))
```

get_error *Evaluate error metric between data and model prediction*

Description

For a given parameter set, this function computes the predicted viral load curve and evaluates the error metric between the prediction and observed data (to be passed to optim).

Usage

```
get_error(params, param_names, free_param_index, data, model_list,
          inv_param_transform_fn)
```

Arguments

params	named vector of the parameters from which the model prediction should be generated.
param_names	names of parameter vector.
free_param_index	logical TRUE/FALSE vector indicating whether the parameters A, delta, B, gamma are to be recovered. This should be c(TRUE, TRUE, TRUE, TRUE) for the biphasic model and c(FALSE, FALSE, TRUE, TRUE) for the single phase model.
data	dataframe with columns for the subject's viral load measurements ('vl'), and timing of sampling ('time')
model_list	character indicating which model is begin fit. Can be either 'four' for the biphasic model, or 'two' for the single phase model.
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.

get_error_triphasic *Evaluate error metric between data and model prediction*

Description

For a given parameter set, this function computes the predicted viral load curve and evaluates the error metric between the prediction and observed data (to be passed to optim).

Usage

```
get_error_triphasic(params, param_names, free_param_index, data,
                    inv_param_transform_fn)
```

Arguments

params	named vector of the parameters from which the model prediction should be generated.
param_names	names of parameter vector.
free_param_index	logical TRUE/FALSE vector indicating whether the parameters A, delta, A_b, delta_b, B, gamma are to be recovered. This should be c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE) for the triphasic model.
data	dataframe with columns for the subject's viral load measurements ('vl'), and timing of sampling ('time').
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.

get_nonparametricTTS *Non-parametric TTS function*

Description

This function computes the non-parametric form of the time to suppression

Usage

```
get_nonparametricTTS(vl, suppression_threshold, time, npoints)
```

Arguments

vl	numeric vector of viral load measurements.
suppression_threshold	numeric value for the suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.
time	numeric vector indicating the time when vl measurements were taken.
npoints	numeric value indicating the number of interpolation points to be considered.

<code>get_optim_fit</code>	<i>Fit model to data using optim</i>
----------------------------	--------------------------------------

Description

This function uses `optim` to fit either the biphasic or single phase model to data from a given subject

Usage

```
get_optim_fit(initial_params, param_names, free_param_index, data,
             model_list = "four",
             forward_param_transform_fn = forward_param_transform_fn,
             inv_param_transform_fn = inv_param_transform_fn, searchmethod)
```

Arguments

<code>initial_params</code>	named vector of the initial parameter guess.
<code>param_names</code>	names of parameter vector.
<code>free_param_index</code>	logical vector indicating whether the parameters A, delta, B, gamma are to be recovered. This should be <code>c(TRUE, TRUE, TRUE, TRUE)</code> for the biphasic model and <code>c(FALSE, FALSE, TRUE, TRUE)</code> for the single phase model.
<code>data</code>	dataframe with columns for the subject's viral load measurements (<code>'vl'</code>), and timing of sampling (<code>'time'</code>)
<code>model_list</code>	character indicating which model is begin fit. Can be either <code>'four'</code> for the biphasic model, or <code>'two'</code> for the single phase model. Defaults to <code>'four'</code> .
<code>forward_param_transform_fn</code>	list of transformation functions to be used when fitting the model in <code>optim</code> . Defaults to log transformations for all parameters (to allow unconstrained optimization).
<code>inv_param_transform_fn</code>	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.
<code>searchmethod</code>	optimization algorithm to be used in <code>optim</code> . Defaults to Nelder-Mead.

```
get_optim_fit_triphasic
    Fit triphasic model to data using optim
```

Description

This function uses `optim` to fit the triphasic model to data from a given subject

Usage

```
get_optim_fit_triphasic(initial_params, param_names, free_param_index,
  data, forward_param_transform_fn = forward_param_transform_fn,
  inv_param_transform_fn = inv_param_transform_fn, searchmethod)
```

Arguments

`initial_params` named vector of the initial parameter guess.

`param_names` names of parameter vector.

`free_param_index` logical vector indicating whether the parameters A, delta, A_b, delta_b, B, gamma are to be recovered. This should be `c(TRUE, TRUE, TRUE, TRUE, TRUE, TRUE)` for the triphasic model.

`data` dataframe with columns for the subject's viral load measurements ('vl'), and timing of sampling ('time')

`forward_param_transform_fn` list of transformation functions to be used when fitting the model in `optim`. Defaults to log transformations for all parameters (to allow unconstrained optimization).

`inv_param_transform_fn` list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.

`searchmethod` optimization algorithm to be used in `optim`. Defaults to Nelder-Mead.

```
get_parametricTTS    Parametric TTS function
```

Description

This function computes the parametric form of the time to suppression

Usage

```
get_parametricTTS(params, rootfunction, suppression_threshold, uppertime)
```

Arguments

params	named vector of all parameters needed to compute the suppression model, $V(t)$
rootfunction	specifies which function should be used to calculate the root: biphasic or single phase.
suppression_threshold	suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.
uppertime	numeric value indicating the maximum time that will be considered. Default value is 365.

get_params	<i>Extract fitted parameters</i>
------------	----------------------------------

Description

This function extracts all untransformed parameters from the output of `optim` (i.e. the fitted model).

Usage

```
get_params(fit, initial_params, free_param_index, param_names,
           inv_param_transform_fn, index = NULL)
```

Arguments

fit	the output of <code>optim</code> i.e. the fitted model for a particular subject
initial_params	named vector of the initial parameter guess
free_param_index	logical TRUE/FALSE vector indicating whether the parameters A, delta, B, gamma are to be recovered. This should be <code>c(TRUE, TRUE, TRUE, TRUE)</code> for the biphasic model and <code>c(FALSE, FALSE, TRUE, TRUE)</code> for the single phase model.
param_names	character vector of the parameter names. This should be <code>c("A", "delta", "B", "gamma")</code> for the biphasic model or <code>c("B", "gamma")</code> for the single phase model.
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions.
index	indicator value used inside the master function to indicate the subject number.

get_plottheme	<i>Get plotting theme</i>
---------------	---------------------------

Description

This function sets the plotting theme for ggplot.

Usage

```
get_plottheme(textsize)
```

Arguments

textsize	numeric value for base text size. Default is 9.
----------	---

get_singlephase	<i>Compute the single phase model curve</i>
-----------------	---

Description

This function calculates the single phase model, $V(t)$, for vector of input times, t

Usage

```
get_singlephase(params, timevec)
```

Arguments

params	named numeric vector of all parameters needed to compute the single phase model, $V(t)$
timevec	numeric vector of the times, t , at which $V(t)$ should be calculated

Value

numeric vector of viral load predictions, $V(t)$, for each time point in 'timevec'

Examples

```
get_singlephase(params = c(B = 1000, gamma = 0.68), timevec = seq(1, 100, length.out = 100))
```

 get_transformed_params

Transform parameters

Description

This function transforms parameter estimates according to user defined functions

Usage

```
get_transformed_params(params, param_transform_fn)
```

Arguments

params	vector of parameters
param_transform_fn	vector of functions for parameter transformation

get_triphasic

Compute the triphasic model curve

Description

This function calculates the triphasic model, $V(t)$, for a vector of input times, t

Usage

```
get_triphasic(params, timevec)
```

Arguments

params	named numeric vector of all parameters needed to compute the triphasic model, $V(t)$
timevec	numeric vector of the times, t , at which $V(t)$ should be calculated

Value

numeric vector of viral load predictions, $V(t)$, for each time point in 'timevec'

Examples

```
get_triphasic(params = c(A = 10000, delta = 1, B = 1000, gamma = 0.1, C = 100, omega = 0.03),
              timevec = seq(1, 100, length.out = 100))
```

get_TTS	<i>Time to suppression (TTS) function</i>
---------	---

Description

This function calculates the time to suppress HIV below a specified threshold.

Usage

```
get_TTS(model_output = NULL, data = NULL, suppression_threshold = 20,
        uppertime = 365, censor_value = 10, decline_buffer = 500,
        initial_buffer = 3, parametric = TRUE, ARTstart = FALSE,
        npoints = 1000)
```

Arguments

model_output	output from fitting model. Only required if parametric = TRUE.
data	raw data set. Must be a data frame with the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector stating the viral load measurements for each subject; 'time' - numeric vector stating the time at which each measurement was taken. Only required if parametric = FALSE.
suppression_threshold	suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.
uppertime	the maximum time interval to search for the time to suppression. Default value is 365.
censor_value	positive numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
decline_buffer	the maximum allowable deviation of values away from a strictly decreasing sequence in viral load. This allows for e.g. measurement noise and small fluctuations in viral load. Default value is 500.
initial_buffer	numeric (integer) value indicating the maximum number of initial observations from which the beginning of each trajectory will be chosen. Default value is 3.
parametric	logical TRUE/FALSE indicating whether time to suppression should be calculated using the parametric (TRUE) or non-parametric (FALSE) method. If TRUE, a fitted model object is required. If FALSE, the raw data frame is required. Defaults to TRUE.
ARTstart	logical TRUE/FALSE indicating whether the time to suppression should be represented as time since ART initiation. Default = FALSE. If TRUE, ART initiation times must be included as a data column named 'ART'.
npoints	numeric value of the number of interpolation points to be considered. Default is 1000.

Details

Options include: parametric (i.e. using the fitted model) or non-parametric (i.e. interpolating the processed data).

Value

a data frame containing all individuals who fit the inclusion criteria, along with their TTS estimates, and a column indicating whether the parametric or nonparametric approach was used.

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

get_TTS(data = simulated_data, parametric = FALSE)
```

plot_data	<i>Plot data</i>
-----------	------------------

Description

This function plots raw, filtered, or simulated data.

Usage

```
plot_data(data, textsize = 9, pointsize = 1, linesize = 0.5,
  facet_col = NULL, detection_threshold = 20)
```

Arguments

data	data frame of raw, filtered, or simulated data. Must include the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector stating the viral load measurements for each subject; 'time' - numeric vector stating the time at which each measurement was taken.
textsize	numeric value for base text size in ggplot. Default is 9.
pointsize	numeric value for point size in ggplot. Default is 1.
linesize	numeric value for line width in ggplot. Default is 0.5.
facet_col	numeric value for number of columns to use when faceting subject panels. Defaults to NULL (i.e. ggplot default).
detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Default value is 20.

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

plot_data(simulated_data)
```

plot_model	<i>Plot model fits</i>
------------	------------------------

Description

This function plots the output from model fitting.

Usage

```
plot_model(model_output, type = "biphasic", detection_threshold = 20,
           textsize = 9, pointsize = 1, linesize = 0.5, facet_col = NULL)
```

Arguments

model_output	output from model fitting using ushr().
type	character string indicating whether the biphasic or single phase fits should be plotted. Must be either "biphasic", "single", or "triphasic". Defaults to "biphasic".
detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Default value is 20.
textsize	numeric value for base text size in ggplot. Default is 9.
pointsize	numeric value for point size in ggplot. Default is 1.
linesize	numeric value for line width in ggplot. Default is 0.5.
facet_col	numeric value for number of columns to use when faceting subject panels. Defaults to NULL (i.e. ggplot default).

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

model_output <- ushr(data = simulated_data)

plot_model(model_output, type = "biphasic")
```

`plot_pairs`*Plot pairwise parameter distributions*

Description

This function creates pairwise scatterplots of the estimates parameters. The default plotting method requires GGally; if this package is not available, base R is used instead.

Usage

```
plot_pairs(model_output, type = "biphasic", textsize = 9,  
           pointsize = 1, linesize = 0.5)
```

Arguments

<code>model_output</code>	output from model fitting using ushr().
<code>type</code>	character string indicating whether the biphasic or single phase fits should be plotted. Must be either "biphasic", "single", or "triphasic". Defaults to "biphasic".
<code>textsize</code>	numeric value for base text size. Default is 9.
<code>pointsize</code>	numeric value for point size. Default is 1.
<code>linesize</code>	numeric value for line width; only used for GGally plots. Default is 0.5.

Examples

```
set.seed(1234567)  
  
simulated_data <- simulate_data(nsubjects = 20)  
  
model_output <- ushr(data = simulated_data)  
  
plot_pairs(model_output)
```

`plot_TTS`*Plot time to suppression distribution*

Description

This function plots a histogram of the time to suppression estimates.

Usage

```
plot_TTS(TTS_output, textsize = 9, bins = 20)
```

Arguments

TTS_output	output from estimating time to suppression (TTS) values using get_TTS().
textsize	numeric value for base text size on ggplot. Default is 9.
bins	numeric value indicating the number of bins for the histogram. Default is 20.

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

TTSeestimates <- get_TTS(data = simulated_data, parametric = FALSE)

plot_TTS(TTSeestimates, bins = 5)
```

remove_vl0	<i>Prune viral load data</i>
------------	------------------------------

Description

This function removes the first viral load data point for specific subjects

Usage

```
remove_vl0(id, which_ids, subset)
```

Arguments

id	vector of subject ids
which_ids	vector of ids that should have the first point removed
subset	data frame to which the function should be applied

simulate_data	<i>Create data function</i>
---------------	-----------------------------

Description

This function simulates example data that can be used to explore model fitting and plotting within the package. Subjects are assumed to be observed at regular intervals until either the end of the study or they are lost to follow up.

Usage

```
simulate_data(nsubjects = 10, detection_threshold = 20,
             censortime = 365, max_datapoints = 24, min_datapoints = 6,
             sd_noise = 0.1, param_noise = c(1.5, 0.1, 1.5, 0.1),
             mean_params = c(A = 10000, delta = 0.3, B = 10000, gamma = 0.03))
```

Arguments

nsubjects	numeric value indicating the number of subjects you want to simulate data for. Default is 10.
detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Measurements below this value will be assumed to represent undetectable viral load levels. Default value is 20.
censortime	numeric value indicating the maximum time point to include in the analysis. Default value is 365.
max_datapoints	numeric value indicating the maximum number of data points collected from any subject. Defaults to 24.
min_datapoints	numeric value indicating the minimum number of data points collected from any subject. Defaults to 6.
sd_noise	numeric value indicating the standard deviation level to be used when adding noise to the simulated data (on the log10 scale). Default value is 0.1
param_noise	numeric vector indicating the standard deviation to be used when selecting parameter values (on the log scale). Order of entries should be: A, delta, B, gamma. Default value is c(1.5, 0.1, 1.5, 0.1).
mean_params	named numeric vector indicating the mean parameter values for the subject decay curves. Default is c(A = 10000, delta = 0.3, B = 10000, gamma = 0.03).

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)
```

simulate_time	<i>Simulate timepoints for subjects</i>
---------------	---

Description

This function chooses the correct function for sampling observation times.

Usage

```
simulate_time(npoints, censortime, id, index, max_datapoints)
```

Arguments

npoints	numeric value indicating the number of observations to be sampled.
censortime	numeric value indicating the maximum time point to include in the analysis.
id	subject id. Can be numeric or a character.
index	numeric identifier for each subject/model combination.
max_datapoints	numeric value indicating the maximum number of data points collected from any subject.

simulate_time_fixed *Simulate timepoints for subjects according to fixed design.*

Description

This function simulates observed timepoints for each subject according to a fixed sampling design.

Usage

```
simulate_time_fixed(npoints, censortime, id, index, max_datapoints)
```

Arguments

npoints	numeric value indicating the number of observations to be sampled.
censortime	numeric value indicating the maximum time point to include in the analysis.
id	subject id. Can be numeric or a character.
index	numeric identifier for each subject/model combination.
max_datapoints	numeric value indicating the maximum number of data points collected from any subject.

simulate_vl *Simulate vl for subjects*

Description

This function simulates observed vl for each subject.

Usage

```
simulate_vl(params, timevec, id)
```

Arguments

params	named numeric vector of parameter values to simulate the biphasic model.
timevec	numeric vector of observed timepoints.
id	subject id. Can be numeric or a character.

single_root	<i>Single phase root function</i>
-------------	-----------------------------------

Description

This function defines the root equation for the single phase model, i.e. $V(t) - \text{suppression_threshold} = 0$.

Usage

```
single_root(timevec, params, suppression_threshold)
```

Arguments

timevec	numeric vector of the times, t, at which V(t) should be calculated
params	named vector of all parameters needed to compute the single phase model, V(t)
suppression_threshold	suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.

summarize_model	<i>Summarize model output</i>
-----------------	-------------------------------

Description

This function summarizes the output of model fitting..

Usage

```
summarize_model(model_output, data, stats = FALSE)
```

Arguments

model_output	output from model fitting using ushr().
data	dataframe of original data used for model fitting. Must include named 'id' column with subject identifiers.
stats	logical TRUE/FALSE: should the median and sd lifespans also be returned? Default is FALSE.

Value

a list containing (i) a summary of which subjects were successfully fit using the biphasic or single phase models, with their corresponding infected cell lifespan estimates ('summary'); (ii) if stats = TRUE: summary statistics for the estimated parameters from the biphasic model ('biphasicstats'); and (iii) if stats = TRUE: summary statistics for the estimated parameters from the single phase model ('singlestats').

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

model_output <- ushr(data = simulated_data)

summarize_model(model_output, data = simulated_data)
```

switch_params	<i>Switch names of rate parameters</i>
---------------	--

Description

This function switches the names of delta and gamma estimates if $\gamma > \delta$.

Usage

```
switch_params(biphasicCI)
```

Arguments

biphasicCI	data frame of parameter estimates and confidence intervals for the biphasic model.
------------	--

switch_simulated_params	<i>Switch names of simulated rate parameters</i>
-------------------------	--

Description

This function switches the names of delta and gamma estimates if $\gamma > \delta$.

Usage

```
switch_simulated_params(params)
```

Arguments

params	matrix of parameter estimates
--------	-------------------------------

transformVL	<i>Transform viral load data</i>
-------------	----------------------------------

Description

This function takes the log10 transform of viral load data & checks for NAs

Usage

```
transformVL(VL)
```

Arguments

VL	vector of viral load data
----	---------------------------

triphasic_root	<i>Triphasic root function</i>
----------------	--------------------------------

Description

This function defines the root equation for the triphasic model, i.e. $V(t) - \text{suppression_threshold} = 0$.

Usage

```
triphasic_root(timevec, params, suppression_threshold)
```

Arguments

timevec	numeric vector of the times, t, at which $V(t)$ should be calculated
params	named vector of all parameters needed to compute the triphasic model, $V(t)$
suppression_threshold	suppression threshold: measurements below this value will be assumed to represent viral suppression. Typically this would be the detection threshold of the assay. Default value is 20.

tri_switch_params	<i>Switch names of rate parameters</i>
-------------------	--

Description

This function switches the names of delta and gamma estimates if $\gamma > \delta$.

Usage

```
tri_switch_params(triphasicCI)
```

Arguments

triphasicCI	data frame of parameter estimates and confidence intervals for the biphasic model.
-------------	--

ushr	<i>Master function</i>
------	------------------------

Description

This function performs the entire analysis, from data filtering to fitting the biphasic/single phase models. The biphasic/single phase models should be used when ART comprises of RTI/PIs.

Usage

```
ushr(data, filter = TRUE, detection_threshold = 20, censortime = 365,
      censor_value = 10, decline_buffer = 500, initial_buffer = 3,
      threshold_buffer = 10, VL_max_decline = 10000, CI_max_diff = 1000,
      n_min_single = 3, n_min_biphasic = 6, nsuppression = 1,
      forward_param_transform_fn = list(log, log, log, log),
      inv_param_transform_fn = list(exp, exp, exp, exp),
      initial_params = c(A = 10000, delta = 0.68, B = 1000, gamma = 0.03),
      searchmethod = "Nelder-Mead")
```

Arguments

data	raw data set. Must be a data frame with the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector stating the viral load measurements for each subject; 'time' - numeric vector stating the time at which each measurement was taken.
filter	Logical TRUE/FALSE indicating whether the data should be processed (highly recommended) prior to model fitting. Default is TRUE.

detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Measurements below this value will be assumed to represent undetectable viral levels. Default value is 20.
censortime	numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded from model fitting. Units are assumed to be same as the 'time' measurements. Default value is 365.
sensor_value	positive numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
decline_buffer	numeric value indicating the maximum allowable deviation of values away from a strictly decreasing sequence in viral load. This allows for e.g. measurement noise and small fluctuations in viral load. Default value is 500.
initial_buffer	integer value indicating the maximum number of initial observations from which the beginning of each trajectory will be chosen. Default value is 3.
threshold_buffer	numeric value indicating the range above the detection threshold which represents potential skewing of model fits. Subjects with their last two data points within this range will have the last point removed. Default value is 10.
VL_max_decline	numeric value indicating the maximum allowable difference between first and second viral load measurements. Default is 10,000.
CI_max_diff	numeric value indicating the maximum allowable relative difference between lower and upper 95% confidence intervals i.e. $(\text{upper CI} - \text{lower CI})/\text{lower CI}$. Default is 1000.
n_min_single	numeric value indicating the minimum number of data points required to be included in the analysis. Defaults to 3. It is highly advised not to go below this threshold.
n_min_biphasic	numeric value indicating the minimum number of data points required to fit the biphasic model. Defaults to 6. It is highly advised not to go below this threshold.
nsuppression	numerical value (1 or 2) indicating whether suppression is defined as having one observation below the detection threshold, or two sustained observations. Default value is 1.
forward_param_transform_fn	list of transformation functions to be used when fitting the model in optim. Defaults to log transformations for all parameters (to allow unconstrained optimization).
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.
initial_params	named numeric vector of initial parameter guesses. Defaults to $c(A = 10000, \text{delta} = 0.68, B = 1000, \text{gamma} = 0.03)$.
searchmethod	optimization algorithm to be passed to 'optim()'. Defaults to 'Nelder-Mead'.

Details

Steps include: 1. Processing the raw data. 2. Fitting the biphasic model to subjects with eligible data e.g. those with enough data points and reliable confidence interval estimates. 3. Fitting the single phase model to the remaining subjects.

Value

a list containing the filtered data ('data_filtered'); parameter estimates for the biphasic and single phase models ('biphasicCI' and 'singleCI'); and predictions from the biphasic and single phase models ('biphasic_fits' and 'single_fits').

Examples

```
set.seed(1234567)

simulated_data <- simulate_data(nsubjects = 20)

model_output <- ushr(data = simulated_data)
```

ushr_triphasic	<i>Master function for the triphasic model</i>
----------------	--

Description

This function performs the entire analysis, from data filtering to triphasic model fitting. The triphasic model should be used when ART includes an integrase inhibitor.

Usage

```
ushr_triphasic(data, filter = TRUE, detection_threshold = 20,
  censortime = 365, censor_value = 10, decline_buffer = 500,
  initial_buffer = 3, threshold_buffer = 10, VL_max_decline = 10000,
  CI_max_diff = 1000, n_min_triphasic = 9, nsuppression = 1,
  forward_param_transform_fn = list(log, log, log, log, log, log),
  inv_param_transform_fn = list(exp, exp, exp, exp, exp, exp),
  initial_params = c(A = 10000, delta = 1, A_b = 1000, delta_b = 0.15, B
  = 10, gamma = 0.05), searchmethod = "Nelder-Mead")
```

Arguments

data	raw data set. Must be a data frame with the following columns: 'id' - stating the unique identifier for each subject; 'vl' - numeric vector stating the viral load measurements for each subject; 'time' - numeric vector stating the time at which each measurement was taken.
filter	Logical TRUE/FALSE indicating whether the data should be processed (highly recommended) prior to model fitting. Default is TRUE.

detection_threshold	numeric value indicating the detection threshold of the assay used to measure viral load. Measurements below this value will be assumed to represent undetectable viral levels. Default value is 20.
censortime	numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded from model fitting. Units are assumed to be same as the 'time' measurements. Default value is 365.
censor_value	positive numeric value indicating the maximum time point to include in the analysis. Subjects who do not suppress viral load below the detection threshold within this time will be discarded. Units are assumed to be the same as the 'time' column. Default value is 365.
decline_buffer	numeric value indicating the maximum allowable deviation of values away from a strictly decreasing sequence in viral load. This allows for e.g. measurement noise and small fluctuations in viral load. Default value is 500.
initial_buffer	integer value indicating the maximum number of initial observations from which the beginning of each trajectory will be chosen. Default value is 3.
threshold_buffer	numeric value indicating the range above the detection threshold which represents potential skewing of model fits. Subjects with their last two data points within this range will have the last point removed. Default value is 10.
VL_max_decline	numeric value indicating the maximum allowable difference between first and second viral load measurements. Default is 10,000.
CI_max_diff	numeric value indicating the maximum allowable relative difference between lower and upper 95% confidence intervals i.e. $(\text{upper CI} - \text{lower CI})/\text{lower CI}$. Default is 1000.
n_min_triphasic	numeric value indicating the minimum number of data points required to be included in the analysis. Defaults to 9. It is highly advised not to go below this threshold.
nsuppression	numerical value (1 or 2) indicating whether suppression is defined as having one observation below the detection threshold, or two sustained observations. Default value is 1.
forward_param_transform_fn	list of transformation functions to be used when fitting the model in optim. Defaults to log transformations for all parameters (to allow unconstrained optimization).
inv_param_transform_fn	list of transformation functions to be used when back-transforming the transformed parameters. Should be the inverse of the forward transformation functions. Defaults to exponential.
initial_params	named numeric vector of initial parameter guesses. Defaults to $c(A = 10000, \text{delta} = 1, A_b = 1000, \text{delta}_b = 0.15, B = 10, \text{gamma} = 0.05)$.
searchmethod	optimization algorithm to be passed to 'optim()'. Defaults to 'Nelder-Mead'.

Details

Steps include: 1. Processing the raw data. 2. Fitting the triphasic model to subjects with eligible data e.g. those with enough data points and reliable confidence interval estimates.

Value

a list containing the filtered data ('data_filtered'); parameter estimates for the triphasic model ('triphasicCI'); and predictions from the triphasic model ('triphasic_fits').

Index

* datasets

- actg315raw, 3
- actg315raw, 3
- add_noise, 4
- biphasic_root, 4
- filter_data, 5
- filter_dataTTS, 6
- fit_model, 7
- fit_model_triphasic, 8
- get_biphasic, 9
- get_CI, 10
- get_Citable, 10
- get_curve, 11
- get_error, 12
- get_error_triphasic, 12
- get_nonparametricTTS, 13
- get_optim_fit, 14
- get_optim_fit_triphasic, 15
- get_parametricTTS, 15
- get_params, 16
- get_plottheme, 17
- get_singlephase, 17
- get_transformed_params, 18
- get_triphasic, 18
- get_TTS, 19
- plot_data, 20
- plot_model, 21
- plot_pairs, 22
- plot_TTS, 22
- remove_vl0, 23
- simulate_data, 23
- simulate_time, 24
- simulate_time_fixed, 25
- simulate_vl, 25
- single_root, 26
- summarize_model, 26
- switch_params, 27
- switch_simulated_params, 27
- transformVL, 28
- tri_switch_params, 29
- triphasic_root, 28
- ushr, 29
- ushr_triphasic, 31