Package: StratPal (via r-universe)

September 13, 2024

Title Stratigraphic Paleobiology Modeling Pipelines

Version 0.1.1.9000

Description The fossil record is a joint expression of ecological, taphonomic, evolutionary, and stratigraphic processes (Holland and Patzkowsky, 2012, ISBN:978-0226649382). This package allowing to simulate biological processes in the time domain (e.g., trait evolution, fossil abundance), and examine how their expression in the rock record (stratigraphic domain) is influenced based on age-depth models, ecological niche models, and taphonomic effects. Functions simulating common processes used in modeling trait evolution or event type data such as first/last occurrences are provided and can be used standalone or as part of a pipeline. The package comes with example data sets and tutorials in several vignettes, which can be used as a template to set up one's own simulation.

License Apache License (>= 2)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports admtools (>= 0.3.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 4.2)

LazyData true

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

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

Config/testthat/edition 3

Repository https://mindthegap-erc.r-universe.dev

RemoteUrl https://github.com/mindthegap-erc/stratpal

RemoteRef HEAD

RemoteSha c6a44cc6ef714e6e14ce1524ac90f625819e6e05

Contents

`apply_niche` *apply niche model to events*

Description

Models niches by removing events (fossil occurrences) when they are outside of their niche using the function `thin`. Combines the functions `niche_def` and `gc` ("gradient change") to determine how the taxons' collection probability changes with time/position. This is done by composing `niche_def` and `gc`. The result is then used as a thinning on the events `x`.

Usage

```
apply_niche(x, niche_def, gc)
```

Arguments

`x`	events, e.g. times/ages of fossil occurrences or their stratigraphic position.
`niche_def`	function, specifying the niche along a gradient. Should return 0 when taxon is outside of niche, and 1 when inside niche. Values between 0 and 1 are interpreted as collection probabilities.
`gc`	function, stands for "gradient change". Specifies how the gradient changes, e.g. with time

Value

numeric vector, timing/location of events (e.g. fossil ages/locations) preserved after the niche model is applied

See Also

apply_taphonomy() to model taphonomic effects based on the same principle, thin() for the underlying mathematical procedure. Basic niche models available are bounded_niche() and snd_niche()

Examples

```
## setup
# using water depth as gradient
 t = scenarioA$t_myr
 wd = scenarioA$wd_m[,"8km"]
 gc = approxfun(t, wd)
 plot(t, gc(t), type = "l", xlab = "Time", ylab = "water depth [m]",
  main = "gradient change with time")
 # define niche
# preferred wd 10 m, tolerant to intermediate wd changes (standard deviation 10 m), non-terrestrial
 niche_def = snd_niche(opt = 10, tol = 10, cutoff_val = 0)
 plot(seq(-1, 50, by = 0.5), niche_def(seq(-1, 50, by = 0.5)), type = "l",
 xlab = "water depth", ylab = "collection probability", main = "Niche def")
 # niche pref with time
 plot(t, niche_def(gc(t)), type = "l", xlab = "time",
 ylab = "collection probability", main = "collection probability with time")

 ## simulate fossil occurrences
 foss_occ = p3(rate = 100, from = 0, to = max(t))
 # foss occ without niche pref
 hist(foss_occ, xlab = "time")
 foss_occ_niche = apply_niche(foss_occ, niche_def, gc)
 # fossil occurrences with niche preference
 hist(foss_occ_niche, xlab = "time")

 # see also
 #vignette("event_data")
 # for a detailed example on niche modeling
```

---

apply_taphonomy *model taphonomic effects*

Description

Models taphonomy by combining the change in taphonomic conditions with the preservation potential as a function of taphonomic conditions to determine how preservation potential changes. This is then used to systematically remove (thin) the event data using thin.

Usage

```
apply_taphonomy(x, pres_potential, ctc)
```

Arguments

x	events, e.g. times/ages of fossil occurrences or their stratigraphic position.
pres_potential	function. Takes taphonomic conditions as input and returns the preservation potential (a number between 0 and 1)
ctc	function, change in taphonomic conditions (ctc) with time or stratigraphic position.

Value

numeric vector, location/timing of events (e.g. fossil occurrences) after the taphonomic filter is applied

See Also

apply_niche() for modeling niche preferences based on the same principle, thin() for the underlying mathematical procedure.

Examples

```
# see
#vignette("advanced_functionality")
# for details on usage
```

---

bounded_niche *define niche from boundaries*

Description

Defines a simple niche model where the niche defined is given by a lower limit (g_min) and an upper limie (g_max) of a gradient the taxon can tolerate

Usage

```
bounded_niche(g_min, g_max)
```

Arguments

g_min	lowest value of the gradient the taxon can tolerate
g_max	highest value of the gradient the taxon can tolerate

Value

a function describing the niche for usage with apply_niche. The function returns 1 if the taxon is within its niche (the gradient is between g_min and g_max), and 0 otherwise

See Also

snd_niche() for an alternative niche model, apply_niche() for the function that uses the function returned

Examples

```
x = seq(0, 10, by = 0.2)
f = bounded_niche(2,5)
plot(x, f(x), type = "l",
xlab = "Gradient", ylab = "Observation probability",
main = "Observation probability of taxon")

# see also
#vignette("event_data")
# for details how to use this functionality
```

---

ornstein_uhlenbeck *simulate ornstein-uhlenbeck (OU) process*

Description

Simulates an Ornstein-Uhlenbeck process using the Euler-Maruyama method. The process is simulated on a scale of `0.25 * min(diff(t))` and then interpolated to the values of `t`.

Usage

```
ornstein_uhlenbeck(t, mu = 0, theta = 1, sigma = 1, y0 = 0)
```

Arguments

`t`	times at which the process is simulated. Can be heterodistant
`mu`	scalar, long term mean
`theta`	scalar, mean reversion speed
`sigma`	positive scalar, strength of randomness
`y0`	scalar, initial value (value of process at the first entry of t)

Value

A list with two elements: `t` and `y`. `t` is a duplicate of the input `t`, `y` are the values of the OU process at these times. Output list is of S3 class `timelist` (inherits from `list`) and can thus be plotted directly using `plot`, see `?admtools::plot.timelist`

Examples

```
library("admtools") # required for plotting of results
t = seq(0, 3, by = 0.01)
l = ornstein_uhlenbeck(t, y0 = 3) # start away from optimum (mu)
plot(l, type = "l")
l2 = ornstein_uhlenbeck(t, y0 = 0) # start in optimum
lines(l2$t, l2$y, col = "red")
```

---

p3 *simulate Poisson point process*

Description

Simulates events in the interval `from` to `to` based on a Poisson point process with rate `rate`. If the parameter `n` is used, the number of fossils is conditioned to be `n` In the context of paleontology, these events can be interpreted as fossil occurrences or first/last occurrences of species. In this case, the rate is the average number of fossil occurrences (resp first/last occurrences) per unit

Usage

```
p3(rate, from, to, n = NULL)
```

Arguments

`rate`	strictly positive scalar, rate of events (avg events per unit)
`from`	lowest boundary of observed interval
`to`	upper boundary of observed interval
`n`	integer of NULL (default). Number of events to return. If NULL, the number is random and determined by the rate parameter

Value

a numeric vector with timing/location of events.

See Also

`p3_var_rate()` for the variable rate implementation

Examples

```
# for fossil occ.
x = p3(rate = 5, from = 0, to = 1) # 5 fossil occurrences per myr on avg.
hist(x, xlab = "Time (Myr)", ylab = "Fossil Occurrences" )

x = p3(rate = 3, from = 0, to = 4)
```

```
hist(x, main = paste0(length(x), " samples")) # no of events is random

x = p3(rate = 3, from = 0, to = 4, n = 10)
hist(x, main = paste0(length(x), " samples")) # no of events is fixed to n

# see also
#vignette("event_data")
# for details on usage and applications to paleontology
```

`p3_var_rate` *simulate variable rate Poisson point process*

Description

simulates events based on a variable rate Poisson point process. Rates can be either specified by a function passed to `x`, or by providing two vectors `x` and `y`. In this case the rate is specified by approxfun(x, y, rule = 2), i.e. by linear interpolation between the values of x (abscissa) and y (ordinate) In the context of paleontology, these events can be interpreted as fossil occurrences or first/last occurrences of species. In this case, the rate is the average number of fossil occurrences (resp first/last occurrences) per unit

Usage

```
p3_var_rate(x, y = NULL, from = 0, to = 1, f_max = 1, n = NULL)
```

Arguments

`x`	numeric vector or function. If x is a function, it is used to specify the variable rate. If x is a vector, x and y together specify the variable rate using linear interpolation
`y`	numeric vector. If specified, determines the variable rate. This is done by using linear interpolation between the values of y. Here x specifies the ordinate and y the abscissa
`from`	lower boundary of the observed interval
`to`	upper boundary of the observed
`f_max`	maximum value of `x` in the interval from `x_min` to `x_max`. If x attains values larger than `f_max` a warning is throw, `f_max` is adjusted, and sampling is started again
`n`	NULL or an integer. Number of events drawn. If NULL, the number of events is determined by the rate (specified by x and y). If an integer is passed, n events are returned.

Value

numeric vector, timing/location of events. Depending on the modeling framework, tyese events can represent location/age of fossils, or first/last occurrences of a group of taxa.

See Also

p3() for the constant rate implementation, rej_samp() for the underlying random number generation.

Examples

```
# assuming events are fossil occurrences
# then rate is the avg rate of fossil occ. per unit
#linear decrease in rate from 50 at x = 0 to 0 at x = 1
x = c(0, 1)
y = c(50, 0)
s = p3_var_rate(x, y, f_max = 50)
hist(s, xlab = "Time (myr)", main = "Fossil Occurrences")
# conditioned to return 100 samples
s = p3_var_rate(x, y, f_max = 50, n = 100)
# hand over function
s = p3_var_rate(x = sin, from = 0 , to = 3 * pi, n = 50)
hist(s) # note that negative values of f (sin) are ignored in sampling

# see also
#vignette("event_data")
# for details on usage and applications to paleontology
```

---

random_walk *continuous time random walk*

Description

Simulates a (continuous time) random walk as a Brownian drift

Usage

```
random_walk(t, sigma = 1, mu = 0, y0 = 0)
```

Arguments

t	numeric vector with strictly increasing elements, can be heterodistant. Times at which the random walk is evaluated
sigma	positive scalar, variance parameter
mu	scalar, directionality parameter
y0	scalar, starting value (value of the random walk at the first entry of t)

Value

A list with elements t and y. t is a duplicate of the input parameter and is the times at which the random walk is evaluated. y are the values of the random walk at said times. Output list is of S3 class timelist (inherits from list) and can thus be plotted directly using plot, see ?admtools::plot.timelist

Examples

```
library("admtools") # required for plotting of results
t = seq(0, 1, by = 0.01)
l = random_walk(t, sigma = 3) # high variability, no direction
plot(l, type = "l")
l2 = random_walk(t, mu = 1) # low variabliity, increasing trend
lines(l2$t, l2$y, col = "red")
```

`rej_samp` *rejection sampling*

Description

Rejection sampling from the (pseudo) pdf `f` in the interval between `x_min` and `x_max`. Returns `n` samples. Note that values of `f` below 0 are capped to zero

Usage

```
rej_samp(f, x_min, x_max, n = 1L, f_max = 1, max_try = 10^4)
```

Arguments

`f`	function. (pseudo) pdf from which the sample is drawn
`x_min`	scalar. lower limit of the examined interval
`x_max`	scalar. upper limit of the examined interval
`n`	integer. number of samples drawn
`f_max`	maximum value of `f` in the interval from `x_min` to `x_max`. If `f` attains values larger than `f_max` a warning is throw, `f_max` is adjusted, and sampling is started again
`max_try`	maximum number of tries in the rejection sampling algorithm. If more tries are needed, an error is thrown. If this is the case, inspect of your function `f` is well-defined and positive, and if `f_max` provides a reasonable upper bound on it. Adjust `max_try` if you are certain that both is the case, e.g. if `f` is highly irregular.

Value

numeric vector, sample of size `n` drawn from the (pseudo) pdf specified by `f`

See Also

`p3_var_rate()` for the derived variable rate Poisson point process implementation.

Examples

```
f = sin
x = rej_samp(f, 0, 3*pi, n = 100)
hist(x) # note that no samples are drawn where sin is negative
```

`scenarioA` *example data, scenario A from Hohmann et al. (2024)*

Description

Scenario A as described in Hohmann et al. (2024), published in Hohmann et al. (2023). Contains data from a carbonate platform simulated using CarboCAT Lite (Burgess 2013, 2023)

Usage

```
scenarioA
```

Format

A list with 6 elements:

- `t_myr` : numeric vector. timesteps of the simulation in Myr
- `sl_m` : numeric vector. eustatic sea level in m
- `dist_from_shore` : character vector. Distance from shore in km of locations at which the observations were made
- `h_m` : matrix of size length(t_myr) x length(dist_from_shore). Accumulated sediment height in m at examined locations
- `wd_m`: matrix of size length(t_myr) x length(dist_from_shore). Water depth in m at examined locations
- `strat_col`: list with length(dist_from shore) elements. Represents a stratigraphic column. Each element is a list with two elements:
  - `bed_thickness_m`: numeric vector. Bed thickness in m
  - `facies_code` : integer vector. facies code of the bed

References

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

snd_niche *simple niche model*

Description

Defines niche model based in the "Probability of collection" model by Holland and Patzkowsky (1999). The collection probability follows the shape of a bell curve across a gradient, where `opt` determines the peak (mean) of the bell curve, and `tol` the standard deviation. "snd" stands for "scaled normal distribution", as the collection probability has the shape of the probability density of the normal distribution.

Usage

```
snd_niche(opt, tol, prob_modifier = 1, cutoff_val = NULL)
```

Arguments

`opt`	optimum value, gradient value where collection probability is highest
`tol`	tolerance to changes in gradient. For large values, collection probability drops off slower away from `opt`
`prob_modifier`	collection probability modifier, collection probability at `opt`.
`cutoff_val`	NULL or a number. If a number, all collection probabilities at gradient values below `cutoff_value` are set to 0. This can for example be used to model exclusively marine species when the gradient is water depth (see examples).

Value

a function for usage with `apply_niche`.

References

- Holland, Steven M. and Patzkowsky, Mark E. 1999. "Models for simulating the fossil record." Geology. https://doi.org/10.1130/0091-7613(1999)027%3C0491:MFSTFR%3E2.3.CO;2

See Also

`apply_niche()` for usage of the returned function, `bounded_niche()` for another niche model

Examples

```
# using water depth as niche
wd = seq(-3, 40, by = 0.5)
f = snd_niche(opt = 10, tol = 5)

plot(wd, f(wd), xlab = "Water depth", ylab = "Prob. of collection")
# set cutoff value at to 0 to model non-terrestrial species.
f = snd_niche(opt = 10, tol = 5, cutoff_val = 0)
```

```
plot(wd, f(wd), xlab = "Water depth", ylab = "Prob. of collection")

# see also
#vignette("event_data")
#for examples how to use it for niche modeling
```

`stasis` *simulate phenotypic stasis*

Description

Simulates stasis as independent, normally distributed random variables with mean `mean` and standard deviation `sd`

Usage

```
stasis(t, mean = 0, sd = 1)
```

Arguments

`t`	times at which the traits are determined
`mean`	scalar, mean trait value
`sd`	strictly positive scalar, standard deviation of traits

Value

A list with two elements: `t` and `y`. `t` is a duplicate of the input `t`, `y` are the corresponding trait values. Output list is of S3 class `timelist` (inherits from `list`) and can thus be plotted directly using `plot`, see `?admtools::plot.timelist`

Examples

```
library("admtools") # required for plotting of results
t = seq(0, 1, by = 0.01)
l = stasis(t)
plot(l, type = "l") # plot lineage
l2 = stasis(t, mean = 0.5, sd = 0.3) # simulate second lineage
lines(l2$t, l2$y, col = "red") # plot second lineage
```

`thin` *thin a series of events (e.g. fossil occurrences)*

Description

Thins a vector of events using the function thin, meaning the probability that the ith event in x is preserved is given by *thin(x(i))*. Values of `thin` below 0 and above 1 are ignored. Is used to model niche preferences in `apply_niche` and taphonomic effects in `apply_taphonomy`.

Usage

```
thin(x, thin)
```

Arguments

`x`	numeric vectors with events (e.g. locations, height, times)
`thin`	a function used for thinning

Value

numeric vector, events after thinning. Depending on the modeling framework, these events can represent fossil ages/locations or first/last occurrences, and the thinning taphonomic or ecological effects.

See Also

`apply_niche()` and `apply_taphonomy()` for use cases with biological meaning

Examples

```
x = p3(rate = 100, from = 0, to = 3 * pi) # simulate Poisson point process
y = thin(x, sin)
hist(y) # not how negative values of sin are treated as 0
yy = thin(x, function(x) 5 * sin(x))
hist(yy) # note how values of 5 * sin above 1 are not affecting the thinning
```

Index