

Package ‘mobsim’

November 2, 2017

Type Package

Title Spatial Simulation and Scale-Dependent Analysis of Biodiversity Changes

Version 0.1.0

Date 2017-11-02

Description Tools for the simulation, analysis and sampling of spatial biodiversity data (May et al. 2017) <doi:10.1101/209502>. In the simulation tools user define the numbers of species and individuals, the species abundance distribution and species aggregation. Functions for analysis include species rarefaction and accumulation curves, species-area relationships and the distance decay of similarity.

License GPL-3

LazyData TRUE

Imports Rcpp, vegan, sads (>= 0.4.1), grDevices, utils, graphics, stats

LinkingTo Rcpp

Suggests rmarkdown, spatstat, testthat, knitr

VignetteBuilder knitr

RoxygenNote 6.0.1

URL <http://github.com/MoBiodiv/mobsim>

BugReports <https://github.com/MoBiodiv/mobsim>

NeedsCompilation yes

Author Felix May [aut, cre]

Maintainer Felix May <felix.may@posteo.de>

Repository CRAN

Date/Publication 2017-11-02 15:31:05 UTC

R topics documented:

abund_rect	2
community	3
community_to_sad	4
dist_decay	4
divar	5
div_rand_rect	6
div_rect	7
mobsim	8
plot.community	9
plot.dist_decay	9
plot.divar	10
plot.sad	10
plot.spec_sample_curve	11
rare_curve	12
sample_quadrats	13
sim_poisson_community	14
sim_poisson_coords	15
sim_sad	16
sim_thomas_community	18
sim_thomas_coords	20
spec_sample	22
spec_sample_curve	23
summary.community	24
summary.sad	24
Index	25

abund_rect	<i>Get local species abundance distribution</i>
------------	---

Description

Get local abundance distribution in rectangle bounded by x_0 , y_0 , $x_0 + xsize$, $y_0 + ysize$

Usage

```
abund_rect(x0, y0, xsize, ysize, comm)
```

Arguments

x_0	x-coordinate of lower left corner
y_0	y-coordinate of lower left corner
xsize	Size of the subplot in x-direction
ysize	Size of the subplot in y-direction
comm	<code>community</code> object

Value

Integer vector with local species abundances

community	<i>Create spatial community object</i>
-----------	--

Description

Creates a spatial community object with defined extent and with coordinates and species identities of all individuals in the community.

Usage

```
community(x, y, spec_id, xrange = c(0, 1), yrange = c(0, 1))
```

Arguments

x, y	Coordinates of individuals (numeric)
spec_id	Species names or IDs; can be integers, characters or factors
xrange	Extent of the community in x-direction (numeric vector of length 2)
yrange	Extent of the community in y-direction (numeric vector of length 2)

Value

Community object which includes three items:

1. census: data.frame with three columns: x, y, and species names for each individual
2. x_min_max: extent of the community in x-direction
3. y_min_max: extent of the community in y-direction

Examples

```
x <- runif(100)
y <- runif(100)
species_names <- rep(paste("species", 1:10, sep = " "), each = 10)

com1 <- community(x,y, species_names)
plot(com1)
summary(com1)
```

community_to_sad	<i>Get species abundance distribution from community object</i>
------------------	---

Description

Get species abundance distribution from community object

Usage

```
community_to_sad(comm)
```

Arguments

comm Community object

Value

Object of class sad, which contains a named integer vector with species abundances

Examples

```
sim1 <- sim_poisson_community(s_pool = 200, n_sim = 20000, sad_type = "lnorm",
                             sad_coef = list("cv_abund" = 2))
sad1 <- community_to_sad(sim1)
plot(sad1, method = "rank")
plot(sad1, method = "octave")
```

dist_decay	<i>Distance decay of similarity</i>
------------	-------------------------------------

Description

Estimate pairwise similarities of communities in subplots as function of distance

Usage

```
dist_decay(comm, prop_area = 0.005, n_samples = 20, method = "bray",
           binary = F)
```

Arguments

comm [community](#) object
prop_area Subplot size as proportion of the total area
n_samples Number of randomly located subplots
method Choice of (dis)similarity index. See [vegdist](#)
binary Perform presence/absence standardization before analysis? See [vegdist](#)

Value

Dataframe with distances between subplot pairs and the respective similarity indices

Examples

```
sim_com1 <- sim_thomas_community(100, 10000, sigma = 0.1, mother_points = 2)
dd1 <- dist_decay(sim_com1, prop_area = 0.005, n_samples = 20)
plot(dd1)
```

 divar

Diversity-area relationships

Description

Estimate diversity indices in subplots of different sizes. This includes the well-known species-area and endemics-area relationships.

Usage

```
divar(comm, prop_area = seq(0.1, 1, by = 0.1), n_samples = 100,
       exclude_zeros = T)
```

Arguments

comm	community object
prop_area	Subplot sizes as proportion of the total area (numeric)
n_samples	Number of randomly located subplots per subplot size (single integer)
exclude_zeros	Should subplots without individuals be excluded? (logical)

Value

Dataframe with the proportional area of the subplots and mean and standard deviation of the following diversity indices:

1. Number of species
2. Number of endemics
3. Shannon index
4. Effective number of species (ENS) based on Shannon index
5. Simpson index
6. Effective number of species (ENS) based on Simpson index

See the documentation of [div_rect](#) for detailed information on the definition of the diversity indices.

See Also

[div_rand_rect](#), [div_rect](#)

Examples

```
sim1 <- sim_thomas_community(100, 1000)
divar1 <- divar(sim1, prop_area = seq(0.01, 1.0, length = 20))
plot(divar1)
```

div_rand_rect

Distribution of local diversity indices

Description

Get mean and standard deviation of diversity indices in several equally sized subplots of a community

Usage

```
div_rand_rect(prop_area = 0.25, comm, n_rect = 100, exclude_zeros = F)
```

Arguments

prop_area	Size of subplots as proportion of the total area
comm	community object
n_rect	Number of randomly located subplots
exclude_zeros	Should subplots without individuals be excluded? (logical)

Value

Vector with mean and standard deviation of the following diversity indices:

1. Number of species
2. Number of endemics
3. Shannon index
4. Effective number of species (ENS) based on Shannon index
5. Simpson index
6. Effective number of species (ENS) based on Simpson index

See the documentation of [div_rect](#) for detailed information on the definition of the diversity indices.

Examples

```
sim1 <- sim_poisson_community(100,1000)
div_rand_rect(prop_area = 0.1, comm = sim1)
```

div_rect
Get local diversity indices

Description

Get diversity indices including species richness, no. of endemics, Shannon and Simpson diversity for one rectangle subplot in the community.

Usage

```
div_rect(x0, y0, xsize, ysize, comm)
```

Arguments

x0	x-coordinate of lower left corner
y0	y-coordinate of lower left corner
xsize	Size of the subplot in x-direction
ysize	Size of the subplot in y-direction
comm	community object

Details

The effective number of species is defined as the number of equally abundant species that produce the same value of a certain diversity index as an observed community (Jost 2006). According to Chao et al. 2014 and Chiu et al. 2020 ENS_shannon can be interpreted as the number of common species and ENS_simpson as the number of dominant species in a community.

Value

Named vector with six diversity indices

1. n_species: Number of species
2. n_endemics: Number of endemics
3. shannon: Shannon index index defined as $H = -\sum p_i * \log(p_i)$, where p_i is the relative abundance of species i :
4. ens_shannon: Effective number of species (ENS) based on the Shannon index $\exp(H)$
5. simpson: Simpson index index (= probability of interspecific encounter PIE) defined as $D = 1 / \sum p_i^2$
6. ens_simpson: Effective number of species (ENS) based on the Simpson index $1/D$

References

- Jost 2006. Entropy and diversity. *Oikos*, 113, 363-375.
- Chao et al. 2014. Rarefaction and extrapolation with Hill numbers: a framework for sampling and estimation in species diversity studies. *Ecological Monographs*, 84, 45-67.
- Hsieh et al. 2016. iNEXT: an R package for rarefaction and extrapolation of species diversity (Hill numbers). *Methods Ecol Evol*, 7, 1451-1456.

Examples

```
sim1 <- sim_poisson_community(100,1000)
div_rect(0, 0, 0.3, 0.3, sim1)
```

mobsim

mobsim: A package for spatial analysis of scale-dependent biodiversity changes.

Description

The package includes functions to simulate species distributions in space as well as for the analysis of spatially-explicit data, where each individual is described by its xy-coordinates and a species identity label.

Functions to simulate species abundances and distributions

```
sim_sad
sim_poisson_coords
sim_thomas_coords
sim_poisson_community
sim_thomas_community
```

Functions to analyse species abundances and distributions

```
rare_curve
spec_sample_curve
divar
dist_decay
sample_quadrats
```

Author(s)

Felix May

plot.community *Plot spatial community object*

Description

Plot positions and species identities of all individuals in a community object.

Usage

```
## S3 method for class 'community'  
plot(x, ..., col = NULL, pch = NULL)
```

Arguments

x	Community object
...	Other parameters to plot
col	Colour vector to mark species identities
pch	Plotting character to mark species identities

Examples

```
sim1 <- sim_thomas_community(30, 500)  
plot(sim1)
```

plot.dist_decay *Plot distance decay of similarity*

Description

Plot distance decay of similarity

Usage

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

Arguments

x	Dataframe generated by dist_decay
...	Additional graphical parameters used in plot .

Details

The function plots the similarity indices between all pairs of subplots as function of distance. To indicate the relationship a [loess](#) smoother is added to the plot.

Examples

```
sim_com1 <- sim_thomas_community(100, 10000)
dd1 <- dist_decay(sim_com1)
plot(dd1)
```

<code>plot.divar</code>	<i>diversity-area relationships</i>	<i>Plot</i>
-------------------------	-------------------------------------	-------------

Description

 Plot diversity-area relationships
Usage

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

Arguments

`x` Dataframe generated function [divar](#).

`...` Additional graphical parameters used in [plot](#).

<code>plot.sad</code>	<i>Plot species abundance distributions</i>
-----------------------	---

Description

Plot species abundance distributions

Usage

```
## S3 method for class 'sad'
plot(x, ..., method = c("octave", "rank"))
```

Arguments

`x` Vector with species abundances (integer vector)

`...` Additional graphical parameters used in [plot](#) or [barplot](#)

`method` Plotting method, partial match to "octave" or "rank"

Details

With `method = "octave"` a histogram showing the number species in several abundance classes is generated. The abundance class are a simplified version of the "octaves" suggested by Preston (1948), which are based on log2-binning. The first abundance class includes species with 1 individual, the second with 2, the third with 3-4, the fourth with 5-8, etc.

With `method = "rank"` rank-abundance curve is generated with species abundance rank on the x-axis (descending) and species abundance on the y-axis (Hubbell 2001).

References

Preston 1948. The Commonness, and rarity, of species. *Ecology* 29(3):254-283.

Hubbell 2001. The unified neutral theory of biodiversity and biogeography. Princeton University Press.

Examples

```
abund1 <- sim_sad(s_pool = 100, n_sim = 10000, sad_type = "lnorm",
                sad_coef = list("cv_abund" = 1))
plot(abund1, method = "octave")
plot(abund1, method = "rank")
```

plot.spec_sample_curve

Plot species sampling curves

Description

Plot species sampling curves

Usage

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

Arguments

`x` Species sampling curve generated by [spec_sample_curve](#)
`...` Additional graphical parameters used in [plot](#).

Examples

```
sim_com1 <- sim_thomas_community(s_pool = 100, n_sim = 1000)
sac1 <- spec_sample_curve(sim_com1, method = c("rare", "acc"))
plot(sac1)
```

rare_curve	<i>Species rarefaction curve</i>
------------	----------------------------------

Description

Expected species richness as a function of sample size

Usage

```
rare_curve(abund_vec)
```

Arguments

abund_vec Species abundance distribution of the community (integer vector)

Details

This function essentially evaluates [spec_sample](#) for sample sizes from 1 to `sum(abund_vec)`. It is similar to the function [rarecurve](#) in the R package [vegan](#).

Value

Numeric Vector with expected species richness in samples of 1, 2, 3 ... n individuals

References

Gotelli & Colwell 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology Letters* 4, 379–391.

Examples

```
sad1 <- sim_sad(100, 2000, sad_type = "lnorm", sad_coef = list("meanlog" = 2,  
                                                            "sdlog" = 1))  
rc1 <- rare_curve(sad1)  
plot(rc1, type = "l", xlab = "Sample size", ylab = "Expected species richness")
```

sample_quadrats	<i>Plot-based samples from a spatially-explicit census</i>
-----------------	--

Description

This function allows to sample quadratic subplots from a spatially-explicit community. The output format are a sites x species abundance table and a sites x xy-coordinates table. The sites x species abundance is a classical data format used in community ecology. The table generated can be for instance be further analysed with the package [vegan](#).

Usage

```
sample_quadrats(comm, n_quadrats = 20, quadrat_area = 0.01, plot = TRUE,
  method = "random", avoid_overlap = F, x0 = 0, y0 = 0, delta_x = 0.1,
  delta_y = 0.1)
```

Arguments

comm	Community object from which the samples are generated
n_quadrats	Number of sampling quadrats
quadrat_area	Area of the sampling quadrats
plot	Should the sampling design be plotted? (logical)
method	Available methods are "random", "transect", "grid"
avoid_overlap	For the random sampling try to generate a design without overlap of quadrats (logical)
x0, y0	Lower left corner of the first quadrat in transect and grid sampling
delta_x	Distance between consecutive quadrats in transect and grid sampling in x-direction (the distance between the left sides is measured)
delta_y	Distance between consecutive quadrats in transect and grid sampling in y-direction (the distance between the lower sides is measured)

Value

A list with two items, spec_dat and xy_dat. spec_dat is a data.frame with sampling quadrats in rows and species abundances in columns, and xy_dat is a data.frame with sampling quadrats in rows and the xy-coordinates of the quadrats (lower left corner) in columns.

Examples

```
library(vegan)
sim_com1 <- sim_poisson_community(100, 10000)
comm_mat1 <- sample_quadrats(sim_com1, n_quadrats = 100,
  quadrat_area = 0.002, method = "grid")
specnumber(comm_mat1$spec_dat)
diversity(comm_mat1$spec_dat, index = "shannon")
```

sim_poisson_community *Simulate community with random spatial positions.*

Description

This function simulates a community with a certain abundance distribution and random spatial coordinates. This function consecutively calls [sim_sad](#) and [sim_poisson_coords](#)

Usage

```
sim_poisson_community(s_pool, n_sim, sad_type = "lnorm",
  sad_coef = list(cv_abund = 1), fix_s_sim = FALSE, xrange = c(0, 1),
  yrange = c(0, 1))
```

Arguments

s_pool	Number of species in the pool (integer)
n_sim	Number of individuals in the simulated community (integer)
sad_type	Root name of the species abundance distribution model of the species pool (character) - e.g., "lnorm" for the lognormal distribution (rlnorm); "geom" for the geometric distribution (rgeom), or "ls" for Fisher's log-series distribution (rls). See the table in Details below, or rsad for all SAD model options.
sad_coef	List with named arguments to be passed to the distribution function defined by the argument sad_type. An overview of parameter names is given in the table below. In <code>mobsim</code> the log-normal and the Poisson log-normal distributions can alternatively be parameterized by the coefficient of variation (cv) of the relative abundances in the species pool. Accordingly, <code>cv_abund</code> is the standard deviation of abundances divided by the mean abundance (no. of individuals / no. of species). <code>cv_abund</code> is thus negatively correlated with the evenness of the species abundance distribution. Please note that the parameters <i>mu</i> and <i>sigma</i> are not equal to the mean and standard deviation of the log-normal distribution.
fix_s_sim	Should the simulation constrain the number of species in the simulated local community? (logical)
xrange	Extent of the community in x-direction (numeric vector of length 2)
yrange	Extent of the community in y-direction (numeric vector of length 2)

Value

A community object as defined by [community](#).

Author(s)

Felix May

Examples

```
com1 <- sim_poisson_community(s_pool = 20, n_sim = 500, sad_type = "lnorm",
sad_coef = list("meanlog" = 2, "sdlog" = 1))
plot(com1)
```

sim_poisson_coords *Simulate random spatial coordinates*

Description

Add random spatial positions to a species abundance distribution.

Usage

```
sim_poisson_coords(abund_vec, xrange = c(0, 1), yrange = c(0, 1))
```

Arguments

abund_vec	Species abundance vector (integer)
xrange	Extent of the community in x-direction (numeric vector of length 2)
yrange	Extent of the community in y-direction (numeric vector of length 2)

Value

A community object as defined by [community](#).

Author(s)

Felix May

Examples

```
abund <- sim_sad(s_pool = 100, n_sim = 1000)
sim_com1 <- sim_poisson_coords(abund)
plot(sim_com1)
summary(sim_com1)
```

sim_sad *Simulate species abundance distributions*

Description

Simulate species abundance distribution (SAD) of a local community with user-defined number of species and relative abundance distribution in the pool, and user-defined number of individuals in the simulated local community.

Usage

```
sim_sad(s_pool, n_sim, sad_type = c("lnorm", "bs", "gamma", "geom", "ls",
  "mzsm", "nbinom", "pareto", "poilog", "power", "powbend", "weibull"),
  sad_coef = list(cv_abund = 1), fix_s_sim = FALSE)
```

Arguments

s_pool	Number of species in the pool (integer)
n_sim	Number of individuals in the simulated community (integer)
sad_type	Root name of the species abundance distribution model of the species pool (character) - e.g., "lnorm" for the lognormal distribution (rlnorm); "geom" for the geometric distribution (rgeom), or "ls" for Fisher's log-series distribution (r1s). See the table in Details below, or rsad for all SAD model options.
sad_coef	List with named arguments to be passed to the distribution function defined by the argument sad_type. An overview of parameter names is given in the table below. In mobsim the log-normal and the Poisson log-normal distributions can alternatively be parameterized by the coefficient of variation (cv) of the relative abundances in the species pool. Accordingly, cv_abund is the standard deviation of abundances divided by the mean abundance (no. of individuals / no. of species). cv_abund is thus negatively correlated with the evenness of the species abundance distribution. Please note that the parameters <i>mu</i> and <i>sigma</i> are not equal to the mean and standard deviation of the log-normal distribution.
fix_s_sim	Should the simulation constrain the number of species in the simulated local community? (logical)

Details

The function `sim_sad` was built using code of the function `rsad` from the R package [sads](#). However, in contrast to `rsad`, the function `sim_sad` allows to define the number of individuals in the simulated local community. This is implemented by converting the abundance distribution simulated based on `rsad` into a relative abundance distribution. This relative abundance distribution is considered as the species pool for the local community. In a second step the required no. of individuals (`n_sim`) is sampled (with replacement) from this relative abundance distribution.

Please note that this might effect the interpretation of the parameters of the underlying statistical distribution, e.g. the mean abundance will always be $n_{\text{sim}}/n_{\text{pool}}$ irrespective of the settings of `sad_coef`.

When `fix_s_sim = FALSE` the species number in the local community might deviate from `s_pool` due to stochastic sampling. When `fix_s_sim = TRUE` the local number of species will equal `s_pool`, but this constraint can result in systematic biases from the theoretical distribution parameters. Generally, with `fix_s_sim = TRUE` additional very rare species will be added to the community, while the abundance of the most common ones is reduced to keep the defined number of individuals.

Here is an overview of all available models (`sad_type`) and their respective coefficients (`sad_coef`). Further information is provided by the documentation of the specific functions that can be accessed by the links. Please note that the coefficient `cv_abund` for the log-normal and Poisson log-normal model are only available within `mobsim`.

SAD function	Distribution name	coef #1	coef #2	coef #3
rbs	Mac-Arthur's brokenstick	N	S	
rgamma	Gamma distribution	shape	rate	scale
rgeom	Geometric distribution	prob		
rlnorm	Log-normal distributions	meanlog	sdlog	cv_abund
rls	Fisher's log-series distribution	N	alpha	
rmzsm	Metacommunity zero-sum multinomial	J	theta	
rnbinom	Negative binomial distribution	size	prob	mu
rpareto	Pareto distribution	shape	scale	
rpoilog	Poisson-lognormal distribution	mu	sigma	cv_abund
rpower	Power discrete distributions	s		
rpowbend	Puyeo's Power-bend discrete distribution	s	omega	
rweibull	Weibull distribution	shape	scale	

Value

Object of class `sad`, which contains a named integer vector with species abundances

Author(s)

Felix May

Examples

```
#Simulate log-normal species abundance distribution
sad_lnorm1 <- sim_sad(s_pool = 100, n_sim = 10000, sad_type = "lnorm",
                    sad_coef = list("meanlog" = 5, "sdlog" = 0.5))
plot(sad_lnorm1, method = "octave")
plot(sad_lnorm1, method = "rank")

# Alternative parameterization of the log-normal distribution
sad_lnorm2 <- sim_sad(s_pool = 100, n_sim = 10000, sad_type = "lnorm",
                    sad_coef = list("cv_abund" = 0.5))
plot(sad_lnorm2, method = "octave")
```

```

# Fix species richness in the simulation by adding rare species
sad_lnorm3a <- sim_sad(s_pool = 500, n_sim = 10000, sad_type = "lnorm",
  sad_coef = list("cv_abund" = 5), fix_s_sim = TRUE)
sad_lnorm3b <- sim_sad(s_pool = 500, n_sim = 10000, sad_type = "lnorm",
  sad_coef = list("cv_abund" = 5))

plot(sad_lnorm3a, method = "rank")
points(1:length(sad_lnorm3b), sad_lnorm3b, type = "b", col = 2)
legend("topright", c("fix_s_sim = TRUE", "fix_s_sim = FALSE"),
  col = 1:2, pch = 1)

# Different important SAD models

# Fisher's log-series
sad_logseries <- sim_sad(s_pool = NULL, n_sim = 10000, sad_type = "ls",
  sad_coef = list("N" = 1e5, "alpha" = 20))

# Poisson log-normal
sad_poilog <- sim_sad(s_pool = 100, n_sim = 10000, sad_type = "poilog",
  sad_coef = list("mu" = 5, "sig" = 0.5))

# Mac-Arthur's broken stick
sad_broken_stick <- sim_sad(s_pool = NULL, n_sim = 10000, sad_type = "bs",
  sad_coef = list("N" = 1e5, "S" = 100))

# Plot all SADs together as rank-abundance curves
plot(sad_logseries, method = "rank")
lines(1:length(sad_lnorm2), sad_lnorm2, type = "b", col = 2)
lines(1:length(sad_poilog), sad_poilog, type = "b", col = 3)
lines(1:length(sad_broken_stick), sad_broken_stick, type = "b", col = 4)
legend("topright", c("Log-series", "Log-normal", "Poisson log-normal", "Broken stick"),
  col = 1:4, pch = 1)

```

sim_thomas_community *Simulate community with clumped spatial positions.*

Description

This function simulates a community with a certain abundance distribution and with intraspecific aggregation, i.e. individuals of the same species are distributed in clusters.

Usage

```

sim_thomas_community(s_pool, n_sim, sad_type = "lnorm",
  sad_coef = list(cv_abund = 1), fix_s_sim = FALSE, sigma = 0.02,
  cluster_points = NA, mother_points = NA, xrange = c(0, 1),
  yrange = c(0, 1))

```

Arguments

s_pool	Number of species in the pool (integer)
n_sim	Number of individuals in the simulated community (integer)
sad_type	Root name of the species abundance distribution model of the species pool (character) - e.g., "lnorm" for the lognormal distribution (rlnorm); "geom" for the geometric distribution (rgeom), or "ls" for Fisher's log-series distribution (rls). See the table in Details below, or rsad for all SAD model options.
sad_coef	List with named arguments to be passed to the distribution function defined by the argument sad_type. An overview of parameter names is given in the table below. In <code>mobsim</code> the log-normal and the Poisson log-normal distributions can alternatively be parameterized by the coefficient of variation (<code>cv</code>) of the relative abundances in the species pool. Accordingly, <code>cv_abund</code> is the standard deviation of abundances divided by the mean abundance (no. of individuals / no. of species). <code>cv_abund</code> is thus negatively correlated with the evenness of the species abundance distribution. Please note that the parameters <code>mu</code> and <code>sigma</code> are not equal to the mean and standard deviation of the log-normal distribution.
fix_s_sim	Should the simulation constrain the number of species in the simulated local community? (logical)
sigma	Mean displacement (along each coordinate axes) of a point from its mother point (= cluster centre).
cluster_points	Mean number of points per cluster.
mother_points	Number of mother points (= cluster centres).
xrange	Extent of the community in x-direction (numeric vector of length 2)
yrange	Extent of the community in y-direction (numeric vector of length 2)

Details

This function consecutively calls [sim_sad](#) and [sim_thomas_coords](#)
See the documentations of [sim_sad](#) and [sim_thomas_coords](#) for details.

Value

A community object as defined by [community](#)

Author(s)

Felix May

Examples

```
com1 <- sim_thomas_community(s_pool = 20, n_sim = 500, sad_type = "lnorm",
                           sad_coef = list("meanlog" = 2, "sdlog" = 1),
                           sigma = 0.01)

plot(com1)
```

sim_thomas_coords *Simulate clumped spatial coordinates*

Description

Add clumped (aggregated) positions to a species abundance distribution. Clumping is simulated using a Thomas cluster process, also known as Poisson cluster process (Morlon et al. 2008, Wiegand & Moloney 2014)

Usage

```
sim_thomas_coords(abund_vec, sigma = 0.02, mother_points = NA,
  cluster_points = NA, xrange = c(0, 1), yrange = c(0, 1))
```

Arguments

abund_vec	Species abundance vector (integer)
sigma	Mean displacement (along each coordinate axes) of a point from its mother point (= cluster centre). Sigma correlates with cluster extent. When <code>length(sigma) == length(abund_vec)</code> , each species receives a specific cluster extent. Otherwise, the first value of sigma is recycled and all species share the same cluster extent. When sigma of any species is more than twice as large as the largest plot dimension, a random Poisson distribution is simulated, which is more efficient than a Thomas cluster process. The parameter sigma corresponds to the scale parameter of the function <code>rThomas</code> in the package <code>spatstat</code> .
mother_points	Number of mother points (= cluster centres). If this is a single value, all species have the same number of clusters. For example <code>mother_points = 1</code> can be used to simulate only one cluster per species, which then represents the complete species range. If <code>mother_points</code> is a vector of the same length as <code>abund_vec</code> , each species has a specific number of clusters. If no value is provided, the number of clusters is determined from the abundance and the number of points per cluster (<code>cluster_points</code>).
cluster_points	Mean number of points per cluster. If this is a single value, species have the same average number of points per cluster. If this is a vector of the same length as <code>abund_vec</code> , each species has a specific mean number of points per cluster. If no value is provided, the number of points per cluster is determined from the abundance and from <code>mother_points</code> . The parameter <code>cluster_points</code> corresponds to the mu parameter of <code>spatstat::rThomas</code> .
xrange	Extent of the community in x-direction (numeric vector of length 2)
yrange	Extent of the community in y-direction (numeric vector of length 2)

Details

To generate a Thomas cluster process of a single species this function uses a C++ re-implementation of the function `rThomas` in the package `spatstat`.

There is an inherent link between the parameters `abund_vec`, `mother_points`, and `cluster_points`. For every species the abundance has to be equal to the number of clusters (`mother_points`) times the number of points per cluster (`cluster_points`).

$$abundance = mother_points * cluster_points$$

Accordingly, if one of the parameters is provided, the other one is directly calculated from the abundance. Values for `mother_points` override values for `cluster_points`. If none of the parameters is specified, it is assumed that for every species there is a similar number of clusters and of points per cluster.

$$mother_points = cluster_points = \sqrt{abundance},$$

In this case rare species have few clusters with few points per cluster, while abundant species have many clusters with many points per cluster.

Value

A community object as defined by [community](#).

Author(s)

Felix May

References

Morlon et al. 2008. A general framework for the distance-decay of similarity in ecological communities. *Ecology Letters* 11, 904-917.

Wiegand and Moloney 2014. *Handbook of Spatial Point-Pattern Analysis in Ecology*. CRC Press

See Also

[rThomas](#)

Examples

```
abund <- c(10,20,50,100)
sim1 <- sim_thomas_coords(abund, sigma = 0.02)
plot(sim1)

# Simulate species "ranges"
sim2 <- sim_thomas_coords(abund, sigma = 0.02, mother_points = 1)
plot(sim2)

# Equal numbers of points per cluster
sim3 <- sim_thomas_coords(abund, sigma = 0.02, cluster_points = 5)
plot(sim3)

# With large sigma the distribution will be essentially random (see Details)
```

```
sim4 <- sim_thomas_coords(abund, sigma = 10)
plot(sim4)
```

spec_sample

Sample species richness

Description

Expected species richness in a random sample of fixed size.

Usage

```
spec_sample(abund_vec, n)
```

Arguments

abund_vec	Species abundance distribution of the community (integer vector)
n	Sample size in terms of number of individuals (integer)

Details

The expected number of species is calculated after Hurlbert 1971, Equation 3.
spec_sample is similar to the function [rarefy](#) in the R package [vegan](#).

Value

Expected number of species in a sample of n individuals

References

Hurlbert, S.H. 1971. The nonconcept of species diversity: a critique and + alternative parameters. Ecology 52, 577-586.

Examples

```
sad1 <- sim_sad(100, 1000)
spec_sample(abund_vec = sad1, n = 20)
```

spec_sample_curve *Non-spatial and spatially-explicit species sampling curves*

Description

Expected species richness as function of sample size (no. of individuals), when individuals are sampled randomly (rarefaction) or when nearest-neighbours are samples (accumulation).

Usage

```
spec_sample_curve(comm, method = c("accumulation", "rarefaction"))
```

Arguments

comm	Community object
method	Partial match to accumulation or rarefaction. Also both methods can be included at the same time.

Details

Non-spatial sampling corresponds to the species rarefaction curve, which only depends on the species abundance distribution and can thus be also calculated from abundance data (see [rare_curve](#)).

In contrast the species-accumulation curve starts from a focal individual and only samples the nearest neighbours of the focal individual. The final species accumulation curves is calculated as the mean over the accumulation curves starting from all individuals.

In contrast to the rarefaction curve the accumulation curve is not only influenced by the species abundance distribution, but also by the spatial distribution of individuals.

Value

A dataframe with 2-3 columns. The first column indicates the sample size (numbers of individuals), and the second and third column indicate the expected species richness for spatial sampling (column: "spec_accum") and/or random sampling (column "spec_rarefied")

Examples

```
sim_com1 <- sim_thomas_community(s_pool = 100, n_sim = 1000)
sac1 <- spec_sample_curve(sim_com1, method = c("rare", "acc"))

head(sac1)
plot(sac1)
```

summary.community *Print summary of spatial community object*

Description

Print summary of spatial community object

Usage

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

Arguments

object	Community object of class community
...	Additional arguments passed to print .

summary.sad *Print summary of species abundance distribution object*

Description

Print summary of species abundance distribution object

Usage

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

Arguments

object	Community object of class sad
...	Additional arguments passed to print .

See Also

[sim_sad](#)

Index

abund_rect, 2

barplot, 10

community, 2, 3, 4–7, 14, 15, 19, 21, 24

community_to_sad, 4

dist_decay, 4, 8, 9

div_rand_rect, 6, 6

div_rect, 5, 6, 7

divar, 5, 8, 10

loess, 9

mobsim, 8

mobsim-package (mobsim), 8

plot, 9–11

plot.community, 9

plot.dist_decay, 9

plot.divar, 10

plot.sad, 10

plot.spec_sample_curve, 11

print, 24

rare_curve, 8, 12, 23

rarecurve, 12

rarefy, 22

rbs, 17

rgamma, 17

rgeom, 14, 16, 17, 19

rlnorm, 14, 16, 17, 19

rls, 14, 16, 17, 19

rmzsm, 17

rnbinom, 17

rpareto, 17

rpoilog, 17

rpowbend, 17

rpower, 17

rsad, 14, 16, 19

rThomas, 21

rweibull, 17

sads, 16

sample_quadrats, 8, 13

sim_poisson_community, 8, 14

sim_poisson_coords, 8, 14, 15

sim_sad, 8, 14, 16, 19, 24

sim_thomas_community, 8, 18

sim_thomas_coords, 8, 19, 20

spec_sample, 12, 22

spec_sample_curve, 8, 11, 23

summary.community, 24

summary.sad, 24

vegan, 12, 13, 22

vegdist, 4