

# Package ‘admtools’

April 24, 2025

**Title** Estimate and Manipulate Age-Depth Models

**Version** 0.5.0

**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** FossilSim, 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** 2025-04-24 06:50:02 UTC

## Contents

add_adm_to_multiadm . . . . .	4
adm_to_ddc . . . . .	4
anchor . . . . .	5
CarboCATLite_data . . . . .	5
condensation . . . . .	6
condensation_fun . . . . .	7
flip_ddc . . . . .	7
flux_const . . . . .	8
flux_linear . . . . .	8
flux_quad . . . . .	9
get_completeness . . . . .	10
get_data_from_eTimeOpt . . . . .	10
get_height . . . . .	11
get_hiat_duration . . . . .	12
get_hiat_list . . . . .	12
get_hiat_no . . . . .	13
get_hiat_pos . . . . .	14
get_L_tp . . . . .	14
get_L_unit . . . . .	15
get_L_units . . . . .	15
get_section_names . . . . .	16
get_time . . . . .	16
get_total_duration . . . . .	17
get_total_thickness . . . . .	18
get_total_thicknesses . . . . .	18
get_T_tp . . . . .	19
get_T_unit . . . . .	19
is_adm . . . . .	20
is_destructive . . . . .	21
is_multiadm . . . . .	21
is_sac . . . . .	22
L_axis_lab . . . . .	22
make_legend . . . . .	23
max_height . . . . .	24
max_time . . . . .	24
mean_adm . . . . .	25
median_adm . . . . .	25
merge_adm_to_multiadm . . . . .	26
merge_multiadm . . . . .	26
min_height . . . . .	27
min_time . . . . .	27
plot.adm . . . . .	28
plot.ddc . . . . .	29
plot.multiadm . . . . .	29
plot.sac . . . . .	30
plot.stratlist . . . . .	30

plot.timelist . . . . .	31
plot_condensation . . . . .	31
plot_erosive_intervals . . . . .	32
plot_sed_rate_l . . . . .	32
plot_sed_rate_t . . . . .	33
quantile_adm . . . . .	33
sac_to_adm . . . . .	34
sedrate_to_multiadm . . . . .	35
sed_rate_from_matrix . . . . .	36
sed_rate_gen_from_bounds . . . . .	37
sed_rate_gen_gamma . . . . .	38
sed_rate_l . . . . .	39
sed_rate_l_fun . . . . .	39
sed_rate_t . . . . .	40
sed_rate_t_fun . . . . .	40
set_L_unit . . . . .	41
set_L_units . . . . .	41
set_section_names . . . . .	42
set_T_unit . . . . .	42
split_multiadm . . . . .	43
strat_cont_gen_from_tracer . . . . .	43
strat_cont_to_multiadm . . . . .	44
strat_to_time . . . . .	45
strat_to_time.fossils . . . . .	46
strat_to_time.list . . . . .	46
strat_to_time.numeric . . . . .	47
strat_to_time.phylo . . . . .	48
strat_to_time.taxonomy . . . . .	48
summary.adm . . . . .	49
summary.ddc . . . . .	49
summary.multiadm . . . . .	50
summary.sac . . . . .	50
timetree . . . . .	51
time_to_strat . . . . .	51
time_to_strat.fossils . . . . .	52
time_to_strat.list . . . . .	52
time_to_strat.numeric . . . . .	53
time_to_strat.phylo . . . . .	54
time_to_strat.taxonomy . . . . .	54
tp_height_det . . . . .	55
tp_time_det . . . . .	55
tp_time_floating_scale . . . . .	56
tp_time_norm . . . . .	57
tp_to_adm . . . . .	57
tp_to_ddc . . . . .	58
tp_to_sac . . . . .	59
T_axis_lab . . . . .	59

---

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

---

adm\_to\_ddc             *construct depth-depth curve from age-depth models*

---

**Description**

constructs a depth-depth curve from section 1 (described by adm1) to section 2 (described by adm2)

**Usage**

adm\_to\_ddc(adm1, adm2)

**Arguments**

adm1                  age-depth model 1. An adm object  
adm2                  age-depth model 2. An adm object

---

anchor	<i>anchor age-depth model</i>
--------	-------------------------------

---

### 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	<i>Example data from CarboCATLite</i>
-------------------	---------------------------------------

---

### 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; Jarochovska, 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; Jarochovska, 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 = "rc11", ...)

**Arguments**

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

---

**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 <i>sed_rate_t</i> , see ? <i>sed_rate_t</i> for details
...	parameters passed to <i>get_time</i> , see ? <i>get_time</i> for details

**Value**

a function

---

flip_ddc	<i>reverse direction of depth-depth curve</i>
----------	---

---

**Description**

reverses the direction of correlation of the depth-depth curve i.e., if the original depth-depth curve maps section 1 to section 2 the flipped curve maps section 2 to section 1

**Usage**

```
flip_ddc(x, ...)
```

**Arguments**

x                    a depth-depth curve (ddc object)  
 ...                   other parameters, currently unused

---

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	<i>quadratic deterministic tracer flux</i>
-----------	--

---

**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 to extract specific data, adjust the output parameter in eTimeOpt to return the correct data (e.g. 2 for r<sup>2</sup> 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	<i>determine stratigraphic height deposited at specific time</i>
------------	--

---

**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

---

<code>get_L_unit</code>	<i>extract length unit</i>
-------------------------	----------------------------

---

**Description**

extracts the length unit from sac, adm or multiadm object

**Usage**

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

**Arguments**

<code>x</code>	adm or multiadm object
<code>...</code>	other parameters

**Value**

character - the length unit of x

**See Also**

[get\\_T\\_unit\(\)](#) [set\\_L\\_unit\(\)](#)

---

<code>get_L_units</code>	<i>extract length units</i>
--------------------------	-----------------------------

---

**Description**

extracts the length unit from depth-depth curves (ddc objects)

**Usage**

```
get_L_units(x, ...)
```

**Arguments**

<code>x</code>	an ddc object
<code>...</code>	other parameters

**Value**

character - the length unit of x

**See Also**

[get\\_L\\_unit\(\)](#) [set\\_L\\_unit\(\)](#) to extract length units from age-depth models or sediment accumulation curves

---

get_section_names	<i>get section names from depth-depth curve</i>
-------------------	---

---

**Description**

get section names from depth-depth curve

**Usage**

```
get_section_names(x, ...)
```

**Arguments**

x	a depth-depth curve (ddc object)
...	other parameters (currently unused)

**Value**

vector of length 2, section names of the correlated sections

---

get_time	<i>Determine times based on age-depth model</i>
----------	---

---

**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 <i>adm</i> or <i>multiadm</i> 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	<i>Total duration covered</i>
--------------------	-------------------------------

---

**Description**

Total duration covered

**Usage**

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

**Arguments**

x	age-depth model ( <i>adm</i> / <i>multiadm</i> ) or sediment accumulation curve ( <i>sac</i> )
...	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\\_total\\_thicknesses\(\)](#) to extract thicknesses from depth-depth curves

---

`get_total_thicknesses`    *get thicknesses of sections from depth-depth curves*

---

**Description**

get thicknesses of sections from depth-depth curves

**Usage**

`get_total_thicknesses(x, ...)`

**Arguments**

`x`                    a depth-depth curve (ddc object)  
`...`                other parameters (currently unused)

**Value**

a numeric vector of length 2 - the thicknesses of section 1 and 2 in the depth-depth curve

**See Also**

- [get\\_total\\_thickness\(\)](#) to determine the total thickness accumulated by age-depth models or sediment accumulation curves.

---

get_T_tp	<i>extract time tie points</i>
----------	--------------------------------

---

**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	<i>extract Time unit</i>
------------	--------------------------

---

**Description**

extracts the Time unit from sac, adm or multiadm object

**Usage**

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

**Arguments**

x	sac, adm or multiadm object
...	other parameters

**Value**

character - the time unit of x

**See Also**

[set\\_T\\_unit\(\)](#) [get\\_L\\_unit\(\)](#)

---

is_adm	<i>Is an adm object a valid age-depth model</i>
--------	---

---

**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	<i>Is deposition destructive?</i>
----------------	-----------------------------------

---

**Description**

Determines whether specified time is destructive or not

**Usage**

```
is_destructive(x, t, mode = "rc11",
  bdry_pts_hiat = "destructive", out_dom_mode = "default")
```

**Arguments**

x	an <i>adm</i> or <i>multiadm</i> object
t	vector of times
mode	string, either "rc11", "lcr1", "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	<i>is valid multiadm object?</i>
-------------	----------------------------------

---

**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	<i>is valid sac objects</i>
--------	-----------------------------

---

**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	<i>plot height axis label</i>
------------	-------------------------------

---

**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 <i>mtext</i> , see ?mtext for details
outer	parameter passed to <i>mtext</i> , see ?mtext for details
at	parameter passed to <i>mtext</i> , see ?mtext for details
adj	parameter passed to <i>mtext</i> , see ?mtext for details
padj	parameter passed to <i>mtext</i> , see ?mtext for details
cex	parameter passed to <i>mtext</i> , see ?mtext for details
col	parameter passed to <i>mtext</i> , see ?mtext for details
font	parameter passed to <i>mtext</i> , see ?mtext for details
...	further graphical parameters passed to <i>mtext</i> , 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	<i>get highest stratigraphic tie point</i>
------------	--

---

**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	<i>last time tie point</i>
----------	----------------------------

---

**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	<i>get mean ADM</i>
----------	---------------------

---

**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	<i>get median ADM</i>
------------	-----------------------

---

**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 objects into multiadm object*

---

**Description**

combine multiple adm objects 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	<i>get lowest stratigraphic tie point</i>
------------	---

---

**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	<i>first time tie point</i>
----------	-----------------------------

---

**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.ddc	<i>plot depth-depth curve</i>
----------	-------------------------------

---

**Description**

plot depth-depth curve

**Usage**

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

**Arguments**

x	depth-depth curve
...	other parameters passed to plot

---

plot.multiadm	<i>plot multiadm object</i>
---------------	-----------------------------

---

**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	<i>plot sediment accumulation curve</i>
----------	---

---

**Description**

plot sediment accumulation curve

**Usage**

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

**Arguments**

x	object of class <i>sac</i>
...	further parameters (currently ignored)

**Value**

invisible NULL

---

plot.stratlist	<i>plot strat list</i>
----------------	------------------------

---

**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	<i>plot time lists</i>
---------------	------------------------

---

**Description**

plot time lists

**Usage**

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

**Arguments**

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

---

plot_condensation	<i>plot condensation in height</i>
-------------------	------------------------------------

---

**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 <i>sed_rate_t</i> , see ? <i>sed_rate_t</i> for details
...	parameters passed to <i>get_time</i> , see ? <i>get_time</i> 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 <i>rect</i> , see ?rect for details
angle	parameter passed to <i>rect</i> , see ?rect for details
col	parameter passed to <i>rect</i> , see ?rect for details
border	parameter passed to <i>rect</i> , see ?rect for details
lty	parameter passed to <i>rect</i> , see ?rect for details
lwd	parameter passed to <i>rect</i> , see ?rect for details

### Value

invisible NULL

---

plot\_sed\_rate\_1      *plot sed. rate in height*

---

### Description

plot sed. rate in height

### Usage

```
plot_sed_rate_1(x, h = "default", mode = "rc11", ...)
```



**Arguments**

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

**Value**

invisible null

---

plot_sed_rate_t	<i>plot sedimentation rate in time</i>
-----------------	--

---

**Description**

plot sedimentation rate in time

**Usage**

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

**Arguments**

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

**Value**

invisible NULL

---

quantile_adm	<i>get quantile ADM</i>
--------------	-------------------------

---

**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	<i>turn sed. acc curve into adm</i>
------------	-------------------------------------

---

**Description**

turn sed. acc curve into adm

**Usage**

```
sac_to_adm(x)
```

**Arguments**

x	object of class <i>sac</i>
---	----------------------------

**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 <i>integrate</i> , see ?stats::integrate for details
stop.on.error	logical passed to <i>integrate</i> , 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()` for construction sedimentation rate generators from simple bounds. See also the vignettes for details on how arbitrary sedimentation rates can be constructed.

**Examples**

```
# see vignette
# vignette("adm_from_sedrate")
# for more general examples
```

---

```
sed_rate_gen_from_bounds
```

```
seg rate gen from upper/lower bounds
```

---

**Description**

constructs a sedimentation rate generator for usage with `sedrate_to_multiadm` based on the following procedure: (1) determine stratigraphic points based on a Poisson point process with rate `rate` (2) at these points, determine the sedimentation rate based on a uniform distribution between the bounds provided by the input parameters (3) linearly interpolate between those points with sedimentation rate determined in step 2. This approach can be used to estimate age-depth models when only rough boundaries on sedimentation rates are available. Here, the uniform distribution is chosen to reflect that no other information other than maximum and minimum sed. rate is available.

**Usage**

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

**Arguments**

<code>h_l</code>	height values for lower bounds
<code>s_l</code>	sed rate values for lower bounds
<code>h_u</code>	height values for upper bounds
<code>s_u</code>	sed rate values for upper bounds
<code>rate</code>	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\\_gen\\_gamma\(\)](#) for sed. rate generator based on a gamma distribution

[sedrate\\_to\\_multiadm\(\)](#) for estimating age-depth models using the outputs, [sed\\_rate\\_from\\_matrix\(\)](#) for other means of defining sedimentation rates, the vignette on how to construct arbitrary sedimentation rate generators.

**Examples**

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

---

sed\_rate\_gen\_gamma      *sed. rate gen based on gamma distribution*

---

**Description**

generates a function factory for usage with `sedrate_to_multiadm`. At `h[i]`, the sedimentation rate is given by a gamma distribution with parameters `shapes[i]` and `rates[i]`. Between those values, sedimentation rate is linearly interpolated. Outside of the range of `h`, behavior is determined by the argument `rule` which is passed to `approxfun`. See there for details.

**Usage**

```
sed_rate_gen_gamma(h, shapes, rates, rule = 1)
```

**Arguments**

<code>h</code>	heights at which sedimentation rate is determined
<code>shapes</code>	shape parameters for the gamma distribution
<code>rates</code>	rate parameter for the gamma distribution
<code>rule</code>	an integer of length 1 or 2, see description for details

**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\\_gen\\_from\\_bounds\(\)](#) for sed. rate generator based on bounds on sedimentation rates.

---

sed_rate_l	<i>sedimentation rate in stratigraphic height</i>
------------	---

---

**Description**

determines instantaneous sedimentation rate at a specified stratigraphic position

**Usage**

```
sed_rate_l(x, h, mode = "rc11", ...)
```

**Arguments**

x	adm object
h	numeric vector, stratigraphic positions
mode	string, handed over to <i>sed_rate_t</i> , see ?sed_rate_t for details
...	parameters passed to <i>get_time</i> , 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	<i>sed rate in height function</i>
----------------	------------------------------------

---

**Description**

returns a function that determines sed. rates in height

**Usage**

```
sed_rate_l_fun(x, mode = "rc11", ...)
```

**Arguments**

x	an adm object
mode	string, handed over to <i>sed_rate_t</i> , see ?sed_rate_t for details
...	parameters passed to <i>get_time</i> , see ?get_time for details

**Value**

a function

---

sed_rate_t	<i>sedimentation rate in time domain</i>
------------	--

---

**Description**

infers the instantaneous sedimentation rate from adm objects

**Usage**

```
sed_rate_t(x, t, mode = "rc11")
```

**Arguments**

x	adm or multiadm object
t	vector of times at which sedimentation rates are determined
mode	string, "rc11" or "lcr1". 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	<i>sedimentation rate function</i>
----------------	------------------------------------

---

**Description**

returns a function that returns sedimentation rate

**Usage**

```
sed_rate_t_fun(x, mode = "rc11")
```

**Arguments**

x	an adm object
mode	string, "rc11" or "lcr1". Should the sedimentation rate be Right Continuous with Left Limits (rc11) or Left Continuous with Right Limits (lcr1)

**Value**

a function



---

set_L_unit	<i>set length units</i>
------------	-------------------------

---

**Description**

set length units for sac, adm and multiadm objects

**Usage**

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

**Arguments**

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

**Value**

an adm or multiadm object with the L unit assigned

**See Also**

[set\\_T\\_unit\(\)](#) [get\\_L\\_unit\(\)](#)

---

set_L_units	<i>set length units</i>
-------------	-------------------------

---

**Description**

set length units for depth-depth curves

**Usage**

```
set_L_units(x, L_units, ...)
```

**Arguments**

x	depth-depth curve (ddc object)
L_units	length units
...	further parameters

**Value**

an ddc object with the L unit assigned

**See Also**

[set\\_L\\_unit\(\)](#) [get\\_L\\_units\(\)](#)

---

set\_section\_names      *set section names for depth-depth curves*

---

**Description**

set section names for depth-depth curves

**Usage**

set\_section\_names(x, sec\_names, ...)

**Arguments**

x	a depth-depth curve (ddc object)
sec_names	vector of length 2, section names assigned to the depth-depth curve
...	other parameters (currently unused)

**Value**

a depth-depth curve with assigned section names

---

set\_T\_unit      *set time units*

---

**Description**

set time units for sac, adm and multiadm objects

**Usage**

set\_T\_unit(x, T\_unit, ...)

**Arguments**

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

**Value**

an sac, adm or multiadm object with the time unit assigned

**See Also**

[set\\_L\\_unit\(\)](#) [get\\_T\\_unit\(\)](#)

---

split_multiadm	<i>split multiadm objects into adm</i>
----------------	--

---

**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	<i>proxy values in strat domain</i>
----------------------------	-------------------------------------

---

**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 <i>integrate</i> , see <code>?stats::integrate</code> for details
stop.on.error	logical passed to <i>integrate</i> , see <code>?stats::integrate</code> 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	<i>transform objects from strat. to time domain</i>
---------------	---

---

**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.fossils *transform fossils object from FossilSim between time and strat domain*

---

### Description

transforms fossil objects from stratigraphic to time domain ‘

### Usage

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

### Arguments

obj	the fossils object
x	the age-depth model
...	further parameters passed to <code>_get_height</code> ‘

---

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 <i>adm</i> object
...	options passed to <code>get_time</code>

### 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("admttools") 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 <i>adm</i> object
...	options passed to <a href="#">get_time</a>

**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("admttools") 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 <code>get_time</code>

### 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.

---

strat\_to\_time.taxonomy  
*transform taxonomy object from strat to time domain*

---

### Description

transform taxonomy object from strat to time domain

### Usage

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

### Arguments

obj	taxonomy object as defined by the FossilSim package
x	age-depth model
...	further parameters passed to <code>get_time</code>



---

summary.adm	<i>summary of age-depth model</i>
-------------	-----------------------------------

---

**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.ddc	<i>summary of depth-depth curve</i>
-------------	-------------------------------------

---

**Description**

displays some summary numbers of a depth-depth curve (ddc object)

**Usage**

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

**Arguments**

object	depth-depth curve (ddc)
...	other variables, currently ignored

**Value**

invisible NULL

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	<i>example time tree</i>
----------	--------------------------

---

**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	<i>transform objects from time domain to strat. domain</i>
---------------	--

---

**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.fossils *transform fossils from FossilSim from time to strat domain*

---

### Description

The fossils object of the FossilSim package describe the location of fossil occurrences along a phylogeny. This function transforms the objects from the time to the stratigraphic domain.

### Usage

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

### Arguments

obj	the fossils object
x	the age-depth model to be used, and <i>adm</i> object
...	further parameters to be passed to to_height, e.g., is_destructive

### Value

a fossils object for usage with the FossilSim package

---

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 <i>adm</i> object
...	options passed to <i>get_height</i>

**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("admttools") 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 <i>adm</i> object
...	options passed to <i>get_height</i>

**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("admttools") 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

---

```
time_to_strat.taxonomy
      transform taxonomy from time to strat domain
```

---

**Description**

transforms taxonomy objects from time to strat domain. Ignored the destructive argument of *get\_height*, which is automatically set to FALSE

**Usage**

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

**Arguments**

obj	a taxonomy object used by FossilSim
x	an age-depth model object
...	further parameters passed to get_height, e.g., <i>is_destructive</i>

---

tp_height_det	<i>deterministic tie points height domain</i>
---------------	---

---

**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\_multiadm* 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	<i>deterministic tie points in time domain</i>
-------------	--

---

**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\_multitadm* 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\_multiadm* 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	<i>time tie points with normal distribution</i>
--------------	---

---

**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_multiadm` 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	<i>Construct age-depth model from tie points</i>
-----------	--

---

**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_ddc`
*construct depth-depth curve from coeval heights*


---

**Description**

constructs a depth-depth curve (`ddc` object) from a vector of coeval heights specified by `h1` and `h2`

**Usage**

```
tp_to_ddc(
  h1,
  h2,
  L_unit_1 = NULL,
  L_unit_2 = NULL,
  sec_1 = "unnamed section 1",
  sec_2 = "unnamed section 2"
)
```

**Arguments**

<code>h1</code>	height tie points in section 1
<code>h2</code>	height tie points in section 2
<code>L_unit_1</code>	Length unit in section 1
<code>L_unit_2</code>	Length unit in section 2
<code>sec_1</code>	name of section 1
<code>sec_2</code>	name of section 2

---

tp_to_sac	<i>define sed. acc. curve</i>
-----------	-------------------------------

---

**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	<i>plot time axis label</i>
------------	-----------------------------

---

**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 <i>mtext</i> , see ?mtext for details
outer	parameter passed to <i>mtext</i> , see ?mtext for details
at	parameter passed to <i>mtext</i> , see ?mtext for details
adj	parameter passed to <i>mtext</i> , see ?mtext for details
padj	parameter passed to <i>mtext</i> , see ?mtext for details
cex	parameter passed to <i>mtext</i> , see ?mtext for details
col	parameter passed to <i>mtext</i> , see ?mtext for details
font	parameter passed to <i>mtext</i> , see ?mtext for details
...	further graphical parameters passed to <i>mtext</i> , see ?mtext for details

### Value

invisible NULL

### See Also

[plot.adm\(\)](#) for plotting of adms

# Index

## \* datasets

- CarboCATLite\_data, 5
- timetree, 51

add\_adm\_to\_multiadm, 4

adm\_to\_ddc, 4

anchor, 5

CarboCATLite\_data, 5

condensation, 6

condensation\_fun, 7

flip\_ddc, 7

flux\_const, 8

flux\_const(), 9, 44

flux\_linear, 8

flux\_linear(), 8, 9, 44

flux\_quad, 9

flux\_quad(), 8, 9, 44

get\_completeness, 10

get\_data\_from\_eTimeOpt, 10

get\_data\_from\_eTimeOpt(), 37

get\_height, 11

get\_height(), 51, 53, 54

get\_hiat\_duration, 12

get\_hiat\_duration(), 13, 14

get\_hiat\_list, 12

get\_hiat\_list(), 12–14

get\_hiat\_no, 13

get\_hiat\_no(), 12–14

get\_hiat\_pos, 14

get\_hiat\_pos(), 12, 13

get\_incompleteness (get\_completeness), 10

get\_L\_tp, 14

get\_L\_tp(), 19, 58, 59

get\_L\_unit, 15

get\_L\_unit(), 16, 20, 41

get\_L\_units, 15

get\_L\_units(), 42

get\_section\_names, 16

get\_T\_tp, 19

get\_T\_tp(), 15, 58, 59

get\_T\_unit, 19

get\_T\_unit(), 15, 42

get\_time, 16

get\_time(), 45, 47, 48

get\_total\_duration, 17

get\_total\_duration(), 24, 27

get\_total\_thickness, 18

get\_total\_thickness(), 19, 24, 27

get\_total\_thicknesses, 18

get\_total\_thicknesses(), 18

is\_adm, 20

is\_adm(), 58

is\_destructive, 21

is\_multiadm, 21

is\_sac, 22

L\_axis\_lab, 22

L\_axis\_lab(), 28

make\_legend, 23

max\_height, 24

max\_height(), 18, 27

max\_time, 24

max\_time(), 17, 27

mean\_adm, 25

mean\_adm(), 25, 34

median\_adm, 25

median\_adm(), 25, 34

merge\_adm\_to\_multiadm, 26

merge\_multiadm, 26

min\_height, 27

min\_height(), 18, 24

min\_time, 27

min\_time(), 17, 24

plot.adm, 28

plot.adm(), 23, 60  
 plot.ddc, 29  
 plot.multiadm, 29  
 plot.sac, 30  
 plot.stratlist, 30  
 plot.timelist, 31  
 plot\_condensation, 31  
 plot\_erosive\_intervals, 32  
 plot\_sed\_rate\_l, 32  
 plot\_sed\_rate\_t, 33  
  
 quantile\_adm, 33  
 quantile\_adm(), 25  
  
 sac\_to\_adm, 34  
 sac\_to\_adm(), 59  
 sed\_rate\_from\_matrix, 36  
 sed\_rate\_from\_matrix(), 11, 38  
 sed\_rate\_gen\_from\_bounds, 37  
 sed\_rate\_gen\_from\_bounds(), 37, 38  
 sed\_rate\_gen\_gamma, 38  
 sed\_rate\_gen\_gamma(), 38  
 sed\_rate\_l, 39  
 sed\_rate\_l\_fun, 39  
 sed\_rate\_t, 40  
 sed\_rate\_t\_fun, 40  
 sedrate\_to\_multiadm, 35  
 sedrate\_to\_multiadm(), 11, 37, 38  
 set\_L\_unit, 41  
 set\_L\_unit(), 15, 16, 42  
 set\_L\_units, 41  
 set\_section\_names, 42  
 set\_T\_unit, 42  
 set\_T\_unit(), 20, 41  
 split\_multiadm, 43  
 strat\_cont\_gen\_from\_tracer, 43  
 strat\_cont\_gen\_from\_tracer(), 8, 9  
 strat\_cont\_to\_multiadm, 44  
 strat\_to\_time, 45  
 strat\_to\_time(), 48, 51  
 strat\_to\_time.fossils, 46  
 strat\_to\_time.list, 46  
 strat\_to\_time.list(), 45, 47, 53  
 strat\_to\_time.numeric, 47  
 strat\_to\_time.numeric(), 45, 53  
 strat\_to\_time.phylo, 48  
 strat\_to\_time.phylo(), 45, 47, 54  
 strat\_to\_time.taxonomy, 48  
 summary.adm, 49  
 summary.ddc, 49  
 summary.multiadm, 50  
 summary.sac, 50  
  
 T\_axis\_lab, 59  
 T\_axis\_lab(), 28  
 time\_to\_strat, 51  
 time\_to\_strat(), 45, 47, 54  
 time\_to\_strat.fossils, 52  
 time\_to\_strat.list, 52  
 time\_to\_strat.list(), 47, 51, 53, 54  
 time\_to\_strat.numeric, 53  
 time\_to\_strat.numeric(), 47, 51, 53, 54  
 time\_to\_strat.phylo, 54  
 time\_to\_strat.phylo(), 48, 51, 53  
 time\_to\_strat.taxonomy, 54  
 timetree, 51  
 tp\_height\_det, 55  
 tp\_height\_det(), 56, 57  
 tp\_time\_det, 55  
 tp\_time\_det(), 55  
 tp\_time\_floating\_scale, 56  
 tp\_time\_floating\_scale(), 55, 57  
 tp\_time\_norm, 57  
 tp\_time\_norm(), 55, 56  
 tp\_to\_adm, 57  
 tp\_to\_adm(), 34  
 tp\_to\_ddc, 58  
 tp\_to\_sac, 59