Package 'stratcols'

August 28, 2025

Title Stratigraphic Columns and Order Metrics

Version 1.0.0

Description Quantify stratigraphic disorder using the metrics defined by Burgess (2016) <doi:10.2110/jsr.2016.10>. Contains a range of utility tools to construct and manipulate stratigraphic columns.

License Apache License (>= 2)

Encoding UTF-8

RoxygenNote 7.3.2

Imports StratigrapheR

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

VignetteBuilder knitr

Config/testthat/edition 3

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

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

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-08-28 13:40:02 UTC

Contents

`as_stratcol` *define stratigraphic column*

Description

defines an S3 object `stratcol` representing a stratigraphic column. Does not check for the validity of the constructed object. For this, use `is_stratcol`

Usage

```
as_stratcol(thickness, facies, L_unit = NULL, base = 0)
```

Arguments

`thickness`	numeric vector, bed thicknesses
`facies`	vector, facies code of beds (numeric or character)
`L_unit`	length unit of bed thickness
`base`	position of lowest bed boundary

Value

an object of S3 class `stratcol`

See Also

`is_stratcol()` to check for validity

Examples

```
n_beds = 10
# 10 beds with thickness between 0.1 and 1 m
thickness = runif(n_beds, 0.1, 1)
# alternations of sand and shale
fa = rep(c("sand", "shale"), 5)
# length unit
L_unit = "m"
base = 2   # start section at 2 m height
s = as_stratcol(thickness, fa, L_unit, base)
```

---

bed_thickness *extract bed thicknesses*

Description

extracts bed thicknesses from stratigraphic column

Usage

```
bed_thickness(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

a numeric vector of bed thicknesses

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
thickness = bed_thickness(s)
hist(thickness, main = "Bed thickness", xlab = paste0("Thickness (m)"))
```

facies_names *extract facies names from stratigraphic column*

Description

extract facies names from stratigraphic column

Usage

```
facies_names(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

vector of facies names for each bed

See Also

`unique_facies_names()` to get a list of unique facies names

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
facies = facies_names(s)
print(facies)
```

facies_repetitions *have successive beds identical facies?*

Description

have successive beds identical facies?

Usage

```
facies_repetitions(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

TRUE or FALSE. Do at least two successive beds have the same facies?

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
facies_repetitions(s) # returns FALSE
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "sand", "shale", "shale"), L_unit = "m")
facies_repetitions(s) # returns TRUE
```

---

get_base *find base of stratigraphic column*

Description

find base of stratigraphic column

Usage

```
get_base(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

A number, position of lowest bed boundary in the stratigraphic column

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m", base = 2)
get_base(s) # returns 2
```

get_L_unit *extract length unit from stratigraphic columns*

Description

extract length unit from stratigraphic columns

Usage

```
get_L_unit(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

string or NULL, the length unit of the stratigraphic column

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
get_L_unit(s) # returns "m"
```

get_mom *Markov order metric (Burgess 2016)*

Description

Markov order metric (Burgess 2016)

Usage

```
get_mom(m)
```

Arguments

m	a facies transition matrix

Value

scalar, the Markov order metric introduced in Burgess (2016), https://doi.org/10.2110/jsr.2016.10

References

Burgess, Peter. 2016. "Identifying Ordered Strata: Evidence, Methods, and Meaning." Journal of Sedimentary Research. doi:10.2110/jsr.2016.10

See Also

transition_matrix() to estimate the facies transition matrix from a stratigraphic column, get_rom() to get the runs order metric

Examples

```
#see vignette for an extended example and explanation via
# vignette("stratorder")
# uniform bed thickness, ordered facies
s = as_stratcol(thickness = runif(30), fa = rep(c(1,2,3), 10))
s = shuffle_col(s, allow_rep = TRUE) # randomize order of beds, allowing  for repetitions
plot(s)
s_merged = merge_beds(s, mode = "identical facies")
plot(s_merged)
s_ord_names = order_facies_names(s_merged)
plot(s_ord_names)
m = transition_matrix(s_ord_names)
get_mom(m)
```

---

get_rom *runs order metric (Burgess 2016)*

Description

Determines the run order metric introduced in Burgess (2016), https://doi.org/10.2110/jsr.2016.10

Usage

```
get_rom(s, strictly = TRUE)
```

Arguments

s	stratigraphic column (a stratcol object)
strictly	logical. Does bed thickness need to be strictly increasing (>) or not (>=) to be counted as thickening?

Value

a number, the runs order metric (rom)

References

Burgess, Peter. 2016. "Identifying Ordered Strata: Evidence, Methods, and Meaning." Journal of Sedimentary Research. doi:10.2110/jsr.2016.10

See Also

get_mom() to get the Markov order metric

Examples

```
#see vignette for an extended example, bootstrapping methods and explanation via
# vignette("stratorder")
s = as_stratcol(thickness = runif(90), facies = rep(c(1,2,3), 30))
plot(s)
get_rom(s) # returns a number, the runs order metric
```

---

is_stratcol *is a valid stratigraphic column?*

Description

determines if x is a valid stratcol object

Usage

```
is_stratcol(x)
```

Arguments

x	stratigraphic column (a `stratcol` object)

Value

logical - is the object a valid stratcol object?

See Also

`as_stratcol()` to define stratcol objects

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
is_stratcol(s) # returns TRUE
s$fa = NULL # break stratcolumn object
is_stratcol(s) # returns FALSE
```

merge_beds *merge beds in stratigraphic column*

Description

merge beds in stratigraphic column

Usage

```
merge_beds(s, mode = "identical facies", ...)
```

Arguments

s	stratigraphic column (a stratcol object)
mode	character. criteria for merging. currently only "identical facies" is implemented
...	other parameters. currently not used

Value

a stratigraphic column (a stratcol object)

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "sand", "shale", "shale"), L_unit = "m")
merge_beds(s, mode = "identical facies")
facies = facies_names(s) # returns "sand" "shale" as the two sandy beds are merged
```

no_beds *number of beds*

Description

number of beds

Usage

```
no_beds(s)
```

Arguments

s	stratigraphic column (a stratcol object)

Value

integer, the number of beds

Examples

```
s =  as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
no_beds(s) # returns 4
```

`no_facies` *number of distinct facies*

Description

number of distinct facies

Usage

```
no_facies(s)
```

Arguments

`s`	stratigraphic column

Value

an integer

`order_facies_names` *order facies names according to appearance*

Description

enumerates the facies according to their order of appearance (counting from the bottom of the section). To be applied to stratigraphic columns before get_mom is called. Replaces the facies codes by integer numbers

Usage

```
order_facies_names(s)
```

Arguments

`s`	stratigraphic column (a `stratcol` object)

Value

a stratigraphic column (a `stratcol` object)

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "clay"), L_unit = "m")
s = order_facies_names(s)
plot(s)
```

`plot.stratcol` *basic plotting of stratigraphic columns*

Description

wraps around `StratigrapheR::litholog()` to plot a stratigraphic column. The beds are plotted as polygons, the boundaries as horizontal lines.

Usage

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

Arguments

`x`	stratigraphic column (a `stratcol` object)
`...`	further plotting options. ignored

Value

invisible NULL

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c(1,2,3,1.5), L_unit = "m")
# facies codes are used as hardness
plot(s)
```

`print.stratcol` *print stratigraphic column to console*

Description

print stratigraphic column to console

Usage

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

Arguments

x	stratigraphic column (a stratcol object)
...	other parameters (currently ignored)

Value

invisible NULL, prints to the console

See Also

summary.stratcol() for a summary of a stratigraphic column

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
print(s)
```

---

rename_facies *rename facies*

Description

replaces old facies names with new ones

Usage

```
rename_facies(s, new_names, old_names = NULL)
```

Arguments

s	stratigraphic column (a stratcol object)
new_names	new facies names
old_names	NULL or a list of old facies names. If NULL, all old facies names will be used

Value

stratigraphic column (a stratcol object) with renamed facies

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
s = rename_facies(s, new_names = c("sandy", "shaly"))
```

set_L_unit *set length unit of strat column*

Description

set length unit of strat column

Usage

```
set_L_unit(s, L_unit)
```

Arguments

s	stratigraphic column (a stratcol object)
L_unit	string or NULL, the length unit

Value

a stratigraphic column (stratcol object) with length unit added

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"))
s = set_L_unit(s, "m")
get_L_unit(s) # returns "m"
```

shuffle_col *rearrange stratigraphic column*

Description

rearrange stratigraphic column

Usage

```
shuffle_col(s, allow_rep = TRUE, max_no_swaps = 10^5)
```

Arguments

s	stratigraphic column (a stratcol object)
allow_rep	logical. Are repetitions in facies allowed?
max_no_swaps	integer. If allow rep is FALSE, what is the number of permutations used to shuffle the column?

Value

a stratcol object, the rearranged stratigraphic column

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("clay", "shale", "sand", "shale"), L_unit = "m")
s = shuffle_col(s, allow_rep = TRUE)
facies_names(s) # returns a random permutation of the facies
```

---

summary.stratcol *summarize stratigraphic column*

Description

summarize stratigraphic column

Usage

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

Arguments

`object`	stratigraphic column (a `stratcol` object)
`...`	further parameters (currently ignored)

Value

invisible NULL. prints to the console

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m", base = 2)
summary(s)
```

total_thickness *get total thickness*

Description

get total thickness

Usage

```
total_thickness(s, ...)
```

Arguments

s	stratigraphic column (a stratcol object)
...	other parameters (currently ignored)

Value

scalar, total thickness of stratigraphic column

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
total_thickness(s) # returns 2.5
```

total_thickness.stratcol *get total thickness of stratigraphic column*

Description

get total thickness of stratigraphic column

Usage

```
## S3 method for class 'stratcol'
total_thickness(s, ...)
```

Arguments

s	stratigraphic column
...	other parameters

Value

scalar, thickness of column

transition_matrix *transition frequency matrix from strat. column*

Description

transition frequency matrix from strat. column

Usage

```
transition_matrix(s)
```

Arguments

s	stratigraphic column (a `stratcol` object)

Value

a matrix of S3 class fa_tran_mat (facies transition matrix). Has dimension names "from" and "to", and facies as row/column names.

See Also

`trans_count_matrix()` for the facies transition matrix with raw transition counts

`get_mom()` to get the Markov order of the transition matrix

trans_count_matrix *facies transition count matrix*

Description

determines the number of facies transitions in a stratigraphic column and stores the output in a matrix

Usage

```
trans_count_matrix(s, ...)
```

Arguments

s	stratigraphic column (a `stratcol` object)
...	other parameters. currently ignored

Value

a transition count matrix of S3 class `fa_tran_mat_c`

See Also

transition_matrix() for the facies transition matrix with transition frequencies

Examples

```
#stratigraphic column with 90 beds
s = as_stratcol(thickness = runif(90), facies = rep(c(1,2,3), 30))
m = trans_count_matrix(s)
```

---

unique_facies_names *return unique facies names from a stratigraphic column*

Description

return unique facies names from a stratigraphic column

Usage

```
unique_facies_names(s)
```

Arguments

s	stratigraphic column (a stratcol object)

Value

a vector of unique facies names in the stratigraphic column

See Also

facies_names() to get facies names for each bed

Examples

```
s = as_stratcol(c(0.5, 1, 0.3, 0.7), c("sand", "shale", "sand", "shale"), L_unit = "m")
unique_facies = unique_facies_names(s) # returns c("sand", "shale")
```

Index