

# Package ‘admtools’

July 17, 2024

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

**Version** 0.3.0

**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, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**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** 2024-07-17 16:20:01 UTC

## Contents

|                               |   |
|-------------------------------|---|
| add_adm_to_multiadm . . . . . | 3 |
| CarboCATLite_data . . . . .   | 4 |
| condensation . . . . .        | 5 |
| condensation_fun . . . . .    | 5 |

|                                    |    |
|------------------------------------|----|
| flux_const . . . . .               | 6  |
| flux_linear . . . . .              | 6  |
| flux_quad . . . . .                | 7  |
| get_completeness . . . . .         | 8  |
| get_data_from_eTimeOpt . . . . .   | 8  |
| get_height . . . . .               | 9  |
| get_hiat_duration . . . . .        | 10 |
| get_hiat_list . . . . .            | 10 |
| get_hiat_no . . . . .              | 11 |
| get_hiat_pos . . . . .             | 12 |
| get_L_unit . . . . .               | 12 |
| get_time . . . . .                 | 13 |
| get_total_duration . . . . .       | 14 |
| get_total_thickness . . . . .      | 14 |
| get_T_unit . . . . .               | 15 |
| is_adm . . . . .                   | 15 |
| is_destructive . . . . .           | 16 |
| is_multiadm . . . . .              | 16 |
| is_sac . . . . .                   | 17 |
| L_axis_lab . . . . .               | 17 |
| make_legend . . . . .              | 18 |
| max_height . . . . .               | 19 |
| max_time . . . . .                 | 19 |
| mean_adm . . . . .                 | 20 |
| median_adm . . . . .               | 20 |
| merge_adm_to_multiadm . . . . .    | 21 |
| merge_multiadm . . . . .           | 21 |
| min_height . . . . .               | 22 |
| min_time . . . . .                 | 22 |
| plot.adm . . . . .                 | 23 |
| plot.multiadm . . . . .            | 24 |
| plot.sac . . . . .                 | 24 |
| plot.stratlist . . . . .           | 25 |
| plot.timelist . . . . .            | 25 |
| plot_condensation . . . . .        | 26 |
| plot_erosive_intervals . . . . .   | 26 |
| plot_sed_rate_l . . . . .          | 27 |
| plot_sed_rate_t . . . . .          | 28 |
| quantile_adm . . . . .             | 28 |
| sac_to_adm . . . . .               | 29 |
| sedrate_to_multiadm . . . . .      | 29 |
| sed_rate_from_matrix . . . . .     | 30 |
| sed_rate_gen_from_bounds . . . . . | 31 |
| sed_rate_l . . . . .               | 32 |
| sed_rate_l_fun . . . . .           | 32 |
| sed_rate_t . . . . .               | 33 |
| sed_rate_t_fun . . . . .           | 33 |
| set_L_unit . . . . .               | 34 |

|                                      |    |
|--------------------------------------|----|
| set_T_unit . . . . .                 | 34 |
| split_multiadm . . . . .             | 35 |
| strat_cont_gen_from_tracer . . . . . | 35 |
| strat_cont_to_multiadm . . . . .     | 36 |
| strat_to_time . . . . .              | 37 |
| strat_to_time.list . . . . .         | 38 |
| strat_to_time.numeric . . . . .      | 39 |
| strat_to_time.phylo . . . . .        | 39 |
| summary.adm . . . . .                | 40 |
| summary.multiadm . . . . .           | 41 |
| timetree . . . . .                   | 41 |
| time_to_strat . . . . .              | 42 |
| time_to_strat.list . . . . .         | 42 |
| time_to_strat.numeric . . . . .      | 43 |
| time_to_strat.phylo . . . . .        | 44 |
| tp_height_det . . . . .              | 45 |
| tp_time_det . . . . .                | 45 |
| tp_time_floating_scale . . . . .     | 46 |
| tp_time_norm . . . . .               | 46 |
| tp_to_adm . . . . .                  | 47 |
| tp_to_sac . . . . .                  | 48 |
| T_axis_lab . . . . .                 | 48 |

## Index 50

---

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

---

CarboCATLite\_data      *Example data from CarboCATLite*

---

### Description

Data exported from CarboCATLite model run

### 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

---

|              |                                 |
|--------------|---------------------------------|
| condensation | <i>condensation with height</i> |
|--------------|---------------------------------|

---

**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 <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

---

|            |   |
|------------|---|
| flux_const | <i>constant deterministic tracer flux</i> |
|------------|---|

---

**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 | <i>linear deterministic tracer flux</i> |
|-------------|---|

---

**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 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 | <i>get hiatus positions</i> |
|--------------|-----------------------------|

---

**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_unit | <i>extract length unit</i> |
|------------|----------------------------|

---

**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 <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      *Total duration covered*

---

**Description**

Total duration covered

**Usage**

```
get_total_duration(x)
```

**Arguments**

x                      an adm object

**Value**

numeric vector, total duration covered by the age-depth models

**See Also**

[min\\_time\(\)](#) and [max\\_time\(\)](#) to extract the first/last tie point in time

---

get\_total\_thickness      *get thickness*

---

**Description**

get thickness

**Usage**

```
get_total_thickness(x)
```

**Arguments**

x                      an adm object

**Value**

numeric vector containing total sediment thickness accumulated

**See Also**

[max\\_height\(\)](#) and [min\\_height\(\)](#) to extract the highest/lowest stratigraphic point

---

|            |                          |
|------------|--------------------------|
| get_T_unit | <i>extract Time unit</i> |
|------------|--------------------------|

---

**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 | <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?

---

|                |                                   |
|----------------|-----------------------------------|
| 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 position in an adm</i> |
|------------|---|

---

**Description**

get highest stratigraphic position in an adm

**Usage**

```
max_height(x)
```

**Arguments**

x                    an age-depth model

**Value**

a scalar

**See Also**

[min\\_height\(\)](#), [get\\_total\\_thickness\(\)](#)

---

|          |                                       |
|----------|---------------------------------------|
| max_time | <i>extract earliest time from adm</i> |
|----------|---------------------------------------|

---

**Description**

extract earliest time from adm

**Usage**

```
max_time(x)
```

**Arguments**

x                    age-depth model

**Value**

a scalar, timing of earliest tie point in the adm

**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

---

|            |                       |
|------------|-----------------------|
| 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

---

*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 position in an adm</i> |
|------------|--|

---

**Description**

get lowest stratigraphic position in an adm

**Usage**

```
min_height(x)
```

**Arguments**

|   |                    |
|---|--------------------|
| x | an age-depth model |
|---|--------------------|

**See Also**

[get\\_total\\_thickness\(\)](#), [max\\_height\(\)](#)

---

|          |                                   |
|----------|-----------------------------------|
| min_time | <i>extract last time from adm</i> |
|----------|-----------------------------------|

---

**Description**

extract last time from adm

**Usage**

```
min_time(x)
```

**Arguments**

|   |                 |
|---|-----------------|
| x | age-depth model |
|---|-----------------|

**Value**

a scalar, timing of last tie point in the adm

**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 = "admttools")`

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 <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\_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_l | <i>plot sed. rate in height</i> |
|-----------------|---------------------------------|

---

**Description**

plot sed. rate in height

**Usage**

```
plot_sed_rate_l(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      *plot sedimentation rate in time*

---

**Description**

plot sedimentation rate in time

**Usage**

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

**Arguments**

x                      adm object  
mode                    string, "rc11" or "lc11". Should the sedimentation rate be Right Continuous with Left Limits (rc11) or Left Continuous with Right Limits (lc11)

**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

**Value**

an adm object

---

|            |                                     |
|------------|-------------------------------------|
| 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 *sac*

**Value**

object of class *adm*

**See Also**

[tp\\_to\\_adm\(\)](#) for the generator of *adm*

---

|                     |   |
|---------------------|---|
| sedrate_to_multiadm | <i>Estimate age-depth model from sedimentation rates &amp; tie points</i> |
|---------------------|---|

---

**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**

at height `height[i]`, the sedimentation rate is specified by the pdf `approxfun(sedrate, matrix[i,])`

**Usage**

```
sed_rate_from_matrix(height, sedrate, matrix, rate = 1, expand_domain = TRUE)
```

**Arguments**

|               |   |
|---------------|---|
| height        | vector of heights   |
| sedrate       | vector of sed. rates x values   |
| matrix        | matrix of sed rate y values   |
| 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 |

**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**

|                   |                                  |
|-------------------|----------------------------------|
| <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_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 retruns 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 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 | <i>set time units</i> |
|------------|-----------------------|

---

**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 | <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.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

|                  |  |
|------------------|--|
| <code>obj</code> | a numeric vector representing stratigraphic positions. |
| <code>x</code>   | an <i>adm</i> object                                   |
| <code>...</code> | options passed to <code>get_time</code>                |

### 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 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 | <i>summary of age-depth model</i> |
|------------------|-----------------------------------|

---

**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

---

|          |                          |
|----------|--------------------------|
| 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.list | <i>transform list from time to height domain</i> |
|--------------------|--|

---

### 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 in "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("admttools") for an example
```

---

```
time_to_strat.phylo  transfrom phylo object
```

---

**Description**

transfrom 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 | <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

*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\_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 | time unit                    |
| L_unit | 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

**Examples**

```
## Not run:
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

## End(Not run)
```

---

|           |                               |
|-----------|-------------------------------|
| 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

---

|            |                             |
|------------|-----------------------------|
| 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, 4
- timetree, 41
  
- add\_adm\_to\_multiadm, 3
  
- CarboCATLite\_data, 4
- condensation, 5
- condensation\_fun, 5
  
- flux\_const, 6
- flux\_const(), 7, 36
- flux\_linear, 6
- flux\_linear(), 6, 7, 36
- flux\_quad, 7
- flux\_quad(), 6, 7, 36
  
- get\_completeness, 8
- get\_data\_from\_eTimeOpt, 8
- get\_data\_from\_eTimeOpt(), 31
- get\_height, 9
- get\_height(), 42–44
- get\_hiat\_duration, 10
- get\_hiat\_duration(), 11, 12
- get\_hiat\_list, 10
- get\_hiat\_list(), 10–12
- get\_hiat\_no, 11
- get\_hiat\_no(), 10–12
- get\_hiat\_pos, 12
- get\_hiat\_pos(), 10, 11
- get\_incompleteness (get\_completeness), 8
- get\_L\_unit, 12
- get\_L\_unit(), 15, 34
- get\_T\_unit, 15
- get\_T\_unit(), 13, 35
- get\_time, 13
- get\_time(), 38–40
- get\_total\_duration, 14
- get\_total\_duration(), 19, 22
- get\_total\_thickness, 14
  
- get\_total\_thickness(), 19, 22
  
- is\_adm, 15
- is\_adm(), 47
- is\_destructive, 16
- is\_multiadm, 16
- is\_sac, 17
  
- L\_axis\_lab, 17
- L\_axis\_lab(), 23
  
- make\_legend, 18
- max\_height, 19
- max\_height(), 14, 22
- max\_time, 19
- max\_time(), 14, 22
- mean\_adm, 20
- median\_adm, 20
- merge\_adm\_to\_multiadm, 21
- merge\_multiadm, 21
- min\_height, 22
- min\_height(), 14, 19
- min\_time, 22
- min\_time(), 14, 19
  
- plot.adm, 23
- plot.adm(), 18, 49
- plot.multiadm, 24
- plot.sac, 24
- plot.stratlist, 25
- plot.timelist, 25
- plot\_condensation, 26
- plot\_erosive\_intervals, 26
- plot\_sed\_rate\_l, 27
- plot\_sed\_rate\_t, 28
  
- quantile\_adm, 28
  
- sac\_to\_adm, 29
- sed\_rate\_from\_matrix, 30
- sed\_rate\_from\_matrix(), 9, 31

sed\_rate\_gen\_from\_bounds, 31  
sed\_rate\_l, 32  
sed\_rate\_l\_fun, 32  
sed\_rate\_t, 33  
sed\_rate\_t\_fun, 33  
sedrate\_to\_multiadm, 29  
sedrate\_to\_multiadm(), 9, 31  
set\_L\_unit, 34  
set\_L\_unit(), 13, 35  
set\_T\_unit, 34  
set\_T\_unit(), 15, 34  
split\_multiadm, 35  
strat\_cont\_gen\_from\_tracer, 35  
strat\_cont\_gen\_from\_tracer(), 6, 7  
strat\_cont\_to\_multiadm, 36  
strat\_to\_time, 37  
strat\_to\_time(), 40, 42  
strat\_to\_time.list, 38  
strat\_to\_time.list(), 38, 39, 43  
strat\_to\_time.numeric, 39  
strat\_to\_time.numeric(), 38, 44  
strat\_to\_time.phylo, 39  
strat\_to\_time.phylo(), 38, 39, 44  
summary.adm, 40  
summary.multiadm, 41

T\_axis\_lab, 48  
T\_axis\_lab(), 23  
time\_to\_strat, 42  
time\_to\_strat(), 38, 39, 44  
time\_to\_strat.list, 42  
time\_to\_strat.list(), 38, 42, 44  
time\_to\_strat.numeric, 43  
time\_to\_strat.numeric(), 39, 42–44  
time\_to\_strat.phylo, 44  
time\_to\_strat.phylo(), 40, 42–44  
timetree, 41  
tp\_height\_det, 45  
tp\_height\_det(), 45–47  
tp\_time\_det, 45  
tp\_time\_det(), 45  
tp\_time\_floating\_scale, 46  
tp\_time\_floating\_scale(), 45, 47  
tp\_time\_norm, 46  
tp\_time\_norm(), 45, 46  
tp\_to\_adm, 47  
tp\_to\_adm(), 29  
tp\_to\_sac, 48