Description

Function *simul.comms* allows for the creation of artificial species communities. Users manipulate characteristics of the regional species pool and individual communities via 14 inputs (see arguments below). The function's output includes three elements: the traits matrix for species in the regional species pool [T], a community-by-species abundance matrix [A], and a matrix summarising settings and observed characteristics of the traits and abundances in each artificial community created [S].

Dependencies

Note that simul.comms requires R packages *scales* (Wickham 2016) and *moments* (Komsta & Novomestky 2015), so these must be pre-installed.

Komsta, L. & Novomestky, F. (2015) moments: Moments, cumulants, skewness, kurtosis and related tests. R package version 0.14.

Wickham, H. (2016) scales: Scale functions for graphics. R package version 0.4.1. R.

Usage

```
simul.comms( s = c(5, 10, 15, 20), r = 10, p = 100, t = 3, tr.method = c("unif", "norm:5", "lnorm:1"),
        tr.type = c("cat:5", "ord", "con"), presence = "random", abun.method = "lnorm:1",
        abundance = "common", commin = NULL, commax = NULL, comnot = NULL, comnot.type =
        NULL, dups = 0)
```

Arguments

s	numeric vector listing species richness per community
r	the number of replicate communities to be created per species richness level
p	the number of species in the common species pool
t	the number of traits
tr.method	single character string or character vector of length t (the number of traits), indicating trait sampling distribution(s), where "unif" signifies uniform, "norm" normal, and "lnorm" lognormal. Defaults to "norm". By default, the normal and lognormal distributions have a mean = 0 and a standard deviation = 1 (for the lognormal, these are the mean and standard deviation on the log scale). To specify a different standard deviation, add a colon and specified value to the end of the distribution name, without spaces (e.g. "norm:3" for normal distribution with standard deviation = 3). For the uniform distribution, which by default samples values between 0 and 1, the value after the colon sets the

	maximum of the range in values from which samples are taken (e.g. "unif:2.5" samples values between 0 and 2.5).
tr.type	single character string or character vector of length t (the number of traits), indicating the nature of each trait, where "con" is continuous, "ord" ordinal and "cat" categorical. Defaults to continuous. When set to categorical or ordinal, the number of levels defaults to 10. Adding a colon followed by a number changes the number of levels (e.g. "cat:5" for a categorical trait with 5 levels). Sampling distributions specified in tr.method are preserved (approximately) during conversion to ordinal or categorical.
presence	single character string to indicate whether species in each community should be picked from the species pool at "random", such that "common" species are favoured, based on the joint (multiplied) density functions of each of the species t trait values, or such that "rare" species are favoured, i.e. creating communities with functionally more extreme species. Defaults to "random".
abun.method	single character string indicating the sampling distribution for species abundances. Same options as for tr.method with the addition of option "fixed:1" which sets all abundances equal (to 1 if no value is specified, or the value specified after the colon). Defaults to lognormal. Sampling from each distribution is shifted to positive values where needed, and offset by 0.5 to ensure that species selected for presence have non-zero abundance values.
abundance	single character string indicating whether abundance should be distributed randomly among species in the community ("random") or favour "common" or "rare" species, based on their traits' joint density function. Defaults to "random".
commin	vector of length t (the number of traits) specifying an optional minimum trait value for each trait (on appropriate scale!) for species represented in communities. Defaults to NULL, with no minima enforced for any trait. To set minima for some traits but not others, enter NA where no minimum is required: e.g. c(0.5, NA, 2) would enforce minima for the first and third trait only.
commax	vector of length t (the number of traits) that specifies an optional maximum trait value for each trait (on appropriate scale!) for species represented in communities, analogous to commin. Note that if in combination commin and commax are too restrictive, not enough species may be available for sampling, causing the function to generate an error.
comnot	list with t (the number of traits) elements, each of which should be a vector of trait values to exclude from communities for each trait,

	respectively. If too restrictive (especially in combination with comin and commax), not enough species may be available for sampling, causing the function to generate an error. Values in each element of the list are interpreted as individual values of the corresponding trait that should be excluded, unless the corresponding element of comnot.type (see below) is set to "range". In that case, values instead are interpreted as the start and stop of a continuous interval.
comnot.type	list with t (the number of traits) elements, each consisting of a single character value set either to "single" (the default) or "range". When set to "range", values in the corresponding element of comnot are interpreted as the starting and stopping points of a continuous interval.
dups	single numeric value indicating the number of species per community that should be duplicates of other species also present, i.e. identical in all traits (abundance may differ). Species are chosen for duplication at random, but each species is only duplicated once per community, so dups should not exceed half of the smallest number of species specified in s.

Details

The function was designed to simulate ecological communities in order to test how indices of functional diversity are affected by changes in community composition in terms of species number, number and type of traits, trait distributions and abundance distributions. For such an analysis, the function would typically be run multiple times with some arguments held constant while others change to generate communities with differing trait and abundance characteristics. Each time the function is run, it creates an artificial regional species pool with p species, each bearing t traits whose values are drawn according to the sample distribution(s) specified in tr.method and interpreted as a continuous variable or otherwise converted to ordinal or categorical depending on settings in tr.type. From this regional species pool, the function draws r replicate communities with as many species as specified in s (users can specify multiple values for s in a single function call, yielding r replicate communities for each value of s specified). Species for each community are drawn from the regional species pool either at random or with preference given to species with rare or common traits, depending on settings in argument presence. Species are assigned abundance values based on the sample distribution specified in abun.method, with sample values assigned either randomly or such that they favour either species with rare or common traits, depending on settings in argument abundance. The degree to which trait values in the regional species pool are represented within individual communities can further be manipulated via arguments commin and commax, which allow minimum and maximum values to be set for each trait (e.g. to shorten the represented trait range), or comnot and comnot.type, which allow individual trait values or a range of trait values to be excluded from representation (e.g. to create holes in the represented trait range). Finally, by setting dups>0, users can force communities to include pairs of duplicate species that are perfectly identical to one another in all traits.

The function returns three outputs. The first, T, is a species-by-traits matrix that represents the regional species pool generated by the function call. The second, A, is a site-by species abundance matrix that indicates the artificial communities generated from the regional species pool. The third, S, returns the settings that determine the composition of the artificial communities created, and also provides summary statistics about the resulting traits and abundance distributions observed in each community. The species-by-traits matrix, T, and the site-by-species abundance matrix, A, together provide the input required for computation of most functional diversity indices. The summary statistics in S, instead might help elucidate sensitivities of the functional diversity indices computed. For example, they could be used as explanatory variables in a regression analysis that takes the value of a functional diversity index as the response.

Value

T — a species-by-traits matrix representing the regional species pool. The matrix has t+2 columns and either p rows or, if dups>0, 2*p rows. The first column provides an alphanumeric species identifier. The identifier of original species begins with SP, whereas duplicate species have identifiers starting with DP. Traits columns are labelled T1, T2, T3, etc. and list the value of each trait per species. The final column, LIKELIHOOD, equates to the joint (multiplied) density functions of each of the species' trait values – where the density function of ordinal and categorical traits was averaged per level or category prior to multiplication.

A — a site-by-species abundance matrix specifying the artificial ecological communities generated. The matrix has 1+p columns (or 1+2*p if dups>0) and x*r rows, where x is the number of species richness levels specified in input argument s. The first column provides an alphanumeric site identifier starting with COM for community. The remaining columns list abundance values per species, with values of zero for species not present in the community.

S — a matrix summarising settings and observed characteristics of the traits and abundances in each artificial community created, with x*r+1 rows and the following columns:

- Site — the same alphanumeric site identifier as in A, except that the first row is identified as 'pool' and provides summary statistics about the entire regional species pool rather than individual artificial communities
- SR_pool — p, i.e. the species richness of the regional species pool
- SR_site — the number of species per community, equal to p for the regional species pool and to s for individual communities
- N_dups — the number of duplicate species pairs to appear in artificial communities as set by argument dups
- N_traits — t, the number of traits recorded for each species
- T1_dist — the sample distribution used to determine trait values of the first trait, as specified in tr.method
- T1_sd — the standard deviation of the sample distribution used to determine trait values of the first trait, as specified in tr.method, or, if the sample distribution was

	uniform, the maximum value of the range from which samples were taken
T1_type	the type of trait represented by the first trait (continuous, ordinal or categorical), as specified in tr.type
T1_levels	the number of levels for ordinal or categorical traits as specified in tr.type; note that for continuous traits this is meaningless and will default to 10
T1_setmin	the minimum value of the first trait required for inclusion in artificial communities, as specified using commin
T1_obsmin	the actual observed minimum value of the first trait among species in the community
T1_setmax	the maximum value of the first trait permitted for inclusion in artificial communities, as specified using commax
T1_obsmax	the actual observed maximum value of the first trait among species in the community
T1_range	the observed range of values of the first trait among species in the community, i.e. T1_obsmax minus T1_obsmin.
T1_unique.values	the number of unique values for the first trait among species of the community
T1_var	the variance of values for the first trait among species in the community, calculated using R function var(). Ordinal traits are treated as continuous in the computation, as are categorical traits. For the latter the statistic is meaningless despite being reported.
T1_skewness	the skewness of values for the first trait among species in the community, calculated using R function skewness().Ordinal traits are treated as continuous in the computation, as are categorical traits. For the latter the statistic is meaningless despite being reported.
T1_kurtosis	the kurtosis of values for the first trait among species in the community, calculated using R function kurtosis().Ordinal traits are treated as continuous in the computation, as are categorical traits. For the latter the statistic is meaningless despite being reported.
T2 columns etc.	*equivalent columns for additional traits*
likeli_min	the observed minimum value of the joint density function of trait values among species in the community
likeli_max	the observed maximum value of the joint density function of trait values among species in the community
likeli_range	likeli_max minus likeli_min

likeli_var	the variance of joint density function values among species in the community
likeli_skewness	the skewness of joint density function values among species in the community
likeli_kurtosis	the kurtosis of joint density function values among species in the community
presence	how species were chosen for inclusion in individual communities, as specified in the input argument presence
abundance	how abundance values were distributed among species within artificial communities, as specified in input argument abundance
abun_dist	the sample distribution used to generate abundance values for species in the community, as specified in abun.method
abun_sd	the standard deviation of the sample distribution used to generate abundance values for species in the community, as specified in abun.method, or, if the sample distribution was “fixed” or uniform, the fixed value or maximum value of the range sampled, respectively.
abun_min	the observed minimum non-zero abundance value of species in the community
abun_max	the observed maximum abundance value in the community
abun_var	the variance of abundance values among species in the community
abun_skewness	the skewness of abundance values among species in the community
abun_kurtosis	the kurtosis of abundance values among species in the community
tot_abun	the sum of observed abundance values across species in the community
relabun_min	the minimum non-zero relative abundance value of species in the community
relabun_max	the maximum relative abundance value in the community
relabun_var	the variance of relative abundance values among species in the community
relabun_skewness	the skewness of relative abundance values among species in the community
relabun_kurtosis	the kurtosis of relative abundance values among species in the community

Note

The function does not at this point allow for correlations among different traits.

Authors

Jana M. McPherson: janam@calgaryzoo.com

Lauren A. Yeager: laurenayeager@gmail.com

Julia K. Baum: baum@uvic.ca

Reference

McPherson, JM , Yeager, LA, Baum, JK (under review). A simulation tool to scrutinise the behaviour of functional diversity metrics. Methods in Ecology and Evolution.

Example

```
# Purely to illustrate use of the different types of input, imagine creating a species pool with 15 species
# [p=15]. From that pool, draw communities with either 4 or 5 species [s=c(4,5)], of which 2 are
# duplicates of other species present [dups=2]. Species have three traits [t=3], one of which is a 5-level
# categorical trait, one ordinal with 10 levels, one continuous [tr.type=c("cat:5", "ord", "con")] based on
# a uniform distribution, normal distribution with standard deviation of 5 and standard lognormal
# distribution, respectively [tr.method=c("unif", "norm:5", "lnorm:1")]. Presence of species within
# communities is random [presence="random"] but abundance favours species with common traits
# [abundance="common"], and certain trait values are excluded: species whose categorical trait equals
# 6 or 8 and species whose continuous trait value falls between 0.3 and 0.5 do not occur in the
# simulated communities [comnot=list(c(6,8), NA, c(0.3,0.5)), comnot.type=list("single", "single",
# "range")]. The relevant call in R would look like this:

example<-simul.comms(s=c(4, 5), r=2, p=15, t=3, tr.method=c("unif", "norm:5", "lnorm:1"),
tr.type=c("cat:5", "ord", "con"), presence="random", abun.method="lnorm:1", abundance="common",
comnot=list(c(6,8), NA, c(0.3,0.5)), comnot.type=list("single", "single", "range"), dups=2)

# The call creates an object called ‘example’, which has three elements. These can be printed for viewing
# by calling each in turn:

example$T

example$A

example$S
```

Description

This function is a companion to simul.comms. Its purpose is the same and it works very similarly, except that it expects a species traits matrix produced by simul.comms under input argument tr.pool. This permits user to retain a specific regional species pool as created by simul.comms, and use it to create additional artificial communities with perhaps other traits and abundance characteristics than specified in the original function call.

Dependencies

As for simul.comms

Usage

simul.comms.2(tr.pool = T, r = 10, presence = "random", abun.method = "lnorm:1", abundance = "common", commin = NULL, commax = NULL, comnot = NULL, comnot.type = NULL, dups = 0)

where T is a species-by-traits matrix as generated by a call to function simul.comms

Arguments

tr.pool	a species-by-traits matrix as produced by a call to function simul.comms, including a final column named LIEKLIHOOD that specifies the joint (multiplied) density function of trait values
s	numeric vector listing species richness per community
r	the number of replicate communities to be created per species richness level
presence	single character string to indicate whether species in each community should be picked from the species pool at "random", such that "common" species are favoured, based on the joint (multiplied) density functions of each of the species t trait values, or such that "rare" species are favoured, i.e. creating communities with functionally more extreme species. Defaults to "random".
abun.method	single character string indicating the sampling distribution for species abundances. Same options as for tr.method with the addition of option "fixed:1" which sets all abundances equal (to 1 if no value is specified, or the value specified after the colon). Defaults to lognormal. Sampling from each distribution is shifted to positive values where needed, and offset by 0.5 to ensure that species selected for presence have non-zero abundance values.

abundance	single character string indicating whether abundance should be distributed randomly among species in the community ("random") or favour "common" or "rare" species, based on their traits' joint density function. Defaults to "random".
commin	vector of length t (the number of traits) specifying an optional minimum trait value for each trait (on appropriate scale!) for species represented in communities. Defaults to NULL, with no minima enforced for any trait. To set minima for some traits but not others, enter NA where no minimum is required: e.g. c(0.5, NA, 2) would enforce minima for the first and third trait only.
commax	vector of length t (the number of traits) that specifies an optional maximum trait value for each trait (on appropriate scale!) for species represented in communities, analogous to commin. Note that if in combination commin and commax are too restrictive, not enough species may be available for sampling, causing the function to generate an error.
comnot	list with t (the number of traits) elements, each of which should be a vector of trait values to exclude from communities for each trait, respectively. If too restrictive (especially in combination with comin and commax), not enough species may be available for sampling, causing the function to generate an error. Values in each element of the list are interpreted as individual values of the corresponding trait that should be excluded, unless the corresponding element of comnot.type (see below) is set to "range". In that case, values instead are interpreted as the start and stop of a continuous interval.
comnot.type	list with t (the number of traits) elements, each consisting of a single character value set either to "single" (the default) or "range". When set to "range", values in the corresponding element of comnot are interpreted as the starting and stopping points of a continuous interval.
dups	single numeric value indicating the number of species per community that should be duplicates of other species also present, i.e. identical in all traits (abundance may differ). Species are chosen for duplication at random, but each species is only duplicated once per community, so dups should not exceed half of the smallest number of species specified in s.

Details

Same as for simul.comms

Value

Same as for simul.comms

Authors

Jana M. McPherson: janam@calgaryzoo.com

Lauren A. Yeager: laurenayeager@gmail.com

Julia K. Baum: baum@uvic.ca

Reference

McPherson, JM , Yeager, LA, Baum, JK (under review). A simulation tool to scrutinise the behaviour of functional diversity metrics. Methods in Ecology and Evolution.

Example

example<-simul.comms(s=c(4, 5), r=2, p=15, t=3, tr.method=c("unif", "norm:5", "lnorm:1"), tr.type=c("cat:5", "ord", "con"), presence="random", abun.method="lnorm:1", abundance="common", comnot=list(c(6,8), NA, c(0.3,0.5)), comnot.type=list("single", "single", "range"), dups=2)

test<-simul.comms.2(tr.pool=example$T, s=c(4,5), r=2, presence="common", abun.method="unif:1", abundance="random", comnot=NULL, comnot.type=NULL, dups=0)