Package 'admtools'

October 4, 2024

Title Estimate and Manipulate Age-Depth Models

Version 0.4.0

Description Estimate age-depth models from stratigraphic and sedimentological data, and transform data
between the time and stratigraphic domain.

URL https://github.com/MindTheGap-ERC/admtools,
https://mindthegap-erc.github.io/admtools/

BugReports https://github.com/MindTheGap-ERC/admtools/issues

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10),

Imports ape

LazyData true

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

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Niklas Hohmann [aut, cre] (<https://orcid.org/0000-0003-1559-1838>)

Maintainer Niklas Hohmann <N.H.Hohmann@uu.nl>

Repository CRAN

Date/Publication 2024-10-03 23:01:00 UTC

Contents

`add_adm_to_multiadm` *add adm object ot multiadm object*

Description

add adm object ot multiadm object

Usage

```
add_adm_to_multiadm(x, ...)
```

Arguments

x	multiadm object
...	adm objects to be added to x

Value

a multiadm object

---

anchor *anchor age-depth model*

Description

anchors a deterministic age-depth model (adm object) at a tie point that is associated with uncertainty.

Usage

```
anchor(x, index = "last", t_anchor = NULL, n = 1000L)
```

Arguments

x	age-depth model
index	"last" or "first", or an integer (marked by L, e.g. 2L), specifying at which tie point the age-depth model will be anchored. If i is passed as integer, the i-th tie point is anchored.
t_anchor	time at which the adm is anchored. must be a function that takes no arguments and returns the timing of the tie point. see example or vignettes for details
n	integer, number of samples drawn from the tie point

Value

a collection of age-depth models (a multiadm object)

Examples

```
t_anchor = function() rnorm(1) # normally distributed uncertainty
x = tp_to_adm(t = c(1,2, 3), h = c(2,3, 4)) # simple age-depth model
m = anchor(x, index = "last", t_anchor = t_anchor, n = 100) # anchor age-depth model
plot(m)
m = anchor(x, index = 2L, t_anchor = t_anchor, n = 100)
plot(m)
```

`CarboCATLite_data` *Example data from CarboCATLite*

Description

Data exported from CarboCATLite model run, equivalent to scenario A from Hohmann et al. (2024). See therein for details.

Usage

```
CarboCATLite_data
```

Format

A list with the following fields:

- *time_myr* : time points in Myr from the model run
- *height_2_km_offshore_m* : sediment thickness accumulated 2 km from shore
- *height_4_km_offshore_m* : sediment thickness accumulated 4 km from shore
- *height_6_km_offshore_m* : sediment thickness accumulated 6 km from shore
- *height_8_km_offshore_m* : sediment thickness accumulated 8 km from shore
- *height_10_km_offshore_m* : sediment thickness accumulated 10 km from shore
- *height_12_km_offshore_m* : sediment thickness accumulated 12 km from shore
- *eustatic_SL_m* : eustatic sea level used for the model run.

Source

Elapsed model time, sea level, and accumulated sediment thickness taken from the scenario A model run here

References

- Burgess, Peter. "CarboCAT: A cellular automata model of heterogeneous carbonate strata." Computers & geosciences 53 (2013): 129-140. doi:10.1016/j.cageo.2011.08.026
- Burgess, Peter. (2023). CarboCATLite (v1.0.1). Zenodo. doi:10.5281/zenodo.8402578
- Hohmann, Niklas; Koelewijn, Joël R.; Burgess, Peter; Jarochowska, Emilia. 2024. "Identification of the mode of evolution in incomplete carbonate successions." BMC Ecology and Evolution 24, 113. doi:10.1186/s12862024022872.
- Hohmann, Niklas, Koelewijn, Joël R.; Burgess, Peter; Jarochowska, Emilia. 2023. "Identification of the Mode of Evolution in Incomplete Carbonate Successions - Supporting Data." Open Science Framework. doi:10.17605/OSF.IO/ZBPWA, published under the CC-BY 4.0 license.

condensation *condensation with height*

Description

returns (instantaneous) condensation (time preserved per length increment) for a section

Usage

```
condensation(x, h, mode = "rcll", ...)
```

Arguments

x	adm or multiadm object
h	numeric vector, positions where condensation is determined
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

if x is an adm object, a numeric vector of condensations. if x is a multiadm object, a list of condensations

condensation_fun *condensation function*

Description

returns a function that determines instantaneous condensation (time preserved per strat. increment)

Usage

```
condensation_fun(x, mode = "rcll", ...)
```

Arguments

x	adm object
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

a function

flux_const *constant deterministic tracer flux*

Description

For usage with *strat_cont_to_multiadm*; defines constant tracer flux in the time domain

Usage

```
flux_const()
```

Value

a function factory that takes no arguments

See Also

`flux_linear()`, `flux_quad()`, `strat_cont_gen_from_tracer()`

Examples

```
## Not run:
# see this vignette for an example
vignette("adm_from_trace_cont")

## End(Not run)
```

flux_linear *linear deterministic tracer flux*

Description

For usage with *strat_cont_to_multiadm* : defines linear tracer flux in the time domain Tracer flux is the linear function passing through the points (x0, y0) and (x1, y1)

Usage

```
flux_linear(x0 = 0, y0 = 1, x1 = 1, y1 = 2)
```

Arguments

x0	numeric, abscissa
y0	numeric, ordinate
x1	numeric, abscissa
y1	numeric, ordinate

Value

a function factory that takes no arguments. Upon each evaluation, it returns a linear function passing through the points (x0, y0) and (x1, y1)

See Also

`flux_const()`, `flux_quad()`, `strat_cont_gen_from_tracer()`

`flux_quad` *quadratic deterministic tracer flux*

Description

For usage with *strat_cont_to_multiadm* : defines quadratic tracer flux in the time domain defined by the function f(x) = ax^2 + bx + c

Usage

```
flux_quad(a = 1, b = 1, c = 1)
```

Arguments

a	numeric
b	numeric
c	numeric

Value

a function factory that takes no arguments. Upon each evaluation, it returns the quadratic function f(x) = ax^2 + bx + c

See Also

`flux_linear()`, `flux_const()`, `strat_cont_gen_from_tracer()`

get_completeness *Determine stratigraphic (in)completeness*

Description

Determine stratigraphic (in)completeness

Usage

```
get_completeness(x)
get_incompleteness(x)
```

Arguments

x	an adm object

Details

Stratigraphic (in)completeness is expressed as a proportion, i.e. a number between 0 and 1

Value

Number between 0 and 1, the stratigraphic (in)completeness

Examples

```
my_adm = tp_to_adm(t = 1:4, h = c(1,2,2,4))
get_completeness(my_adm)
get_incompleteness(my_adm)
```

get_data_from_eTimeOpt *extract data from eTimeOpt results*

Description

Extracts data from eTimeOpt. The type of data extracted depends on the `output` setting used for eTimeOpt. If you want ot extract specific data, adjust the output parameter in eTimeOpt to return the correct data (e.g. 2 for r^2 envelope). See eTimeOpt documentation for details on this. Then call this function on the return variable.

Usage

```
get_data_from_eTimeOpt(res, index = 1)
```

Arguments

res	results generated by eTimeOpt
index	which output should be extracted? See description for details

Value

a list with three entries "sed_rate" : numeric vector, sedimentation rates "height" : numeric vector, heights "results" : matrix with length(height) rows and length(sed_rate) columns. results of eTimeOpt

See Also

sed_rate_from_matrix() to use define sedimentation rates based on this functions outputs, sedrate_to_multiadm() to estimate age-depth models from the outputs.

---

get_height *determine stratigraphic height deposited at specific time*

Description

Takes an adm object and a vector of times, and returns the stratigraphic heights deposited at said times

Usage

```
get_height(x, t, destructive = TRUE, out_dom_val_h = "default", ...)
```

Arguments

x	an *adm* or *multiadm* object
t	vector of times
destructive	logical - should destructive intervals be considered? See Details
out_dom_val_h	"strat_limits", "default", or a vector with one or two entries. What value is assigned to times that are not covered by the age-depth model?
...	parameters handed over to is_destructive

Details

if destructive is true, NA is returned for times that coincide with destructive intervals. This is achieved by calling *is_destructive* with arguments passed by ....

out_dom_val specified the return value for times that are not covered by the age-depth model. For "default", NA is returned. For "strat_limits", the lowest resp. highest stratigraphic position is returned. For a vector of length one, this value is assigned to both sides. For a vector or length 2 or more, the first and second entries are assigned on the left (resp. right) side

Value

a vector with same length as t, containing the strat heights deposited

---

`get_hiat_duration` *extract hiatus duration*

Description

returns a vector of hiatus durations

Usage

```
get_hiat_duration(x)
```

Arguments

x	an adm object

Value

a vector with one element per hiatus: the duration of the hiatus

See Also

- `get_hiat_pos()` to determine only stratigraphic position of hiatuses
- `get_hiat_no()` to determine number of hiatuses in an adm
- `get_hiat_list()` to get hiatus position, stat & end time

`get_hiat_list` *extract hiatus info*

Description

returns a list with hiatus position and timing (start & end)

Usage

```
get_hiat_list(x)
```

Arguments

x	an adm object

Value

a list with one element per hiatus. each element is a named vector with the following entries:

- "height" : stratigraphic position of hiatus
- "start" : time when hiatus begins
- "end" : time when hiatus ends

See Also

- `get_hiat_pos()` to determine only stratigraphic position of hiatuses
- `get_hiat_no()` to determine number of hiatuses in an adm
- `get_hiat_duration()` to determine duration of hiatuses

`get_hiat_no` *get no. of hiatuses*

Description

Determines the number of hiatuses in an age-depth model

Usage

```
get_hiat_no(x)
```

Arguments

`x`	an adm object

Value

An integer, no. of hiatuses in the age-depth model

See Also

- `get_hiat_pos()` to determine stratigraphic positions of hiatuses
- `get_hiat_list()` to determine position and timing of hiatuses
- `get_hiat_duration()` to determine duration of hiatuses

Examples

```
my_adm = tp_to_adm(t = 1:4, h = c(1,2,2,3)) # one hiatus
get_hiat_no(my_adm)
```

get_hiat_pos *get hiatus positions*

Description

Determines stratigraphic position of hiatuses

Usage

```
get_hiat_pos(x)
```

Arguments

x	an adm object

Value

numeric vector with stratigraphic positions of hiatuses

See Also

- get_hiat_list() to get hiatus positions and durations
- get_hiat_no() to determine number of hiatuses
- get_hiat_duration() to determine duration of hiatuses

Examples

```
my_adm = tp_to_adm(t = 1:4, h = c(1,2,2,3)) # one hiatus at height 2
get_hiat_pos(my_adm)
```

get_L_tp *get height/length tie point*

Description

extracts the height/length time points from an age-depth model or sediment accumulation curve

Usage

```
get_L_tp(x, ...)
```

Arguments

x	age-depth model (adm/multiadm) or sediment accumulation curve (sac)
...	other options, currently not used

Value

numeric vector of the time/length tie points

See Also

`get_T_tp()` to extract time tie points

---

get_L_unit *extract length unit*

Description

extracts the length unit from adm or multiadm object

Usage

```
get_L_unit(x, ...)
```

Arguments

`x`	adm or multiadm object
`...`	other parameters

Value

character - the length unit of `x`

See Also

`get_T_unit()` `set_L_unit()`

---

get_time *Determine times based on age-depth model*

Description

Takes an age-depth model and vector of stratigraphic positions to determine the corresponding time of formation

Usage

```
get_time(x, h, hiat_mode = "start",
bdry_pts_hiat = "destructive", out_dom_val_t = "default")
```

Arguments

x	an *adm* or *multiadm* object
h	vector of stratigraphic positions
hiat_mode	"start", "end", or "destroy". If a stratigraphic position coincides with a hiatus, what should be returned?
bdry_pts_hiat	"consistent" or "destructive". How are hiatuses at the start/end of the adm treated?
out_dom_val_t	:"default", "time_limits", or a numeric value. What value is returned for heights not covered by the age-depth model?

Details

If a stratigraphic position coincides with a hiatus, should the start time or the end time of the hiatus be returned? Using "destroy" returns NA If the adm starts/ends with a hiatus, should the time returned be consistent with *hiat_mode*, or should it be NA?

Value

numeric vector. Times of deposition of the provided heights in h

---

get_total_duration *Total duration covered*

Description

Total duration covered

Usage

```
get_total_duration(x, ...)
```

Arguments

x	age-depth model (adm/multiadm) or sediment accumulation curve (sac)
...	other options, currently unused

Value

numeric, total duration covered by the age-depth models/sediment accumulation curve

See Also

`min_time()` and `max_time()` to extract the first/last tie point in time

get_total_thickness *get total thickness*

Description

for sediment accumulation curves, returns the difference between the highest and lowest point of the curve. For age-depth models, returns the total thickness of sediment accumulated.

Usage

```
get_total_thickness(x, ...)
```

Arguments

x	an age-depth model (adm/multiadm) or a sediment accumulation curve (sac)
...	other options, currently unused

Value

numeric, total sediment thickness accumulated

See Also

`max_height()` and `min_height()` to extract the highest/lowest stratigraphic point

get_T_tp *extract time tie points*

Description

Extracts the time tie points from an age-depth model or sediment accumulation curve

Usage

```
get_T_tp(x, ...)
```

Arguments

x	age-depth model (adm/multiadm) or sediment accumulation curve (sac)
...	other options, currently unused

Value

a vector, containing the time tie points

See Also

`get_L_tp()` to extract length/height tie points

get_T_unit *extract Time unit*

Description

extracts the Time unit from adm or multiadm object

Usage

```
get_T_unit(x, ...)
```

Arguments

x	adm or multiadm object
...	other parameters

Value

character - the time unit of x

See Also

set_T_unit() get_L_unit()

is_adm *Is an adm object a valid age-depth model*

Description

Constructors for adm objects such as *tp_to_adm* do not check whether the inputs define a valid age-depth mode, e.g. one where the law of superposition holds. This function performs these checks

Usage

```
is_adm(x, quietly = TRUE)
```

Arguments

x	an object
quietly	logical. should descriptive warnings be shown?

Value

logical. Is the input a valid adm object?

Examples

```
x = tp_to_adm(t = c(2,1), h = c(1,2)) # reversed order of time tie points
is_adm(x) # returns FALSE
```

`is_destructive` *Is deposition destructive?*

Description

Determines whether specified time is destructive or not

Usage

```
is_destructive(x, t, mode = "rcll",
bdry_pts_hiat = "destructive", out_dom_mode = "default")
```

Arguments

`x`	an *adm* or *multiadm* object
`t`	vector of times
`mode`	string, either "rcll", "lcrl", "open", or "closed"
`bdry_pts_hiat`	string, "destructive" or "consistent". If the adm starts/ends with a hiatus, should the start/end be removed, or treated consistently with mode?
`out_dom_mode`	""default", "destructive", or "conservative"

Value

logical vector of same length as t. Is deposition at time t destructive?

`is_multiadm` *is valid multiadm object?*

Description

is valid multiadm object?

Usage

```
is_multiadm(x, quietly = TRUE)
```

Arguments

x	object to be tested
quietly	logical, should a descriptive warning be returned?

Value

Logical. Is the object a valid multiadm object?

---

is_sac *is valid sac objects*

Description

checks if the object is a valid sac object

Usage

```
is_sac(x)
```

Arguments

x	the object to check

Value

logical. Is x a valid sac object?

L_axis_lab *plot height axis label*

Description

plot height axis label

Usage

```
L_axis_lab(
  label = "Height",
  unit = TRUE,
  sep = " ",
  brac = c("[", "]"),
  line = 2,
  outer = FALSE,
  at = NA,
  adj = NA,
```

```
  padj = NA,
  cex = NA,
  col = NA,
  font = NA,
  ...
)
```

Arguments

label	Axis label
unit	Logical or character, should unit be plotted
sep	separator between label and unit
brac	brackets surrounding unit
line	parameter passed to *mtext*, see ?mtext for details
outer	parameter passed to *mtext*, see ?mtext for details
at	parameter passed to *mtext*, see ?mtext for details
adj	parameter passed to *mtext*, see ?mtext for details
padj	parameter passed to *mtext*, see ?mtext for details
cex	parameter passed to *mtext*, see ?mtext for details
col	parameter passed to *mtext*, see ?mtext for details
font	parameter passed to *mtext*, see ?mtext for details
...	further graphical parameters passed to *mtext*, see ?mtext for details

Value

invisible NULL

See Also

`plot.adm()` for plotting of adms

---

make_legend *plot legend*

Description

plots a legend for the multiadm plot

Usage

```
make_legend()
```

Value

invisible NULL

max_height *get highest stratigraphic tie point*

Description

get highest stratigraphic tie point

Usage

```
max_height(x)
```

Arguments

x	age-depth model (adm) or sediment accumulation curve (sac)

Value

number, stratigraphic position of the highest stratigraphic tie point

See Also

`min_height()`, `get_total_thickness()`

max_time *last time tie point*

Description

last time tie point

Usage

```
max_time(x)
```

Arguments

x	age-depth model (adm) or sediment accumulation curve (sac)

Value

number, last time tie point of the age-depth model/sediment accumulation curve

See Also

`min_time()`, `get_total_duration()`

mean_adm *get mean ADM*

Description

returns the mean adm of a multiadm object

Usage

```
mean_adm(x, h)
```

Arguments

x	a multiadm object
h	the heights at which to evaluate the adm

Value

an adm object

See Also

median_adm() and quantile_adm() for median and quantile adms, respectively

median_adm *get median ADM*

Description

returns the median adm of a multiadm object

Usage

```
median_adm(x, h)
```

Arguments

x	a multiadm object
h	the heights at which to evaluate the adm

Value

an adm object

See Also

mean_adm() for the mean age-depth model, quantile_adm() for the more general implementation

merge_adm_to_multiadm *combine multiple adm ojects into multiadm object*

Description

combine multiple adm ojects into multiadm object

Usage

```
merge_adm_to_multiadm(...)
```

Arguments

...	adm objects

Value

object of class multiadm

merge_multiadm *merge multiple multiadm objects*

Description

merge multiple multiadm objects

Usage

```
merge_multiadm(...)
```

Arguments

...	adm objects

Value

multiadm object

min_height *get lowest stratigraphic tie point*

Description

get lowest stratigraphic tie point

Usage

```
min_height(x)
```

Arguments

x	an age-depth model (adm) or sediment accumulation curve (sac)

Value

number, stratigraphic position of lowest tie point

See Also

`get_total_thickness()`, `max_height()`

min_time *first time tie point*

Description

first time tie point

Usage

```
min_time(x)
```

Arguments

x	age-depth model (adm) or sediment accumulation curve (sac)

Value

number, timing of first tie point of the age-depth model/sediment accumulation curve

See Also

`max_time()`, `get_total_duration()`

plot.adm *plotting adm objects*

Description

plotting adm objects

Usage

```
## S3 method for class 'adm'
plot(
  x,
  lwd_destr = 1,
  lwd_acc = 1,
  lty_destr = 3,
  lty_acc = 1,
  col_destr = "black",
  col_acc = "black",
  ...
)
```

Arguments

x	an adm object
lwd_destr	line width of hiatuses
lwd_acc	line width of conservative intervals
lty_destr	linetype of hiatuses
lty_acc	line type of conservative intervals
col_destr	color of erosive intervals
col_acc	color of conservative intervals
...	arguments passed to plot

See Also

L_axis_lab() and T_axis_lab() for plotting time and axis labels, the vignette on plotting available via browseVignettes(package = "admtools")

plot.multiadm *plot multiadm object*

Description

plots the median age (red) and the 95 % envelope (blue) of a multiadm object

Usage

```
## S3 method for class 'multiadm'
plot(x, ...)
```

Arguments

x	multiadm object
...	parameters passed to plot

Value

a plot of the multiadm object

Examples

```
## Not run:
# see
vignette("adm_from_trace_cont")
# and
vignette("adm_from_sedrate")
# for example plots.

## End(Not run)
```

plot.sac *plot sediment accumulation curve*

Description

plot sediment accumulation curve

Usage

```
## S3 method for class 'sac'
plot(x, ...)
```

Arguments

x	object of class *sac*
...	further parameters (currently ignored)

Value

invisible NULL

---

plot.stratlist *plot strat list*

Description

plots a stratlist, i.e. a list of values associated with stratigraphic positions (typically returned by time_to_strat). will plot the element with matching ord_name against stratigraphic positions.

Usage

```
## S3 method for class 'stratlist'
plot(x, orientation = "du", ord_name = "y", ...)
```

Arguments

x	stratlist object
orientation	character, either "du" (down-up) or "lr" (left-right). Orientation of plotting
ord_name	name of the ordinate. Values plotted against time
...	further arguments passed to plot

---

plot.timelist *plot time lists*

Description

plot time lists

Usage

```
## S3 method for class 'timelist'
plot(x, ...)
```

Arguments

x	a time list
...	other options passed to plot

plot_condensation *plot condensation in height*

Description

plots condensation (time per stratigraphic increment) throughout the section

Usage

```
plot_condensation(x, h = "default", mode = "rcll", ...)
```

Arguments

x	an adm object
h	"default" or a numeric vector of height where the sed rate is evaluated
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

invisible null

plot_erosive_intervals *mark erosive time intervals*

Description

mark erosive time intervals

Usage

```
plot_erosive_intervals(
  density = NULL,
  angle = 45,
  col = "azure3",
  border = NA,
  lty = 1,
  lwd = 1
)
```

Arguments

density	parameter passed to *rect*, see ?rect for details
angle	parameter passed to *rect*, see ?rect for details
col	parameter passed to *rect*, see ?rect for details
border	parameter passed to *rect*, see ?rect for details
lty	parameter passed to *rect*, see ?rect for details
lwd	parameter passed to *rect*, see ?rect for details

Value

invisible NULL

---

plot_sed_rate_l *plot sed. rate in height*

Description

plot sed. rate in height

Usage

```
plot_sed_rate_l(x, h = "default", mode = "rcll", ...)
```

Arguments

x	an adm object
h	"default" or a numeric vector of height where the sed rate is evaluated
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

invisible null

plot_sed_rate_t *plot sedimentation rate in time*

Description

plot sedimentation rate in time

Usage

```
plot_sed_rate_t(x, mode = "rcll")
```

Arguments

x	adm object
mode	string, "rcll" or "lcrl". Should the sedimentation rate be Right Continuous with Left Limits (rcll) or Left Continuous with Right Limits (lcrl)

Value

invisible NULL

quantile_adm *get quantile ADM*

Description

returns the quantile adm of a multiadm object

Usage

```
quantile_adm(x, h, p)
```

Arguments

x	a multiadm object
h	the heights at which to evaluate the adm
p	percentile, as number between 0 and 1

Value

an adm object

See Also

`median_adm()` to extract the median adm, `mean_adm()` for the mean adm

sac_to_adm *turn sed. acc curve into adm*

Description

turn sed. acc curve into adm

Usage

```
sac_to_adm(x)
```

Arguments

x	object of class *sac*

Value

object of class *adm*

See Also

`tp_to_adm()` for the generator of *adm*

sedrate_to_multiadm *Estimate age-depth model from sedimentation rates & tie points*

Description

Combines information on tie points and sedimentation rates to estimate age-depth models and their associated uncertainty. For an example, see `vignette("adm_from_sedrate")`.

Usage

```
sedrate_to_multiadm(
  h_tp,
  t_tp,
  sed_rate_gen,
  h,
  no_of_rep = 100L,
  subdivisions = 100L,
  stop.on.error = TRUE,
  T_unit = NULL,
  L_unit = NULL
)
```

Arguments

h_tp	: function, returns stratigraphic positions of tie points
t_tp	: function, returns times of tie points
sed_rate_gen	: function, returns function describing sedimentation rate
h	: numeric, heights where the adm is calculated
no_of_rep	: integer, number of repetitions
subdivisions	maximum no of subintervals used in numeric integration. passed to *integrate*, see ?stats::integrate for details
stop.on.error	logical passed to *integrate*, see ?stats::integrate for details
T_unit	time unit
L_unit	length unit

Value

object of class multiadm

Examples

```
## Not run:
# see this vignette for an example
vignette("adm_from_sedrate")

## End(Not run)
```

---

sed_rate_from_matrix *make sed rate gen from matrix*

Description

Construct a sedimentation rate generator (function factory) from a matrix, e.g. one returned from `get_data_from_eTimeOpt`. This generator can be passed on to `sedrate_to_multiadm` to estimate age-depth models from it. If mode is "deterministic", the generator evaluates the sedimentation rates at heights specified by `height`, if the mode is "poisson" it is evaluated at heights that are determined based on a poisson point process. At these heights, the value of the sedimentation rate is determined based on the (pseudo) pdf that is determined by the matrix values.

Usage

```
sed_rate_from_matrix(
  height,
  sedrate,
  matrix,
  mode = "deterministic",
  rate = 1,
```

```
  expand_domain = TRUE,
  transform = identity
)
```

Arguments

height	vector of heights
sedrate	vector of sed. rates x values
matrix	matrix of sed rate y values. Must have as many columns as length(height) and as many rows as length(sedrate).
mode	character, "deterministic" or "poisson". Determines at which stratigraphic heights the sed rate is determined. If "deterministic" this will be the heights in height, if "poisson" the heights where the sed rate is determined follows a poisson point process with rate specified by rate
rate	numeric, rate of the Poisson point process determining frequency of sedimentation rate changes.
expand_domain	should sedimentation rates be defined below/above the highest/lowest height in the section? If TRUE, the sed rate values are the values at the closest interpolated point, if FALSE it will be NA
transform	a function, the identity function by default. How should the values of the (pseudo)pdf defined by the entries of matrix be transformed? Using this function allows to (nonlinearly) rescale the values in matrix to put more emphasis on higher/lower values

Value

a function factory for usage with sedrate_to_multiadm

See Also

sedrate_to_multiadm() for estimating sedimentation rates based on the outputs, get_data_from_eTimeOpt() for extracting data from the eTimeOpt function of the astrochron package.

---

sed_rate_gen_from_bounds

seg rate gen from upper/lower bounds

Description

seg rate gen from upper/lower bounds

Usage

```
sed_rate_gen_from_bounds(h_l, s_l, h_u, s_u, rate = 1)
```

Arguments

h_l	height values for lower bounds
s_l	sed rate values for lower bounds
h_u	height values for upper bounds
s_u	sed rate values for upper bounds
rate	rate of poisson point process

Value

a function factory for usage with sedrate_to_multiadm

See Also

sedrate_to_multiadm() for estimating age-depth models using the outputs, sed_rate_from_matrix() for other means of defining sedimentation rates

---

sed_rate_l *sedimentation rate in stratigraphic height*

Description

determines instantaneous sedimentation rate at a specified stratigraphic position

Usage

```
sed_rate_l(x, h, mode = "rcll", ...)
```

Arguments

x	adm object
h	numeric vector, stratigraphic positions
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

a vector of sed rates (if x is an adm object), or a list of sedimentation rates

sed_rate_l_fun *sed rate in height function*

Description

returns a function that determines sed. rates in height

Usage

```
sed_rate_l_fun(x, mode = "rcll", ...)
```

Arguments

x	an adm object
mode	string, handed over to *sed_rate_t*, see ?sed_rate_t for details
...	parameters passed to *get_time*, see ?get_time for details

Value

a function

sed_rate_t *sedimentation rate in time domain*

Description

infers the instantaneous sedimentation rate from adm objects

Usage

```
sed_rate_t(x, t, mode = "rcll")
```

Arguments

x	adm or multiadm object
t	vector of times at which sedimentation rates are determined
mode	string, "rcll" or "lcrl". at non-differential points, is the sed rate left or right continuous?

Value

for adm objects, a vector giving sed. accumulation rates at time t. For multiadm objects, a list with accumulation rates

sed_rate_t_fun *sedimentation rate function*

Description

returns a function that returns sedimentation rate

Usage

```
sed_rate_t_fun(x, mode = "rcll")
```

Arguments

x	an adm object
mode	string, "rcll" or "lcrl". Should the sedimentation rate be Right Continuous with Left Limits (rcll) or Left Continuous with Right Limits (lcrl)

Value

a function

set_L_unit *set length units*

Description

set length units for adm and multiadm objects

Usage

```
set_L_unit(x, L_unit, ...)
```

Arguments

x	adm or multiadm object
L_unit	time unit
...	further parameters

Value

an adm or multiadm object with the L unit assigned

See Also

set_T_unit() get_L_unit()

set_T_unit *set time units*

Description

set time units for adm and multiadm objects

Usage

```
set_T_unit(x, T_unit, ...)
```

Arguments

x	adm or multiadm object
T_unit	time unit
...	further parameters

Value

an adm or multiadm object with the time unit assigned

See Also

set_L_unit() get_T_unit()

split_multiadm *split multiadm objects into adm*

Description

split multiadm objects into adm

Usage

```
split_multiadm(x)
```

Arguments

x	a multiadm object

Value

list with objects of class adm

`strat_cont_gen_from_tracer` *proxy values in strat domain*

Description

Generates a function factory for usage with *strat_cont_to_multiadm* based on empirical tracer measurements in the section

Usage

```
strat_cont_gen_from_tracer(
  bin_borders,
  df,
  distribution = "normal",
  cap = TRUE,
  cap_val = 0
)
```

Arguments

`bin_borders`	borders of sampling bins
`df`	data frame with proxy records
`distribution`	character, currently only "normal" implemented. Specifies the distribution of proxies
`cap`	logical. Should values below `cap_val`be replaced?
`cap_val`	numeric. If `cap = TRUE`, values below `cap_val`will be replaced by `cap_val`

Value

a functional for usage with strat_cont_to_multiadm

See Also

`flux_const()`, `flux_linear()`, `flux_quad()` to define tracer fluxes

Examples

```
## Not run:
# see this vignette for a use case
vignette("adm_from_trace_cont")

## End(Not run)
```

`strat_cont_to_multiadm` *estimate age-depth model from tracer*

Description

Estimates age-depth models by comparing observed tracer values in a section with assumptions on tracer flux in time. See `vignette("adm_from_trace_cont")` for a full example.

Usage

```
strat_cont_to_multiadm(
  h_tp,
  t_tp,
  strat_cont_gen,
  time_cont_gen,
  h,
  no_of_rep = 100L,
  subdivisions = 100L,
  stop.on.error = TRUE,
  T_unit = NULL,
  L_unit = NULL
)
```

Arguments

`h_tp`	function, returning tie point heights
`t_tp`	function, returning tie points times
`strat_cont_gen`	function, describing tracer data observed in the section
`time_cont_gen`	function, describing tracer changes in time
`h`	numeric vector, heights where the age depth model is described
`no_of_rep`	integer, number of age depth models generated
`subdivisions`	integer, max no. of subintervals used by integration procedure. passed to *integrate*, see ?stats::integrate for details
`stop.on.error`	logical passed to *integrate*, see ?stats::integrate for details
`T_unit`	NULL or character, time unit
`L_unit`	NULL or character, length unit

Value

Object of class multiadm

Examples

```
## Not run:
# see this vignette for an example
vignette("adm_from_trace_cont")

## End(Not run)
```

`strat_to_time` *transform objects from strat. to time domain*

Description

Takes an object and transforms it from the time domain into the stratigraphic domain using the provided age-depth model. Currently implemented for the "phylo", "list", and "numeric" class. Wraps around `get_time`.

Usage

```
strat_to_time(obj, x, ...)
```

Arguments

`obj`	the object to be transformed
`x`	age-depth model
`...`	other parameters

Value

an object of the same type as obj

See Also

`time_to_strat()` to transform data from the time to the stratigraphic domain, `strat_to_time.phylo()`, `strat_to_time.numeric()` and `strat_to_time.list()` for details on how to transform phylogenetic trees, vectors, and lists. See `get_time()` for the underlying procedure.

strat_to_time.list *transform list from height to time domain*

Description

Lists are useful to keep data closely associated. This function transforms a list that contains observations associated with a stratigraphic position (recorded in the element with name "h") into a list where the observations are associated with time.

Usage

```
## S3 method for class 'list'
strat_to_time(obj, x, ...)
```

Arguments

obj	a list with one element named "h", which will be interpreted as stratigraphic positions
x	an *adm* object
...	options passed to get_time

Value

a timelist (inherits from list). A list with one named element "t" instead of the element "h". This element contains the times of the stratigraphic positions in "h".

See Also

time_to_strat.list() for the transformation from time to height domain, get_time() for the underlying procedure, time_to_strat() for the higher level function

Examples

```
# see vignette("admtools") for an example
```

strat_to_time.numeric *transform numeric vectors from height to time domain*

Description

This function transforms numeric vectors from the stratigraphic to the time domain Fundamentally a wrapper around get_time for consistent syntax

Usage

```
## S3 method for class 'numeric'
strat_to_time(obj, x, ...)
```

Arguments

obj	a numeric vector representing stratigraphic positions.
x	an *adm* object
...	options passed to get_time

Value

A numeric vector with times of deposition of the entries in obj

See Also

time_to_strat.numeric() for the transformation from time to height domain, get_time() for the underlying procedure, time_to_strat() for the higher level function, strat_to_time.list() and strat_to_time.phylo() for the transformation of lists and phylogenetic trees.

Examples

```
# see vignette("admtools") for an example
```

---

strat_to_time.phylo *transform phylo object*

Description

transform phylo object from the stratigraphic domain to the time domain

Usage

```
## S3 method for class 'phylo'
strat_to_time(obj, x, ...)
```

Arguments

obj	the phylo object to be transformed
x	age-depth model
...	parameters passed to get_time

Value

a phylo object, representation of the tree in the time domain

See Also

get_time() for the underlying procedure, strat_to_time() for the higher level function, and time_to_strat.phylo() for the transformation of phylo objects from the time to the strat domain.

---

summary.adm *summary of age-depth model*

Description

Displays some summary numbers of an age-depth models

Usage

```
## S3 method for class 'adm'
summary(object, ...)
```

Arguments

object	an adm object
...	other variables, are ignored

Value

Invisible NULL, prints summary to the console

Examples

```
my_adm = tp_to_adm(t = 1:5, h = c(2,2,3), L_unit = "m", T_unit = "Myr" )
summary(my_adm)
```

---

summary.multiadm *summary of age-depth model*

Description

Displays some summary numbers of an age-depth models

Usage

```
## S3 method for class 'multiadm'
summary(object, ...)
```

Arguments

object	a multiadm object
...	other variables, are ignored

Value

Invisible NULL, prints summary to the console

---

summary.sac *summary of sediment accumulation curve*

Description

displays some summary numbers of sediment accumulation curve

Usage

```
## S3 method for class 'sac'
summary(object, ...)
```

Arguments

object	sediment accumulation curve (sac)
...	other variables, are ignored

Value

invisible NULL

---

timetree *example time tree*

Description

Time tree generated using the ape package. Code used to generate is
set.seed(1) tree_in_time = ape::rlineage(birth = 1.8, death = 0.2, Tmax = 2)

Usage

```
timetree
```

Format

An object of class phylo of length 4.

time_to_strat *transform objects from time domain to strat. domain*

Description

Takes an object and transforms it from the time domain into the stratigraphic domain using the provided age-depth model. Currently implemented for the "phylo", "list", and "numeric" class.

Usage

```
time_to_strat(obj, x, ...)
```

Arguments

obj	the object to be transformed
x	age-depth model for the transformation
...	other parameters

Value

an object of the same type as obj

See Also

strat_to_time() to transform data from the stratigraphic domain to the time domain, time_to_strat.phylo(), time_to_strat.numeric() and time_to_strat.list() for details on how to transform phylo objects, vectors, and lists. See get_height() for the underlying procedure.

time_to_strat.list *transform list from time to height domain*

Description

Lists are useful to keep data closely associated. This function transforms a list that contains observations associated with a time (recorded in the element with name "t") into a list where the observations are associated with stratigraphic position.

Usage

```
## S3 method for class 'list'
time_to_strat(obj, x, ...)
```

Arguments

obj	a list with one element named "t", which will be interpreted as time
x	an *adm* object
...	options passed to *get_height*

Value

a stratlist (inherits from list): A list with one named element "h" instead of the element "t", containing the stratigraphic positions corresponding to the times inf "t"

See Also

strat_to_time.list() for the transformation from height to time domain, time_to_strat.phylo() and time_to_strat.numeric() for transformations of phylogenetic trees and vectors. See get_height() for the underlying procedure.

Examples

```
# see vignette("admtools") for an example
```

---

time_to_strat.numeric *transform vectors from time to height domain*

Description

This function transforms numeric vectors from the time to the stratigraphic domain Fundamentally a wrapper around get_height for consistent syntax

Usage

```
## S3 method for class 'numeric'
time_to_strat(obj, x, ...)
```

Arguments

obj	a numeric vector, interpreted as timing of events
x	an *adm* object
...	options passed to *get_height*

Value

a numeric vector - stratigraphic position of the events

See Also

strat_to_time.numeric() for the transformation from height to time domain, time_to_strat.phylo() and time_to_strat.list() for transformations of phylogenetic trees and lists. See get_height() for the underlying procedure.

Examples

```
# see vignette("admtools") for an example
```

---

time_to_strat.phylo *transform phylo object*

Description

transform phylo object from the time domain to the stratigraphic domain

Usage

```
## S3 method for class 'phylo'
time_to_strat(obj, x, ...)
```

Arguments

obj	the phylo object to be transformed
x	age-depth model
...	other parameters, currently ignored

Value

a phylo object, representation of the tree in the strat domain

See Also

get_height() for the underlying procedure, time_to_strat() for the higher level function, and strat_to_time.phylo() for the transformation of phylo objects from strat domain to the time domain. See time_to_strat.list() and time_to_strat.numeric() for the transformation of lists and numeric vectors

tp_height_det *deterministic tie points height domain*

Description

defines deterministic stratigraphic tie points

Usage

```
tp_height_det(heights)
```

Arguments

heights	numeric vector. Stratigraphic positions of the tie points

Value

a function for usage with *strat_cont_to_multiadm* and *sedrate_to_multiamd* as h_tp input

See Also

tp_time_det() for deterministic tie points in time, tp_time_norm() for tie points following a normal distribution, tp_time_floating_scale() for tie points for a floating scale,

tp_time_det *deterministic tie points in time domain*

Description

defines deterministic tie points in time.

Usage

```
tp_time_det(times)
```

Arguments

times	numeric vector, times of the tie points

Value

a function for usage with *strat_cont_to_multiadm* and *sedrate_to_mulitadm* as t_tp input

See Also

tp_height_det() for deterministic tie points in height, tp_time_norm() for tie points following a normal distribution

tp_time_floating_scale *tie points for floating time scale*

Description

Defines tie points for a floating (auxiliary) time scale for usage with *sedrate_to_multiadm* and *strat_cont_to_multiadm* as t_tp input. This floating time scale consists of two tie points in time, the first at time t = 0, the second at time t = 1. `tp_time_floating_scale` is a synonym of `tp_time_det(times = c(0,1))`

Usage

```
tp_time_floating_scale()
```

Value

function for usage with *strat_cont_to_multiadm* and *sedrate_to_multiamd* as t_tp input

See Also

`tp_time_norm()` for tie points following a normal distribution, `tp_height_det()` for deterministic height tie points

Examples

```
## Not run:
# see this vignette for an example
vignette("adm_from_trace_cont")

## End(Not run)
```

tp_time_norm *time tie points with normal distribution*

Description

defines a function factory that returns normally distributed times. FOr usage with `sedrate_to_multiadm` and `strat_cont_to_multiadm`.

Usage

```
tp_time_norm(mean, sd, force_order = TRUE)
```

Arguments

mean	numeric vector, mean age of tie points
sd	numeric vector, standard deviation of tie points
force_order	logical, enforce strictly increasing times

Value

function for usage with *strat_cont_to_multiadm* and *sedrate_to_multiamd* as t_tp input

See Also

tp_time_floating_scale() for tie points for a floating scale, tp_height_det() for deterministic height tie points

---

tp_to_adm *Construct age-depth model from tie points*

Description

Turns tie points into an adm object that represents an age-depth model

Usage

```
tp_to_adm(t, h, T_unit = NULL, L_unit = NULL)
```

Arguments

t	Vector, tie points in time
h	Vector, tie points in height
T_unit	character, time unit
L_unit	character, length unit

Details

by default, intervals with no sediment accumulation are marked as destructive. tp_to_adm does not check whether the inputs define a valid age-depth model. For this, use is_adm

Value

object of class adm

See Also

is_adm() to check validity of adm objects, get_T_tp() and get_L_tp() to extract time and height/length tie points

Examples

```
my_adm = tp_to_adm(t = 1:4, h = c(1,2,2,3), T_unit = "kyr", L_unit = "m")
plot(my_adm)
# see vignette("admtools") for other examples
```

`tp_to_sac` *define sed. acc. curve*

Description

defines *sac* (sediment accumulation curve) object from tie points

Usage

```
tp_to_sac(t, h, T_unit = NULL, L_unit = NULL)
```

Arguments

`t`	numeric vector, time coordinates of tie points
`h`	numeric vector, height coordinates of tie points
`T_unit`	time unit
`L_unit`	length unit

Value

a *sac* object reflecting a sediment accumulation curve

See Also

`sac_to_adm()` to transform sediment accumulation curves into age-depth models, `get_T_tp()` and `get_L_tp()` to extract time and height/length tie points

`T_axis_lab` *plot time axis label*

Description

plot time axis label

Usage

```
T_axis_lab(
  label = "Time",
  unit = TRUE,
  sep = " ",
  brac = c("[", "]"),
  line = 2,
  outer = FALSE,
  at = NA,
  adj = NA,
  padj = NA,
  cex = NA,
  col = NA,
  font = NA,
  ...
)
```

Arguments

`label`	Axis label
`unit`	Logical or character, should unit be plotted
`sep`	separator between label and unit
`brac`	brackets surrounding unit
`line`	parameter passed to *mtext*, see ?mtext for details
`outer`	parameter passed to *mtext*, see ?mtext for details
`at`	parameter passed to *mtext*, see ?mtext for details
`adj`	parameter passed to *mtext*, see ?mtext for details
`padj`	parameter passed to *mtext*, see ?mtext for details
`cex`	parameter passed to *mtext*, see ?mtext for details
`col`	parameter passed to *mtext*, see ?mtext for details
`font`	parameter passed to *mtext*, see ?mtext for details
`...`	further graphical parameters passed to *mtext*, see ?mtext for details

Value

invisible NULL

See Also

`plot.adm()` for plotting of adms

Index