Package 'BAT'

January 20, 2025

Title Biodiversity Assessment Tools

Version 2.9.6

Date 2024-02-16

Description Includes algorithms to assess alpha and beta diversity in all their dimensions (taxonomic, phylogenetic and functional). It allows performing a number of analyses based on species identities/abundances, phylogenetic/functional distances, trees, convex-hulls or kernel density n-dimensional hypervolumes depicting species relationships.
Cardoso et al. (2015) <doi:10.1111/2041-210X.12310>.

Depends R (>= 3.0.0)

Imports ape, geometry, graphics, hypervolume, MASS, methods, nls2, parallel, phytools, stats, terra, utils, vegan

Encoding UTF-8

License GPL-3

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Pedro Cardoso [aut, cre],
Stefano Mammola [aut],
Francois Rigal [aut],
Jose Carvalho [aut]

Maintainer Pedro Cardoso <pmcardoso@fc.ul.pt>

Repository CRAN

Date/Publication 2024-02-17 21:50:02 UTC

Contents

accuracy *Scaled mean squared error of accumulation curves.*

Description

Accuracy (scaled mean squared error) of accumulation curves compared with a known true diversity value (target).

Usage

```
accuracy(accum, target = -1)
```

Arguments

`accum`	A matrix resulting from the alpha.accum or beta.accum functions (sampling units x diversity values).
`target`	The true known diversity value, with which the curve will be compared. If not specified, default is the diversity observed with all sampling units.

Details

Among multiple measures of accuracy (Walther & Moore 2005) the SMSE presents several advantages, as it is (Cardoso et al. 2014): (i) scaled to true diversity, so that similar absolute differences are weighted according to how much they represent of the real value; (ii) scaled to the number of sampling units, so that values are independent of sample size; (iii) squared, so that small, mostly meaningless fluctuations around the true value are down-weighted; and (iv) independent of positive or negative deviation from the real value, as such differentiation is usually not necessary. For alpha diversity accuracy may also be weighted according to how good the data is predicted to be. The weight of each point in the curve is proportional to its sampling intensity (i.e. n/Sobs).

Value

Accuracy values (both raw and weighted) for all observed and estimated curves.

References

Cardoso, P., Rigal, F., Borges, P.A.V. & Carvalho, J.C. (2014) A new frontier in biodiversity inventory: a proposal for estimators of phylogenetic and functional diversity. Methods in Ecology and Evolution, 5: 452-461.

Walther, B.A. & Moore, J.L. (2005) The concepts of bias, precision and accuracy, and their use in testing the performance of species richness estimators, with a literature reviewof estimator performance. Ecography, 28, 815-829.

Examples

```
comm1 <- matrix(c(2,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,0,2,2), nrow = 4, ncol = 5, byrow = TRUE)
comm2 <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
acc.alpha = alpha.accum(comm1)
accuracy(acc.alpha)
accuracy(acc.alpha, 10)
acc.beta = beta.accum(comm1, comm2, tree)
accuracy(acc.beta)
accuracy(acc.beta, c(1,1,0))
```

---

aic *Akaike Information Criterion.*

Description

Calculates the Akaike Information Criterion (AIC) of any model based on observed and estimated values.

Usage

```
aic(obs, est = NULL, param = 0, correct = FALSE)
```

Arguments

`obs` Either a model or a vector with observed values.

`est` A vector with estimated values. Only used if obs is not a model.

`param` Number of parameters in the model. If obs is a model param will be ignored and the number of parameters will be calculated from the model.

`correct` Boolean indicating whether the corrected version of AIC (AICc) should be calculated, mostly for models with few observations.

Details

Useful for models or functions that do not provide logLik values.

Value

The AIC or AICc value.

Examples

```
obs = c(1,4,5,6)
est = c(0,1,4,7)

#example using values
aic(obs, est)
aic(obs, est, param = 1)
aic(obs, est, param = 1, correct = TRUE)

#example using model
mod = lm(obs ~ est)
aic(mod)
extractAIC(mod)[2]
aic(mod, correct = TRUE)
```

`alpha` *Alpha diversity (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Observed richness with possible rarefaction, multiple sites simultaneously.

Usage

```
alpha(comm, tree, raref = 0, runs = 100)
```

Arguments

`comm` A sites x species matrix, with either abundance or incidence data.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

`raref` An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.

`runs` Number of resampling runs for rarefaction. If not specified, default is 100.

Details

TD is equivalent to species richness. Calculations of PD and FD are based on Faith (1992) and Petchey & Gaston (2002, 2006), which measure PD and FD of a community as the total branch length of a tree linking all species represented in such community. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree. The rarefaction option is useful to compare communities with much different numbers of individuals sampled, which might bias diversity comparisons (Gotelli & Colwell 2001)

Value

A matrix of sites x diversity values (either "Richness" OR "Mean, Median, Min, LowerCL, UpperCL and Max").

References

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation, 61, 1-10.

Gotelli, N.J. & Colwell, R.K. (2001) Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. Ecology Letters, 4, 379-391.

Petchey, O.L. & Gaston, K.J. (2002) Functional diversity (FD), species richness and community composition. Ecology Letters, 5, 402-411.

Petchey, O.L. & Gaston, K.J. (2006) Functional diversity: back to basics and looking forward. Ecology Letters, 9, 741-758.

Examples

```
comm <- matrix(c(0,0,1,1,0,0,2,1,0,0), nrow = 2, ncol = 5, byrow = TRUE)
trait = 1:5
tree <- tree.build(trait)
plot(tree, "u")
alpha(comm)
alpha(comm, raref = 0)
alpha(comm, tree)
alpha(comm, tree, 2, 100)
```

alpha.accum *Alpha diversity accumulation curves (observed and estimated).*

Description

Estimation of alpha diversity of a single site with accumulation of sampling units.

Usage

```
alpha.accum(
  comm,
  tree,
  func = "nonparametric",
  target = -2,
  runs = 100,
  prog = TRUE
)
```

Arguments

`comm`	A sampling units x species matrix, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`func`	The class of estimators to be used: If func is partial match of "curve", TD, PD or FD are based on extrapolating the accumulation curve of observed diversity. If func is partial match of "nonparametric", TD, PD or FD are based on non-parametric estimators. If func is partial match of "completeness", PD or FD estimates are based on the completeness of TD (requires a tree to be used). If not specified, default is "nonparametric.
`target`	True diversity value to calculate the accuracy of curves (scaled mean squared error). If not specified do not calculate accuracy (default), -1 uses the total observed diversity as true diversity and any other value is the true known diversity.
`runs`	Number of random permutations to be made to the sampling order. If not specified, default is 100.
`prog`	Present a text progress bar in the R console.

Details

Observed diversity often is an underestimation of true diversity. Several approaches have been devised to estimate species richness (TD) from incomplete sampling. These include: (1) fitting asymptotic functions to randomised accumulation curves (Soberon & Llorente 1993; Flather 1996; Cardoso et al. in prep.) (2) the use of non-parametric estimators based on the incidence or abundance of rare species (Heltshe & Forrester 1983; Chao 1984, 1987; Colwell & Coddington 1994). A correction to non-parametric estimators has also been recently proposed, based on the proportion of singleton or unique species (species represented by a single individual or in a single sampling unit respectively; Lopez et al. 2012). Cardoso et al. (2014) have proposed a way of adapting these approaches to estimate PD and FD, also adding a third possible approach for these dimensions of diversity: (3) correct PD and FD values based on the completeness of TD, where completeness equals the proportion of estimated true diversity that was observed. Calculations of PD and FD are based on Faith (1992) and Petchey & Gaston (2002, 2006), which measure PD and FD of a community as the total branch length of a tree linking all species represented in such community. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree.

Value

A matrix of sampling units x diversity values (sampling units, individuals, observed and estimated diversity). The values provided by this function are:

Sampl - Number of sampling units;

Ind - Number of individuals;

Obs - Observed diversity;

S1 - Singletons;

S2 - Doubletons;

Q1 - Uniques;

Q2 - Duplicates;

Jack1ab - First order jackknife estimator for abundance data;

Jack1in - First order jackknife estimator for incidence data;

Jack2ab - Second order jackknife estimator for abundance data;

Jack2in - Second order jackknife estimator for incidence data;

Chao1 - Chao estimator for abundance data;

Chao2 - Chao estimator for incidence data;

Clench - Clench or Michaelis-Menten curve;

Exponential - Exponential curve;

Rational - Rational function;

Weibull - Weibull curve;

The P-corrected version of all non-parametric estimators is also provided.

Accuracy - if accuracy is to be calculated a list is returned instead, with the second element being the scaled mean squared error of each estimator.

References

Cardoso, P., Rigal, F., Borges, P.A.V. & Carvalho, J.C. (2014) A new frontier in biodiversity inventory: a proposal for estimators of phylogenetic and functional diversity. Methods in Ecology and Evolution, 5: 452-461.

Chao, A. (1984) Nonparametric estimation of the number of classes in a population. Scandinavian Journal of Statistics, 11, 265-270.

Chao, A. (1987) Estimating the population size for capture-recapture data with unequal catchability. Biometrics 43, 783-791.

Colwell, R.K. & Coddington, J.A. (1994) Estimating terrestrial biodiversity through extrapolation. Phil. Trans. Roy. Soc. London B 345, 101-118.

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation, 61, 1-10.

Flather, C. (1996) Fitting species-accumulation functions and assessing regional land use impacts on avian diversity. Journal of Biogeography, 23, 155-168.

Heltshe, J. & Forrester, N.E. (1983) Estimating species richness using the jackknife procedure. Biometrics, 39, 1-11.

Lopez, L.C.S., Fracasso, M.P.A., Mesquita, D.O., Palma, A.R.T. & Riul, P. (2012) The relationship between percentage of singletons and sampling effort: a new approach to reduce the bias of richness estimates. Ecological Indicators, 14, 164-169.

Petchey, O.L. & Gaston, K.J. (2002) Functional diversity (FD), species richness and community composition. Ecology Letters, 5, 402-411.

Petchey, O.L. & Gaston, K.J. (2006) Functional diversity: back to basics and looking forward. Ecology Letters, 9, 741-758.

Soberon, M.J. & Llorente, J. (1993) The use of species accumulation functions for the prediction of species richness. Conservation Biology, 7, 480-488.

Examples

```
comm <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
alpha.accum(comm)
alpha.accum(comm, func = "nonparametric")
alpha.accum(comm, tree, "completeness")
alpha.accum(comm, tree, "curve", runs = 1000)
alpha.accum(comm, target = -1)
```

`alpha.estimate` *Alpha diversity estimates.*

Description

Estimation of alpha diversity of multiple sites simultaneously.

Usage

```
alpha.estimate(comm, tree, func = "nonparametric")
```

Arguments

`comm` A sites x species matrix, with either abundances or number of incidences.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

`func` The class of estimators to be used: If func is partial match of "nonparametric", TD, PD or FD are based on non-parametric estimators. If func is partial match of "completeness", PD or FD estimates are based on the completeness of TD (requires a tree to be used). If not specified, default is "nonparametric".

Details

Observed diversity often is an underestimation of true diversity. Non-parametric estimators based on the incidence or abundance of rare species have been proposed to overcome the problem of undersampling (Heltshe & Forrester 1983; Chao 1984, 1987; Colwell & Coddington 1994). A correction to non-parametric estimators has also been recently proposed, based on the proportion (P) of singleton or unique species (species represented by a single individual or in a single sampling unit respectively; Lopez et al. 2012). Cardoso et al. (2014) have proposed a way of adapting non-parametric species richness estimators to PD and FD. They have also proposed correcting PD and FD values based on the completeness of TD, where completeness equals the proportion of estimated true diversity that was observed. Calculations of PD and FD are based on Faith (1992) and Petchey & Gaston (2002, 2006), which measure PD and FD of a community as the total branch length of a tree linking all species represented in such community. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree.

Value

A matrix of sites x diversity values (individuals, observed and estimated diversity). The values provided by this function are:

Ind - Number of individuals;

Obs - Observed diversity;

S1 - Singletons;

S2 - Doubletons;

Jack1ab - First order jackknife estimator for abundance data;

Jack2ab - Second order jackknife estimator for abundance data;

Chao1 - Chao estimator for abundance data.

The P-corrected version of all estimators is also provided.

References

Cardoso, P., Rigal, F., Borges, P.A.V. & Carvalho, J.C. (2014) A new frontier in biodiversity inventory: a proposal for estimators of phylogenetic and functional diversity. Methods in Ecology and Evolution, 5: 452-461.

Chao, A. (1984) Nonparametric estimation of the number of classes in a population. Scandinavian Journal of Statistics, 11, 265-270.

Chao, A. (1987) Estimating the population size for capture-recapture data with unequal catchability. Biometrics 43, 783-791.

Colwell, R.K. & Coddington, J.A. (1994) Estimating terrestrial biodiversity through extrapolation. Phil. Trans. Roy. Soc. London B 345, 101-118.

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation, 61, 1-10.

Heltshe, J. & Forrester, N.E. (1983) Estimating species richness using the jackknife procedure. Biometrics, 39, 1-11.

Lopez, L.C.S., Fracasso, M.P.A., Mesquita, D.O., Palma, A.R.T. & Riul, P. (2012) The relationship between percentage of singletons and sampling effort: a new approach to reduce the bias of richness estimates. Ecological Indicators, 14, 164-169.

Petchey, O.L. & Gaston, K.J. (2002) Functional diversity (FD), species richness and community composition. Ecology Letters, 5, 402-411.

Petchey, O.L. & Gaston, K.J. (2006) Functional diversity: back to basics and looking forward. Ecology Letters, 9, 741-758.

Examples

```
comm <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
alpha.estimate(comm)
alpha.estimate(comm, tree)
alpha.estimate(comm, tree, func = "completeness")
```

arrabida *Sample data of spiders in Arrabida (Portugal)*

Description

A dataset containing the abundance of 338 spider species in each of 320 sampling units. Details are described in: Cardoso, P., Gaspar, C., Pereira, L.C., Silva, I., Henriques, S.S., Silva, R.R. & Sousa, P. (2008) Assessing spider species richness and composition in Mediterranean cork oak forests. Acta Oecologica, 33: 114-127.

Usage

```
data(arrabida)
```

Format

A data frame with 320 sampling units (rows) and 338 species (variables).

beta — *Beta diversity (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Beta diversity with possible rarefaction, multiple sites simultaneously.

Usage

```
beta(
  comm,
  tree,
  func = "jaccard",
  abund = TRUE,
  raref = 0,
  runs = 100,
  comp = FALSE
)
```

Arguments

`comm`	A sites x species matrix, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`func`	Partial match indicating whether the Jaccard or Soerensen family of beta diversity measures should be used. If not specified, default is Jaccard.
`abund`	A boolean (T/F) indicating whether abundance data should be used or converted to incidence before analysis.
`raref`	An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.
`runs`	Number of resampling runs for rarefaction. If not specified, default is 100.
`comp`	Boolean indicating whether beta diversity components (shared and unique fractions) should be returned.

Details

The beta diversity measures used here follow the partitioning framework independently developed by Podani & Schmera (2011) and Carvalho et al. (2012) and later expanded to PD and FD by Cardoso et al. (2014), where Btotal = Brepl + Brich. Btotal = total beta diversity, reflecting both species replacement and loss/gain; Brepl = beta diversity explained by replacement of species alone; Brich = beta diversity explained by species loss/gain (richness differences) alone. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree. The rarefaction option is useful to compare communities with much different numbers of individuals sampled, which might bias diversity comparisons (Gotelli & Colwell 2001).

Value

Three distance matrices between sites, one per each of the three beta diversity measures (either "Obs" OR "Mean, Median, Min, LowerCL, UpperCL and Max").

References

Cardoso, P., Rigal, F., Carvalho, J.C., Fortelius, M., Borges, P.A.V., Podani, J. & Schmera, D. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41, 749-761.

Carvalho, J.C., Cardoso, P. & Gomes, P. (2012) Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. Global Ecology and Biogeography, 21, 760-771.

Gotelli, N.J. & Colwell, R.K. (2001) Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. Ecology Letters, 4, 379-391.

Podani, J. & Schmera, D. (2011) A new conceptual and methodological framework for exploring and explaining pattern in presence-absence data. Oikos, 120, 1625-1638.

Examples

```
comm <- matrix(c(2,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,1,2,2), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
beta(comm)
beta(comm, abund = FALSE, comp = TRUE)
beta(comm, tree)
beta(comm, raref = 1)
beta(comm, tree, "s", abund = FALSE, raref = 2)
```

beta.accum *Beta diversity accumulation curves.*

Description

Beta diversity between two sites with accumulation of sampling units.

Usage

```
beta.accum(
  comm1,
  comm2,
  tree,
  func = "jaccard",
  abund = TRUE,
  runs = 100,
  prog = TRUE
)
```

Arguments

`comm1`	A sampling units x species matrix for the first site, with either abundance or incidence data.
`comm2`	A sampling units x species matrix for the second site, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`func`	Partial match indicating whether the Jaccard or Soerensen family of beta diversity measures should be used. If not specified, default is jaccard.
`abund`	A boolean (T/F) indicating whether abundance data should be used or converted to incidence before analysis.
`runs`	Number of random permutations to be made to the sampling order. If not specified, default is 100.
`prog`	Present a text progress bar in the R console.

Details

As widely recognized for species richness, beta diversity is also biased when communities are undersampled. Beta diversity accumulation curves have been proposed by Cardoso et al. (2009) to test if beta diversity has approached an asymptote when comparing two undersampled sites. The beta diversity measures used here follow the partitioning framework independently developed by Podani & Schmera (2011) and Carvalho et al. (2012) and later expanded to PD and FD by Cardoso et al. (2014), where Btotal = Brepl + Brich. Btotal = total beta diversity, reflecting both species replacement and loss/gain; Brepl = beta diversity explained by replacement of species alone; Brich = beta diversity explained by species loss/gain (richness differences) alone. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm1 and comm2 must be the same as in tree. Also, the number of sampling units should be similar in both sites.

Value

Three matrices of sampling units x diversity values, one per each of the three beta diversity measures (sampling units, individuals and observed diversity).

References

Cardoso, P., Borges, P.A.V. & Veech, J.A. (2009) Testing the performance of beta diversity measures based on incidence data: the robustness to undersampling. Diversity and Distributions, 15, 1081-1090.

Cardoso, P., Rigal, F., Carvalho, J.C., Fortelius, M., Borges, P.A.V., Podani, J. & Schmera, D. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41, 749-761.

Carvalho, J.C., Cardoso, P. & Gomes, P. (2012) Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. Global Ecology and Biogeography, 21, 760-771.

Podani, J. & Schmera, D. (2011) A new conceptual and methodological framework for exploring and explaining pattern in presence-absence data. Oikos, 120, 1625-1638.

Examples

```
comm1 <- matrix(c(2,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,0,2,2), nrow = 4, byrow = TRUE)
comm2 <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 4, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
beta.accum(comm1, comm2)
beta.accum(comm1, comm2, func = "Soerensen")
beta.accum(comm1, comm2, tree)
beta.accum(comm1, comm2, abund = FALSE)
beta.accum(comm1, comm2, tree,, FALSE)
```

beta.evenness *Beta diversity evenness (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Difference of evenness between pairs of sites.

Usage

```
beta.evenness(
  comm,
  tree,
  distance,
  method = "expected",
  func = "camargo",
  abund = TRUE
)
```

Arguments

comm	A sites x species matrix, with either abundance or incidence data.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
distance	A dist or matrix object representing the phylogenetic or functional distance between species. If both tree and distance are missing, taxonomic evenness is calculated.

method	Calculate evenness using "expected" values (default) or values based on "contribution" of species to the tree.
func	Calculate evenness using "Camargo" (default) or "Bulla" index.
abund	A boolean (T/F) indicating whether evenness should be calculated using abundance data.

Details

This measure is simply the pairwise difference of evenness calculated based on the index of Camargo (1993) or Bulla (1994) using the values of both species abundances and edge lengths in the tree (if PD/FD).

If no tree or distance is provided the result is the original index.

Value

Distance matrix between sites.

References

Bulla, L. (1994) An index of evenness and its associated diversity measure. Oikos, 70: 167-171.

Camargo, J.A. (1993) Must dominance increase with the number of subordinate species in competitive interactions? Journal of Theoretical Biology, 161: 537-542.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,1,1,1,1,100), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method = "euclidean")
tree <- hclust(distance, method = "average")
beta.evenness(comm)
beta.evenness(comm, tree)
beta.evenness(comm, tree, method = "contribution")
beta.evenness(comm, tree, abund = FALSE)
```

beta.multi *Beta diversity among multiple communities.*

Description

Beta diversity with possible rarefaction - multiple sites measure calculated as the average or variance of all pairwise values.

Usage

```
beta.multi(comm, tree, func = "jaccard", abund = TRUE, raref = 0, runs = 100)
```

Arguments

`comm` A sites x species matrix, with either abundance or incidence data.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

`func` Indicates whether the Jaccard or Soerensen family of beta diversity measures should be used. If not specified, default is jaccard.

`abund` A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis.

`raref` An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.

`runs` Number of resampling runs for rarefaction. If not specified, default is 100.

Details

Beta diversity of multiple sites simultaneously is calculated as either the average or the variance among all pairwise comparisons (Legendre, 2014). The beta diversity measures used here follow the partitioning framework independently developed by Podani & Schmera (2011) and Carvalho et al. (2012) and later expanded to PD and FD by Cardoso et al. (2014), where Btotal = Brepl + Brich. Btotal = total beta diversity, reflecting both species replacement and loss/gain; Brepl = beta diversity explained by replacement of species alone; Brich = beta diversity explained by species loss/gain (richness differences) alone. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree.

Value

A matrix of beta measures x diversity values (average and variance).

References

Cardoso, P., Rigal, F., Carvalho, J.C., Fortelius, M., Borges, P.A.V., Podani, J. & Schmera, D. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41, 749-761.

Carvalho, J.C., Cardoso, P. & Gomes, P. (2012) Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. Global Ecology and Biogeography, 21, 760-771.

Legendre, P. (2014) Interpreting the replacement and richness difference components of beta diversity. Global Ecology and Biogeography, 23: 1324-1334.

Podani, J. & Schmera, D. (2011) A new conceptual and methodological framework for exploring and explaining pattern in presence-absence data. Oikos, 120, 1625-1638.

Examples

```
comm <- matrix(c(2,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,0,2,2), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
beta.multi(comm)
beta.multi(comm, func = "Soerensen")
beta.multi(comm, tree)
beta.multi(comm, raref = 1)
beta.multi(comm, tree, "s", FALSE, raref = 2)
```

---

contribution *Contribution of species or individuals to total phylogenetic/functional diversity.*

Description

Contribution of each species or individual to the total PD or FD of a number of communities.

Usage

```
contribution(comm, tree, abund = FALSE, relative = FALSE)
```

Arguments

`comm` A sites x species matrix, with either abundance or incidence data. If missing, the contribution of all species to the full tree is calculated.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

`abund` A boolean (T/F) indicating whether contribution should be weighted by abundance of each species.

`relative` A boolean (T/F) indicating whether contribution should be relative to total PD or FD (proportional contribution per individual or species). If FALSE, the sum of contributions for each site is equal to total PD/FD, if TRUE it is 1.

Details

Contribution is equivalent to the evolutionary distinctiveness index (ED) of Isaac et al. (2007) if done by species and to the abundance weighted evolutionary distinctiveness (AED) of Cadotte et al. (2010) if done by individual.

Value

A matrix of sites x species values (or values per species if no comm is given).

References

Isaac, N.J.B., Turvey, S.T., Collen, B., Waterman, C. & Baillie, J.E.M. (2007) Mammals on the EDGE: conservation priorities based on threat and phylogeny. PLoS One, 2: e296.

Cadotte, M.W., Davies, T.J., Regetz, J., Kembel, S.W., Cleland, E. & Oakley, T.H. (2010) Phylogenetic diversity metrics for ecological communities: integrating species richness, abundance and evolutionary history. Ecology Letters, 13: 96-105.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,1,0,1), nrow = 4, byrow = TRUE)
tree = tree.build(1:5)

contribution(comm, tree)
contribution(comm, tree, TRUE)
contribution(comm, tree, relative = TRUE)
```

`coverage` *Coverage of datasets.*

Description

Coverage is a measure of completeness of a dataset.

Usage

```
coverage(comm, tree)
```

Arguments

`comm` A matrix of sites x species with abundance values.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

Details

Calculated as the estimated proportion of individuals that belong to the species (or phylogenetic, or functional diversity) already collected (Chao and Jost 2012).

Value

A vector with coverage values per site.

References

Chao, A. & Jost, L. (2012). Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size. Ecology, 93: 2533-2547.

Examples

```
comm <- matrix(c(2,1,0,0,100,1,2,0,0,3,1,2,4,0,0,0,0,0,2,2), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
coverage(comm)
coverage(comm, tree)
```

---

cwd *Community Weighted Dispersion.*

Description

Standard deviation value of each of a series of traits in multiple communities.

Usage

```
cwd(comm, trait, abund = TRUE, na.rm = FALSE)
```

Arguments

comm	A sites x species matrix, with incidence or abundance data about the species in the community.
trait	A species x traits matrix, with trait values for each species in comm.
abund	A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis. If not specified, default is TRUE.
na.rm	Remove NA values before calculating cwd.

Details

Community weighted dispersion is used to compare communities in terms of their dispersion of trait values around a mean, reflecting individual trait variability or diversity.

Value

A sites x trait matrix with sd value per site and trait.

Examples

```
comm <- matrix(c(2,5,0,0,0,1,1,0,0,0,0,1,2,0,0,0,0,0,10,1), nrow = 4, ncol = 5, byrow = TRUE)
rownames(comm) = c("Site1","Site2","Site3","Site4")
colnames(comm) = c("Sp1","Sp2","Sp3","Sp4","Sp5")
trait <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 5, ncol = 4, byrow = TRUE)
rownames(trait) = colnames(comm)
colnames(trait) = c("Trait1","Trait2","Trait3","Trait4")
cwd(comm, trait)
cwd(comm, trait, FALSE)
```

cwe *Community Weighted Evenness.*

Description

Evenness value of each of a series of traits in multiple communities.

Usage

```
cwe(comm, trait, func = "camargo", abund = TRUE, na.rm = FALSE)
```

Arguments

`comm` A sites x species matrix, with incidence or abundance data about the species in the community.

`trait` A species x traits matrix, with trait values for each species in comm.

`func` Calculate evenness using Camargo (1993; default) or Bulla (1994) index.

`abund` A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis. If not specified, default is TRUE.

`na.rm` Remove NA values before calculating cwe.

Details

Community weighted evenness is used to compare communities in terms of their evenness of trait values, reflecting trait abundance and distances between values.

Value

A sites x trait matrix with evenness value per site and trait.

References

Bulla, L. (1994) An index of evenness and its associated diversity measure. Oikos, 70: 167-171.

Camargo, J.A. (1993) Must dominance increase with the number of subordinate species in competitive interactions? Journal of Theoretical Biology, 161: 537-542.

Examples

```
comm <- matrix(c(1,1,1,1,0,1,1,0,0,0,0,1,2,0,0,0,0,0,10,1), nrow = 4, ncol = 5, byrow = TRUE)
rownames(comm) = c("Site1","Site2","Site3","Site4")
colnames(comm) = c("Sp1","Sp2","Sp3","Sp4","Sp5")
trait <- matrix(c(4,1,3,4,2,2,2,1,3,3,2,0,1,4,0,0,5,5,2,1), nrow = 5, ncol = 4, byrow = TRUE)
rownames(trait) = colnames(comm)
colnames(trait) = c("Trait1","Trait2","Trait3","Trait4")
cwe(comm, trait)
cwe(comm, trait, abund = FALSE)
cwe(comm, trait, "bulla")
```

cwm *Community Weighted Mean.*

Description

Average value of each of a series of traits in multiple communities.

Usage

```
cwm(comm, trait, abund = TRUE, na.rm = FALSE)
```

Arguments

`comm` A sites x species matrix, with incidence or abundance data about the species in the community.

`trait` A species x traits matrix, with trait values for each species in comm.

`abund` A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis. If not specified, default is TRUE.

`na.rm` Remove NA values before calculating cwm.

Details

Community weighted mean is used to compare communities in terms of their "typical" trait values.

Value

A sites x trait matrix with mean value per site and trait.

Examples

```
comm <- matrix(c(2,5,0,0,0,1,1,0,0,0,0,1,2,0,0,0,0,0,10,1), nrow = 4, ncol = 5, byrow = TRUE)
rownames(comm) = c(”Site1”,”Site2”,”Site3”,”Site4”)
colnames(comm) = c(”Sp1”,”Sp2”,”Sp3”,”Sp4”,”Sp5”)
trait <- data.frame(Trait1 = c(1,0,0,2,0), Trait2 = c(rep(”A”,2), rep(”B”,3)))
rownames(trait) = colnames(comm)
cwm(comm, trait)
cwm(comm, trait, FALSE)
```

`dispersion` *Phylogenetic/functional dispersion of species or individuals.*

Description

Average dissimilarity between any two species or individuals randomly chosen in a community.

Usage

```
dispersion(
  comm,
  tree,
  distance,
  func = "originality",
  abund = TRUE,
  relative = TRUE
)
```

Arguments

`comm`	A sites x species matrix, with either abundance or incidence data. If missing, the dispersion using the full tree or distance matrix is calculated.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`distance`	A dist object representing the phylogenetic or functional distance between species.
`func`	Calculate dispersion using originality (default), uniqueness or contribution.
`abund`	A boolean (T/F) indicating whether dispersion should be calculated using individuals (T) or species (F).
`relative`	A boolean (T/F) indicating whether dispersion should be relative to the maximum distance between any two species in the tree or distance matrix.

Details

If abundance data is used and a tree is given, dispersion is the quadratic entropy of Rao (1982). If abundance data is not used but a tree is given, dispersion is the phylogenetic dispersion measure of Webb et al. (2002).

Value

A vector of values per site (or a single value if no comm is given).

References

Rao, C.R. (1982) Diversity and dissimilarity coefficients: a unified approach. Theoretical Population Biology, 21: 24-43.

Webb, C.O., Ackerly, D.D., McPeek, M.A. & Donoghue, M.J. (2002) Phylogenies and community ecology. Annual Review of Ecology and Systematics, 33: 475-505.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,1,1,1), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method="euclidean")
tree <- hclust(distance, method="average")
dispersion(tree = tree)
dispersion(distance = distance)
dispersion(comm, tree)
dispersion(comm, tree, abund = FALSE)
dispersion(comm, tree, abund = FALSE, relative = FALSE)
```

`dummy` *Dummify variables.*

Description

Convert factor variables to dummy variables.

Usage

```
dummy(trait, convert = NULL, weight = FALSE)
```

Arguments

`trait` A species x traits matrix or data.frame.

`convert` A vector of column numbers, usually categorical variables, to be converted to dummy variables.

`weight` Indicates whether weights of variables should be returned (TRUE/FALSE) or a vector with weights per variable.

Details

If convert is given the algorithm will convert these column numbers to dummy variables. Otherwise it will convert all columns with factors or characters.

Value

A matrix with variables converted or, if weight == TRUE or a vector, a list also with weights.

Examples

```
trait = data.frame(length = c(2,4,6,3,1), wing = c("A", "B", "A", "A", "B"))
dummy(trait)
dummy(trait, weight = TRUE)
dummy(trait, convert = 2, weight = c(0.9, 0.1))
```

evenness *Taxonomic/phylogenetic/functional evenness of species or individuals.*

Description

Regularity of abundances and distances (if PD/FD) between species in a community.

Usage

```
evenness(
  comm,
  tree,
  distance,
  method = "expected",
  func = "camargo",
  abund = TRUE
)
```

Arguments

`comm`	A sites x species matrix, with either abundance or incidence data. If missing, the evenness using the full tree or distance matrix is calculated.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`distance`	A dist or matrix object representing the phylogenetic or functional distance between species. If both tree and distance are missing, taxonomic evenness is calculated.
`method`	Calculate evenness using "expected" values (default) or values based on "contribution" of species to the tree.
`func`	Calculate evenness using "Camargo" (default) or "Bulla" index.
`abund`	A boolean (T/F) indicating whether evenness should be calculated using abundance data.

Details

Evenness is calculated based on the index of Camargo (1993) or Bulla (1994) using the values of both species abundances and edge lengths in the tree (if PD/FD).

If no tree or distance is provided the result is the original index.

If any site has < 2 species its value will be NA.

Value

A vector of values per site (or a single value if no comm is given).

References

Bulla, L. (1994) An index of evenness and its associated diversity measure. Oikos, 70: 167-171.

Camargo, J.A. (1993) Must dominance increase with the number of subordinate species in competitive interactions? Journal of Theoretical Biology, 161: 537-542.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,1,1,1,1,100), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method = "euclidean")
tree <- hclust(distance, method = "average")
evenness(comm)
evenness(tree = tree, func = "bulla")
evenness(comm, tree)
evenness(comm, tree, method = "contribution")
evenness(comm, tree, abund = FALSE)
```

---

evenness.contribution *Contribution of each species or individual to the total taxonomic/phylogenetic/functional evenness.*

Description

Contribution of each observation to the regularity of abundances and distances (if PD/FD) between species in a community (or individuals in a species).

Usage

```
evenness.contribution(
  comm,
  tree,
  distance,
  method = "expected",
  func = "camargo",
  abund = TRUE
)
```

Arguments

comm	A sites x species matrix, with either abundance or incidence data. If missing, the evenness using the full tree or distance matrix is calculated.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
distance	A dist or matrix object representing the phylogenetic or functional distance between species. If both tree and distance are missing, taxonomic evenness is calculated.
method	Calculate evenness using "expected" values (default) or values based on "contribution" of species to the tree.

`func`	Calculate evenness using "Camargo" (1993; default) or "Bulla" (1994) index.
`abund`	A boolean (T/F) indicating whether evenness should be calculated using abundance data.

Details

Contribution to evenness is calculated using a leave-one-out approach, whereby the contribution of a single observation is the total evenness minus the evenness calculated without that observation. Evenness is based on the index of Camargo (1993) or Bulla (1994) using the values of both species abundances and edge lengths in the tree (if PD/FD). Note that the contribution of a species or individual can be negative, if the removal of an observation increases the total evenness.

If no tree or distance is provided the result is calculated for taxonomic evenness using the original index.

Value

A matrix of sites x species (or a vector if no comm is given).

References

Bulla, L. (1994) An index of evenness and its associated diversity measure. Oikos, 70: 167-171.

Camargo, J.A. (1993) Must dominance increase with the number of subordinate species in competitive interactions? Journal of Theoretical Biology, 161: 537-542.

Examples

```
comm <- matrix(c(1,2,0,5,5,1,1,0,0,0,0,2,2,0,0,1,1,1,1,100), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method = "euclidean")
tree <- hclust(distance, method = "average")
evenness.contribution(comm)
evenness.contribution(tree = tree, func = "bulla")
evenness.contribution(comm, tree)
evenness.contribution(comm, tree, method = "contribution")
evenness.contribution(comm, tree, abund = FALSE)
```

`fill` *Filling missing data.*

Description

Estimation of missing trait values (NA) based on different methods.

Usage

```
fill(trait, method = "regression", group = NULL, weight = NULL, step = TRUE)
```

Arguments

`trait` A species x traits matrix (a species or individual for each row and traits as columns).

`method` Method for imputing missing data. One of "mean" (mean value of the trait), "median" (median value of the trait), "similar" (input from closest species), "regression" (linear regression), "w_regression" (regression weighted by species distance), or "PCA" (Principal Component Analysis).

`group` A vector (string of characters, factorial, etc.) whose values indicate which species belong to the same group as the missing and should be used in the estimation of missing data. If NULL all species will be used.

`weight` A hclust, phylo or dist object to calculate the distance between species and use as weights. Note that the order of tip labels in trees or of species in the distance matrix should be the same as the order of species in trait.

`step` A boolean (T/F) indicating if a stepwise regression model based on AIC should be performed. Ignored is regression is not used.

Details

Inputs missing data in the trait matrix based on different methods (see Taugourdeau et al. 2014; Johnson et al. 2021 for comparisons among the performance of different methods). The simplest approach is the average imputation ("mean" or "median"), calculating the mean/median of the values for that trait based on all the observations that are non-missing. It has the advantage of keeping the same mean and the same sample size, but many disadvantages. The "similar" method inputs a systematically chosen value from the closest species who has similar values on other variables. The default method is linear regression ("regression"), where the predicted value is obtained by regressing the missing variable on other variables. This preserves relationships among variables involved in the imputation model, but not variability around predicted values (i.e., may lead to extrapolations). The "w_regression" takes into account the relative distance among species in the imputation of missing traits, based on the phylogenetic or functional distance between missing and non-missing species. The "PCA" method performs PCA with incomplete data sensu Podani et al. (2021). Note that for PCA and regressions methods the performance of the prediction increases as the number of collinear traits increase.

Value

A trait matrix with missing data (NA) filled with predicted values. If method = "PCA" the function returns the standard output of a principal component analysis as a list with: Eigenvalues Positive eigenvalues Positive eigenvalues as percent Square root of eigenvalues Eigenvectors Component scores Variable scores Object scores in a biplot Variable scores in a biplot

References

Johnson, T.F., Isaac, N.J., Paviolo, A. & Gonzalez-Suarez, M. (2021). Handling missing values in trait data. Global Ecology and Biogeography, 30: 51-62.

Podani, J., Kalapos, T., Barta, B. & Schmera, D. (2021). Principal component analysis of incomplete data. A simple solution to an old problem. Ecological Informatics, 101235.

Taugourdeau, S., Villerd, J., Plantureux, S., Huguenin-Elie, O. & Amiaud, B. (2014). Filling the gap in functional trait databases: use of ecological hypotheses to replace missing data. Ecology and Evolution, 4: 944-958.

Examples

```
## Not run:
trait <- iris[,-5]
group <- iris[,5]

#Generating some random missing data
for (i in 1:10)
trait[sample(nrow(trait), 1), sample(ncol(trait), 1)] <- NA

#Estimating the missing data with different methods
fill(trait, "mean")
fill(trait, "mean", group)
fill(trait, "median")
fill(trait, "median", group)
fill(trait, "similar")
fill(trait, "similar", group)
fill(trait, "regression", step = FALSE)
fill(trait, "regression", group, step = TRUE)
fill(trait, "w_regression", step = TRUE)
fill(trait, "w_regression", weight = dist(trait), step = TRUE)
fill(trait, "PCA")

## End(Not run)
```

functree *Functional tree for 338 species of spiders*

Description

A dataset representing the functional tree for 338 species of spiders captured in Portugal. For each species were recorded: average size, type of web, type of hunting, stenophagy, vertical stratification in vegetation and circadial activity. Details are described in: Cardoso, P., Pekar, S., Jocque, R. & Coddington, J.A. (2011) Global patterns of guild composition and functional diversity of spiders. PLoS One, 6: e21710.

Usage

```
data(functree)
```

Format

An hclust object with 338 species.

gamma *Gamma diversity (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Observed richness among multiple sites.

Usage

```
gamma(comm, tree)
```

Arguments

`comm`	A sites x species matrix, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

Details

TD is equivalent to species richness. Calculations of PD and FD are based on Faith (1992) and Petchey & Gaston (2002, 2006), which measure PD and FD of a community as the total branch length of a tree linking all species represented in such community. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in comm must be the same as in tree.

Value

A single value of gamma.

References

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation, 61, 1-10.

Petchey, O.L. & Gaston, K.J. (2002) Functional diversity (FD), species richness and community composition. Ecology Letters, 5, 402-411.

Petchey, O.L. & Gaston, K.J. (2006) Functional diversity: back to basics and looking forward. Ecology Letters, 9, 741-758.

Examples

```
comm <- matrix(c(0,0,1,1,0,0,2,1,0,0), nrow = 2, ncol = 5, byrow = TRUE)
trait = 1:5
tree <- hclust(dist(c(1:5), method = "euclidean"), method = "average")
alpha(comm)
gamma(comm)
gamma(comm, trait)
gamma(comm, tree)
```

gdm *General dynamic model of oceanic island biogeography (GDM).*

Description

Fits and compares several of the most supported models for the GDM (using TD, PD or FD).

Usage

```
gdm(comm, tree, area, time)
```

Arguments

comm	Either a vector with the diversity values per island, or an island x species matrix.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree (used only to fit the PD or FD GDM, requires comm to be a sites x species matrix).
area	A vector with the area of islands.
time	A vector with the age of islands. If not given, the species-area relationship is returned instead.

Details

The general dynamic model of oceanic island biogeography was proposed to account for diversity patterns within and across oceanic archipelagos as a function of area and age of the islands (Whittaker et al. 2008). Several different equations have been found to describe the GDM, extending the different SAR models with the addition of a polynomial term using island age and its square (TT2), depicting the island ontogeny. The first to be proposed was an extension of the exponential model (Whittaker et al. 2008), the power model extensions following shortly after (Fattorini 2009; Steinbauer et al. 2013), as was the linear model (Cardoso et al. 2020). The relationships for PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A matrix with the different model parameters and explanatory power.

References

Cardoso, P., Branco, V.V., Borges, P.A.V., Carvalho, J.C., Rigal, F., Gabriel, R., Mammola, S., Cascalho, J. & Correia, L. (2020) Automated discovery of relationships, models and principles in ecology. Frontiers in Ecology and Evolution, 8: 530135.

Fattorini, S. (2009) On the general dynamic model of oceanic island biogeography. Journal of Biogeography, 36: 1100-1110.

Steinbauer, M.J, Klara, D., Field, R., Reineking, B. & Beierkuhnlein, C. (2013) Re-evaluating the general dynamic theory of oceanic island biogeography. Frontiers of Biogeography, 5: 185-194.

Whittaker, R.J., Triantis, K.A. & Ladle, R.J. (2008) A general dynamic theory of oceanic island biogeography. Journal of Biogeography, 35: 977-994.

Examples

```
div <- c(1,3,5,8,10)
comm <- matrix(c(2,0,0,0,3,1,0,0,2,4,5,0,1,3,2,5,1,1,1,1), nrow = 5, ncol = 4, byrow = TRUE)
tree <- hclust(dist(c(1:4), method="euclidean"), method="average")
area <- c(10,40,80,160,160)
time <- c(1,2,3,4,5)
gdm(div,,area,time)
gdm(comm,tree,area,time)
gdm(div,,area)
```

geres *Sample data of spiders in Geres (Portugal)*

Description

A dataset containing the abundance of 338 spider species in each of 320 sampling units. Details are described in: Cardoso, P., Scharff, N., Gaspar, C., Henriques, S.S., Carvalho, R., Castro, P.H., Schmidt, J.B., Silva, I., Szuts, T., Castro, A. & Crespo, L.C. (2008) Rapid biodiversity assessment of spiders (Araneae) using semi-quantitative sampling: a case study in a Mediterranean forest. Insect Conservation and Diversity, 1: 71-84.

Usage

```
data(geres)
```

Format

A data frame with 320 sampling untis (rows) and 338 species (variables).

gower *Gower distance.*

Description

Calculates Gower distances between observations.

Usage

```
gower(trait, convert = NULL, weight = NULL)
```

Arguments

`trait` A species x traits matrix or data.frame.

`convert` A vector of column numbers, usually categorical variables, to be converted to dummy variables.

`weight` A vector of column numbers with weights for each variable. Its length must be equal to the number of columns in trait.

Details

The Gower distance allows continuous, ordinal, categorical or binary variables, with possible weighting (Pavoine et al. 2009). NAs are allowed as long as each pair of species has at least one trait value in common. If convert is given the algorithm will convert these column numbers to dummy variables. Otherwise it will convert all columns with factors or characters as values.

Value

A dist object with pairwise distances between species.

References

Pavoine et al. (2009) On the challenge of treating various types of variables: application for improving the measurement of functional diversity. Oikos, 118: 391-402.

Examples

```
trait = data.frame(body = c(NA,2,3,4,4), beak = c(1,1,1,1,2))
gower(trait)
gower(trait, weight = c(1, 0))
```

`guadiana` *Sample data of spiders in Guadiana (Portugal)*

Description

A dataset containing the abundance of 338 spider species in each of 320 sampling units. Details are described in: Cardoso, P., Henriques, S.S., Gaspar, C., Crespo, L.C., Carvalho, R., Schmidt, J.B., Sousa, P. & Szuts, T. (2009) Species richness and composition assessment of spiders in a Mediterranean scrubland. Journal of Insect Conservation, 13: 45-55.

Usage

```
data(guadiana)
```

Format

A data frame with 192 sampling units (rows) and 338 species (variables).

hill *Hill numbers.*

Description

Hill numbers with possible rarefaction, multiple sites simultaneously.

Usage

```
hill(comm, q = 0, raref = 0, runs = 100)
```

Arguments

comm	A sites x species matrix, with abundance data.
q	Hill number order: q(0) = species richness, q(1) ~ Shannon diversity, q(2) ~ Simpson diversity, and so on...
raref	An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.
runs	Number of resampling runs for rarefaction. If not specified, default is 100.

Details

Hill numbers are based on the number of equally abundant species that would match the current diversity. Depending on the single parameter they give more or less weight to rare species (Jost 2002).

Value

A matrix of sites x diversity values (either "Hill q" OR "Mean, Median, Min, LowerCL, UpperCL and Max").

References

Hill, M.O. (1973). Diversity and evenness: a unifying notation and its consequences. Ecology, 54: 427-432.

Examples

```
comm <- matrix(c(0,0,1,1,0,0,100,1,0,0), nrow = 2, ncol = 5, byrow = TRUE)
hill(comm)
hill(comm, q = 1)
hill(comm, q = 4, 1)
```

hull.alpha *Alpha diversity using convex hull hypervolumes.*

Description

Estimation of functional richness of one or multiple sites, based on convex hull hypervolumes.

Usage

```
hull.alpha(comm)
```

Arguments

comm	A ’convhulln’ object or list, preferably built with function hull.build.

Details

Estimates the functional richness (alpha FD) of one or more communities using convex hull hypervolumes. Functional richness is expressed as the total volume of the convex hull.

Value

One value or a vector of alpha diversity values for each site.

Examples

```
comm = rbind(c(1,3,0,5,3), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = hull.build(comm[1,], trait)
hull.alpha(hv)
hvlist = hull.build(comm, trait, axes = 2)
hull.alpha(hvlist)
```

hull.beta *Beta diversity partitioning using convex hull hypervolumes.*

Description

Pairwise beta diversity partitioning into replacement and net difference in amplitude components of convex hulls.

Usage

```
hull.beta(comm, func = "jaccard", comp = FALSE)
```

Arguments

comm	A list of 'convhulln' objects, preferably built with function hull.build.
func	Partial match indicating whether the Jaccard (default) or Soerensen family of beta diversity measures should be used.
comp	Boolean indicating whether beta diversity components (shared and unique fractions) should be returned.

Details

Computes a pairwise decomposition of the overall differentiation among kernel hypervolumes into two components: the replacement (shifts) of space between hypervolumes and net differences between the amount of space enclosed by each hypervolume. The beta diversity measures used here follow the FD partitioning framework where Btotal = Breplacement + Brichness. Beta diversity ranges from 0 (when hypervolumes are identical) to 1 (when hypervolumes are fully dissimilar). See Carvalho & Cardoso (2020) and Mammola & Cardoso (2020) for the full formulas of beta diversity used here.

Value

Three pairwise distance matrices, one per each of the three beta diversity components.

References

Carvalho, J.C. & Cardoso, P. (2020) Decomposing the causes for niche differentiation between species using hypervolumes. Frontiers in Ecology and Evolution. https://doi.org/10.3389/fevo.2020.00243

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution. https://doi.org/10.1111/2041-210X.13424

Examples

```
comm <- rbind(c(1,1,1,1,1), c(1,1,1,1,1), c(0,0,1,1,1),c(0,0,1,1,1))
colnames(comm) = c("SpA","SpB","SpC","SpD", "SpE")
rownames(comm) = c("Site 1","Site 2","Site 3","Site 4")

trait <- cbind(c(2.2,4.4,6.1,8.3,3),c(0.5,1,0.5,0.4,4))
colnames(trait) = c("Trait 1","Trait 2")
rownames(trait) = colnames(comm)

hvlist = hull.build(comm, trait)
hull.beta(hvlist)
hvlist = hull.build(comm, trait, axes = 2)
hull.beta(hvlist, comp = TRUE)
```

hull.build — *Build convex hull hypervolumes.*

Description

Builds convex hull hypervolumes for each community from incidence and trait data.

Usage

```
hull.build(
  comm,
  trait,
  distance = "gower",
  weight = NULL,
  axes = 0,
  convert = NULL
)
```

Arguments

`comm`	A sites x species matrix, data.frame or vector, with incidence data about the species in the community.
`trait`	A species x traits or axes matrix or data.frame (often from hyper.build) or, alternatively, a dist object.
`distance`	One of "gower" or "euclidean". Not used if trait is a dist object.
`weight`	A vector of column numbers with weights for each variable. Its length must be equal to the number of columns in trait. Only used if axes > 0 and if trait is not a dist object.
`axes`	If 0, no transformation of data is done. If 0 < axes <= 1 a PCoA is done with Gower/euclidean distances and as many axes as needed to achieve this proportion of variance explained are selected. If axes > 1 these many axes are selected.
`convert`	A vector of column numbers, usually categorical variables, to be converted to dummy variables. Only used if axes > 0 and if trait is not a dist object.

Details

The hypervolumes can be constructed with the given data or data can be transformed using PCoA after traits are dummyfied (if needed) and standardized (always). Beware that if transformations are required, all communities to be compared should be built simultaneously to guarantee comparability. In such case, one might want to first run hyper.build and use the resulting data in different runs of hull.build. See function hyper.build for more details.

Value

A 'convhulln' object or a list, representing the hypervolumes of each community.

Examples

```
comm = rbind(c(1,3,0,5,3), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = hull.build(comm[1,], trait)
plot(hv)
hvlist = hull.build(comm, trait)
plot(hvlist[[2]])
hvlist = hull.build(comm, trait, axes = 2, weight = c(1,2))
plot(hvlist[[1]])
```

`hull.contribution` *Contribution of each observation to a convex hull hypervolume.*

Description

Contribution of each species or individual to the total volume of one or more convex hulls.

Usage

```
hull.contribution(comm, relative = FALSE)
```

Arguments

`comm`	A ’convhulln’ object or list, preferably built with function hull.build.
`relative`	A boolean (T/F) indicating whether contribution should be relative to total PD or FD (proportional contribution per individual or species). If FALSE, the sum of contributions for each site is equal to total PD/FD, if TRUE it is 1.

Details

The contribution of each observation (species or individual) to the total volume of a convex hull, calculated as the difference in volume between the total convex hull and a second hypervolume lacking this specific observation (i.e., leave-one-out approach; Mammola & Cardoso, 2020).

Value

A vector or matrix with the contribution values of each species or individual for each site.

References

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution. https://doi.org/10.1111/2041-210X.13424

Examples

```
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = hull.build(comm[1,], trait)
hull.contribution(hv)
hvlist = hull.build(comm, trait, axes = 2)
hull.contribution(hvlist, relative = TRUE)
```

`hull.gamma` *Gamma diversity using convex hull hypervolumes.*

Description

Estimation of functional richness of multiple sites, based on convex hull hypervolumes.

Usage

```
hull.gamma(comm)
```

Arguments

`comm`	A ’convhulln’ object or list, preferably built with function hull.build.

Details

Estimates the functional richness (gamma FD) of multiple communities using convex hull hypervolumes. Functional richness is expressed as the total volume of the convex hull.

Value

A single value of gamma.

Examples

```
comm = rbind(c(1,3,0,5,3), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = hull.build(comm[1,], trait)
hull.alpha(hv)
hull.gamma(hv)
```

```
hvlist = hull.build(comm, trait, axes = 2)
hull.alpha(hvlist)
hull.gamma(hvlist)
```

hull.sad *Species-abundance distribution (SAD) using convex hulls.*

Description

Fits the SAD to community abundance data using convex hulls.

Usage

```
hull.sad(comm, octaves = TRUE, scale = FALSE, raref = 0, runs = 100)
```

Arguments

`comm` A ’convhulln’ object or list, preferably built with function hull.build.

`octaves` a boolean indicating whether octaves should be calculated.

`scale` scale y-axis to sum 1.

`raref` An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.

`runs` Number of resampling runs for rarefaction. If not specified, default is 100.

Details

The Species Abundance Distribution describes the commonness and rarity in ecological systems. It was recently expanded to accomodate phylegenetic and functional differences between species (Matthews et al., subm.). Classes defined as n = 1, 2-3, 4-7, 8-15, ... Rarefaction allows comparison of sites with different total abundances.

Value

A vector or matrix with the different values per class per community.

References

Matthews et al. (subm.) Phylogenetic and functional dimensions of the species abundance distribution.

Examples

```
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = hull.build(comm, trait)
hull.sad(hv, scale = TRUE)
hull.sad(hv, octaves = FALSE)
hull.sad(hv, raref = TRUE)
```

`hyper.build` *Build hyperspace.*

Description

Builds hyperspace by transforming trait or distance data to use with either hull.build or kernel.build.

Usage

```
hyper.build(trait, distance = "gower", weight = NULL, axes = 1, convert = NULL)
```

Arguments

`trait` A species x traits matrix or data.frame or, alternatively, a dist object.

`distance` One of "gower" or "euclidean". Not used if trait is a dist object.

`weight` A vector of column numbers with weights for each variable. Its length must be equal to the number of columns in trait. Only used if axes > 0 and if trait is not a dist object.

`axes` If 0, no transformation of data is done. If 0 < axes <= 1 a PCoA is done with Gower/euclidean distances and as many axes as needed to achieve this proportion of variance explained are selected. If axes > 1 these many axes are selected.

`convert` A vector of column numbers, usually categorical variables, to be converted to dummy variables. Only used if axes > 0 and if trait is not a dist object.

Details

The hyperspace can be constructed with the given data or data can be transformed using PCoA after traits are dummyfied (if needed) and standardized (always). Gower distance (Pavoine et al. 2009) allows continuous, ordinal, categorical or binary variables, with possible weighting. NAs are allowed as long as each pair of species has at least one trait value in common. If convert is given the algorithm will convert these column numbers to dummy variables. Otherwise it will convert all columns with factors or characters as values. Note that each community should have at least 3 species and more species than traits or axes (if axes > 0) to build convex hull hypervolumes.

Transformation of traits is recommended if (Carvalho & Cardoso, 2020): 1) Some traits are not continuous; 2) Some traits are correlated; or 3) There are less species than traits + 1, in which case the number of axes should be smaller.

Value

A matrix with the coordinates of each species in hyperspace.

References

Carvalho, J.C. & Cardoso, P. (2020) Decomposing the causes for niche differentiation between species using hypervolumes. Frontiers in Ecology and Evolution, 8: 243.

Pavoine et al. (2009) On the challenge of treating various types of variables: application for improving the measurement of functional diversity. Oikos, 118: 391-402.

Examples

```
trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = c("SpA", "SpB", "SpC", "SpD", "SpE")

hs = hyper.build(trait, weight = c(1,2), axes = 2)
plot(hs)
```

---

hyper.quality *Quality of hyperspace.*

Description

Assess the quality of a functional hyperspace.

Usage

```
hyper.quality(distance, hyper)
```

Arguments

`distance`	A dist object representing the initial distances between species.
`hyper`	A matrix with coordinates data from function hyper.build.

Details

This is used for any representation using hyperspaces, including convex hull and kernel-density hypervolumes. The algorithm calculates the inverse of the squared deviation between initial and euclidean distances (Maire et al. 2015) after standardization of all values between 0 and 1 for simplicity of interpretation. A value of 1 corresponds to maximum quality of the functional representation. A value of 0 corresponds to the expected value for an hyperspace where all distances between species are 1.

Value

A single value of quality.

References

Maire et al. (2015) How many dimensions are needed to accurately assess functional diversity? A pragmatic approach for assessing the quality of functional spaces. Global Ecology and Biogeography, 24: 728:740.

Examples

```
trait = data.frame(body = c(1,2,3,4,4), beak = c(1,1,1,1,2))
distance = gower(trait)

hyper = hyper.build(trait, axes = 2)
hyper.quality(distance, hyper)

hyper = hyper.build(trait, axes = 0)
hyper.quality(distance, hyper)
```

iaor *Interspecific abundance-occupancy relationship (IAOR).*

Description

Fits and compares several of the most supported models for the IAOR.

Usage

```
iaor(comm)
```

Arguments

`comm`	A sites x species matrix with abundance values.

Details

Locally abundant species tend to be widespread while locally rare species tend to be narrowly distributed. That is, for a given species assemblage, there is a positive interspecific abundance-occupancy relationship (Brown 1984). This function compares some of the most commonly used and theoretically or empirically suported models (Nachman 1981; He & Gaston 2000; Cardoso et al. 2020).

Value

A matrix with the different model parameters and explanatory power.

References

Brown, J.H. (1984) On the relationship between abundance and distribution of species. American Naturalist, 124: 255-279.

Cardoso, P., Branco, V.V., Borges, P.A.V., Carvalho, J.C., Rigal, F., Gabriel, R., Mammola, S., Cascalho, J. & Correia, L. (2020) Automated discovery of relationships, models and principles in ecology. Frontiers in Ecology and Evolution, 8: 530135.

He, F.L. & Gaston, K.J. (2000) Estimating species abundance from occurrence. American Naturalist, 156: 553-559.

Nachman, G. (1981) A mathematical model of the functional relationship between density and spatial distribution of a population. Journal of Animal Ecology, 50: 453-460.

Examples

```
comm <- matrix(c(4,3,2,1,5,4,3,2,3,2,1,0,6,3,0,0,0,0,0,0), nrow = 5, ncol = 4, byrow = TRUE)
iaor(comm)
```

---

kernel.alpha *Alpha diversity using kernel density hypervolumes.*

Description

Estimation of functional richness of one or multiple sites, based on n-dimensional hypervolumes.

Usage

```
kernel.alpha(comm)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.

Details

Estimates the functional richness (alpha FD) of one or more communities using kernel density hypervolumes, as implemented in Blonder et al. (2014, 2018). Functional richness is expressed as the total volume of the n-dimensional hypervolume (Mammola & Cardoso, 2020). Note that the hypervolume is dimensionless, and that only hypervolumes with the same number of dimensions can be compared in terms of functional richness. Given that the density and positions of stochastic points in the hypervolume are probabilistic, the functional richness of the trait space will intimately depend on the quality of input hypervolumes (details in Mammola & Cardoso, 2020).

Value

A value or vector of alpha diversity values for each site.

References

Blonder, B., Lamanna, C., Violle, C. & Enquist, B.J. (2014) The n-dimensional hypervolume. Global Ecology and Biogeography, 23: 595-609.

Blonder, B., Morrow, C.B., Maitner, B., Harris, D.J., Lamanna, C., Violle, C., ... & Kerkhoff, A.J. (2018) New approaches for delineating n-dimensional hypervolumes. Methods in Ecology and Evolution, 9: 305-319.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.alpha(hv)
hvlist = kernel.build(comm, trait, axes = 0.8)
kernel.alpha(hvlist)

## End(Not run)
```

kernel.beta *Beta diversity partitioning using kernel density hypervolumes.*

Description

Pairwise beta diversity partitioning into replacement and net difference in amplitude components of n-dimensional hypervolumes.

Usage

```
kernel.beta(comm, func = "jaccard", comp = FALSE)
```

Arguments

comm	A 'HypervolumeList' object, preferably built using function kernel.build.
func	Partial match indicating whether the Jaccard or Soerensen family of beta diversity measures should be used. If not specified, default is Jaccard.
comp	Boolean indicating whether beta diversity components (shared and unique fractions) should be returned

Details

Computes a pairwise decomposition of the overall differentiation among kernel hypervolumes into two components: the replacement (shifts) of space between hypervolumes and net differences between the amount of space enclosed by each hypervolume. The beta diversity measures used here follow the FD partitioning framework developed by Carvalho & Cardoso (2020), where Btotal = Breplacement + Brichness. Beta diversity ranges from 0 (when hypervolumes are identical) to 1 (when hypervolumes are fully dissimilar). See Carvalho & Cardoso (2020) and Mammola & Cardoso (2020) for the full formulas of beta diversity used here.

Value

Three pairwise distance matrices, one per each of the three beta diversity components. If comp = TRUE also three distance matrices with beta diversity components.

References

Carvalho, J.C. & Cardoso, P. (2020) Decomposing the causes for niche differentiation between species using hypervolumes. Frontiers in Ecology and Evolution, 8: 243.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm <- rbind(c(1,1,1,1,1), c(1,1,1,1,1), c(0,0,1,1,1),c(0,0,1,1,1))
colnames(comm) = c("SpA","SpB","SpC","SpD", "SpE")
rownames(comm) = c("Site 1","Site 2","Site 3","Site 4")

trait <- cbind(c(2.2,4.4,6.1,8.3,3),c(0.5,1,0.5,0.4,4),c(0.7,1.2,0.5,0.4,5),c(0.7,2.2,0.5,0.3,6))
colnames(trait) = c("Trait 1","Trait 2","Trait 3","Trait 4")
rownames(trait) = colnames(comm)

hvlist = kernel.build(comm, trait)
kernel.beta(hvlist)
hvlist = kernel.build(comm, trait, axes = 0.9)
kernel.beta(hvlist, comp = TRUE)

## End(Not run)
```

kernel.beta.evenness *Functional beta diversity evenness using kernel density hypervolumes.*

Description

Difference of evenness between pairs of sites, measuring the regularity of stochastic points distribution within the total functional space.

Usage

```
kernel.beta.evenness(comm)
```

Arguments

comm	A 'HypervolumeList' object, preferably built using function kernel.build.

Details

This measure is simply the pairwise difference of evenness calculated based on the functional evenness (Mason et al., 2005) of a n-dimensional hypervolume, namely the regularity of stochastic points distribution within the total trait space (Mammola & Cardoso, 2020). Evenness is calculated as the overlap between the observed hypervolume and a theoretical hypervolume where traits and abundances are evenly distributed within the range of their values (Carmona et al., 2016, 2019).

Value

Distance matrix between sites.

References

Carmona, C.P., de Bello, F., Mason, N.W.H. & Leps, J. (2016) Traits without borders: integrating functional diversity across scales. Trends in Ecology and Evolution, 31: 382-394.

Carmona, C.P., de Bello, F., Mason, N.W.H. & Leps, J. (2019) Trait probability density (TPD): measuring functional diversity across scales based on TPD with R. Ecology, 100: e02876.

Mason, N.W.H., Mouillot, D., Lee, W.G. & Wilson, J.B. (2005) Functional richness, functional evenness and functional divergence: the primary components of functional diversity. Oikos, 111: 112-118.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm <- rbind(c(1,1,1,1,1), c(1,1,1,1,1), c(0,0,1,1,1),c(0,0,1,1,1))
colnames(comm) = c("SpA","SpB","SpC","SpD", "SpE")
rownames(comm) = c("Site 1","Site 2","Site 3","Site 4")

trait <- cbind(c(2.2,4.4,6.1,8.3,3),c(0.5,1,0.5,0.4,4),c(0.7,1.2,0.5,0.4,5),c(0.7,2.2,0.5,0.3,6))
colnames(trait) = c("Trait 1","Trait 2","Trait 3","Trait 4")
rownames(trait) = colnames(comm)

hvlist = kernel.build(comm, trait)
kernel.beta.evenness(hvlist)
hvlist = kernel.build(comm, trait, axes = 0.9)
kernel.beta.evenness(hvlist)

## End(Not run)
```

kernel.build *Build kernel hypervolumes.*

Description

Builds kernel density hypervolumes from trait data.

Usage

```
kernel.build(
  comm,
  trait,
  distance = "gower",
  method.hv = "gaussian",
  abund = TRUE,
  weight = NULL,
  axes = 0,
  convert = NULL,
  cores = 1,
  ...
)
```

Arguments

comm	A sites x species matrix, data.frame or vector, with incidence or abundance data about the species in the community.
trait	A species x traits or axes matrix or data.frame (often from hyper.build) or, alternatively, a dist object.
distance	One of "gower" or "euclidean". Not used if trait is a dist object.
method.hv	Method for constructing the 'Hypervolume' object. One of "gaussian" (Gaussian kernel density estimation, default), "box" (box kernel density estimation), or "svm" (one-class support vector machine). See respective functions of the hypervolume R package for details.
abund	A boolean (T/F) indicating whether abundance data should be used as weights in hypervolume construction. Only works if method.hv = "gaussian".
weight	A vector of column numbers with weights for each variable. Its length must be equal to the number of columns in trait. Only used if axes > 0 and if trait is not a dist object.
axes	If 0, no transformation of data is done. If 0 < axes <= 1 a PCoA is done with Gower/euclidean distances and as many axes as needed to achieve this proportion of variance explained are selected. If axes > 1 these many axes are selected.
convert	A vector of column numbers, usually categorical variables, to be converted to dummy variables. Only used if axes > 0 and if trait is not a dist object.

cores	Number of cores to be used in parallel processing. If = 0 all available cores are used. Beware that multicore for Windows is not optimized yet and it often takes longer than single core.
...	further arguments to be passed to hypervolume::hypervolume

Details

The hypervolumes can be constructed with the given data or data can be transformed using PCoA after traits are dummyfied (if needed) and standardized (always). Beware that if transformations are required, all communities to be compared should be built simultaneously to guarantee comparability. In such case, one might want to first run hyper.build and use the resulting data in different runs of kernel.build. See function hyper.build for more details.

Value

A ’Hypervolume’ or ’HypervolumeList’, representing the hypervolumes of each community.

Examples

```
## Not run:
comm = rbind(c(1,1,0,5,1), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site1", "Site2")

trait = data.frame(body = c(1,2,3,1,2), beak = c(1,2,3,2,1))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
plot(hv)
hvlist = kernel.build(comm, trait, abund = FALSE, cores = 2)
plot(hvlist)
hvlist = kernel.build(comm, trait, method.hv = "box", weight = c(1,2), axes = 2)
plot(hvlist)

## End(Not run)
```

kernel.contribution *Contribution of each observation to the kernel density hypervolume.*

Description

Contribution of each species or individual to the total volume of one or more kernel hypervolumes.

Usage

```
kernel.contribution(comm, func = "neighbor", relative = FALSE)
```

Arguments

`comm`	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.
`func`	Calculate contribution using either closest "neighbor" or leave "one out" approach.
`relative`	A boolean (T/F) indicating whether contribution should be relative to total FD (proportional contribution per individual or species). If FALSE, the sum of contributions for each site is equal to total FD, if TRUE it is 1.

Details

Contribution is a measure of functional rarity (sensu Violle et al., 2017; Carmona et al., 2017) that allows to map the contribution of each observation to the richness components of FD (Mammola & Cardoso, 2020). If using func = "neighbor", each random point will be attributed to the closest species. The contribution of each species will be proportional to the number of its points. The sum of contributions of all species is equal to total richness. Note that the contribution of a species or individual can be negative if leave-one-out approach is taken, if the removal of an observation increases the total volume (see Figure 2d in Mammola & Cardoso 2020). This might happen, although not always, in cases when the presence of a given species decreases the average distance between all the species in the community, i.e., when a given species is close to the "average" species of that community, making that community less diverse in some sense (Mammola & Cardoso, 2020).

Value

A matrix with the contribution values of each species or individual for each site.

References

Carmona, C.P., de Bello, F., Sasaki, T., Uchida, K. & Partel, M. (2017) Towards a common toolbox for rarity: A response to Violle et al. Trends in Ecology and Evolution, 32(12): 889-891.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Violle, C., Thuiller, W., Mouquet, N., Munoz, F., Kraft, N.J.B., Cadotte, M.W., ... & Mouillot, D. (2017) Functional rarity: The ecology of outliers. Trends in Ecology and Evolution, 32: 356-367.

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.contribution(hv)
```

```
hvlist = kernel.build(comm, trait, axes = 2)
kernel.contribution(hvlist)
kernel.contribution(hvlist, relative = TRUE)

## End(Not run)
```

kernel.dispersion *Functional dispersion of kernel density hypervolumes.*

Description

Average distance to centroid or dissimilarity between random points within the boundaries of the kernel density hypervolume.

Usage

```
kernel.dispersion(comm, func = "dissimilarity", frac = 0.1)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.
func	Function for calculating dispersion. One of 'divergence', 'dissimilarity' or 'regression'.
frac	A value between 0.01 and 1, indicating the fraction of random points to be used. Default is 0.1.

Details

This function calculates dispersion either: i) as the average distance between stochastic points within the kernel density hypervolume and the centroid of these points (divergence; Laliberte & Legendre, 2010; see also Carmona et al., 2019); ii) as the average distance between all points (dissimilarity, see also function BAT::dispersion); or iii) as the average distance between stochastic points within the kernel density hypervolume and a regression line fitted through the points. The number of stochastic points is controlled by the 'frac' parameter (increase this number for less deviation in the estimation).

Value

A value or vector of dispersion values for each site.

References

Carmona, C.P., de Bello, F., Mason, N.W.H. & Leps, J. (2019) Trait probability density (TPD): measuring functional diversity across scales based on TPD with R. Ecology, 100: e02876.

Laliberte, E. & Legendre, P. (2010) A distance-based framework for measuring functional diversity from multiple traits. Ecology 91: 299-305.

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.dispersion(hv)
hvlist = kernel.build(comm, trait, axes = 2)
kernel.dispersion(hvlist)
kernel.dispersion(hvlist, func = "divergence")

## End(Not run)
```

---

kernel.evenness *Functional evenness of kernel density hypervolumes.*

Description

Functional evenness of a community, measuring the regularity of stochastic points distribution within the total functional space.

Usage

```
kernel.evenness(comm)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.

Details

This function measures the functional evenness (Mason et al., 2005) of a n-dimensional hypervolume, namely the regularity of stochastic points distribution within the total trait space (Mammola & Cardoso, 2020). Evenness is calculated as the overlap between the observed hypervolume and a theoretical hypervolume where traits and abundances are evenly distributed within the range of their values (Carmona et al., 2016, 2019).

Value

A value or vector of evenness values for each site.

References

Carmona, C.P., de Bello, F., Mason, N.W.H. & Leps, J. (2016) Traits without borders: integrating functional diversity across scales. Trends in Ecology and Evolution, 31: 382-394.

Carmona, C.P., de Bello, F., Mason, N.W.H. & Leps, J. (2019) Trait probability density (TPD): measuring functional diversity across scales based on TPD with R. Ecology, 100: e02876.

Mason, N.W.H., Mouillot, D., Lee, W.G. & Wilson, J.B. (2005) Functional richness, functional evenness and functional divergence: the primary components of functional diversity. Oikos, 111: 112-118.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm = rbind(c(100,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.evenness(hv)
hv = kernel.build(comm[1,], trait, abund = FALSE)
kernel.evenness(hv)
hvlist = kernel.build(comm, trait, axes = 2)
kernel.evenness(hvlist)

## End(Not run)
```

---

kernel.evenness.contribution

Contribution of each observation to the evenness of a kernel density hypervolume.

Description

Contribution of each species or individual to the evenness of one or more kernel hypervolumes.

Usage

```
kernel.evenness.contribution(comm)
```

Arguments

`comm`	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.

Details

The contribution of each observation (species or individual) to the total evenness of a kernel hypervolume. Contribution to evenness is calculated as the difference in evenness between the total hypervolume and a second hypervolume lacking this specific observation (i.e., leave-one-out approach; Mammola & Cardoso, 2020). Note that the contribution of a species or individual can be negative, if the removal of an observation increases the total evenness.

Value

A vector or matrix with the contribution values of each species or individual for each community or species respectively.

References

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm = rbind(c(100,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.evenness.contribution(hv)
hvlist = kernel.build(comm, trait)
kernel.evenness.contribution(hvlist)
hvlist = kernel.build(comm, trait, axes = 0.8)
kernel.evenness.contribution(hvlist)

## End(Not run)
```

`kernel.gamma` *Gamma diversity using kernel density hypervolumes.*

Description

Estimation of functional richness of multiple sites, based on n-dimensional hypervolumes.

Usage

```
kernel.gamma(comm)
```

Arguments

comm A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.

Details

Estimates the functional richness (gamma FD) of multiple communities using kernel density hypervolumes, as implemented in Blonder et al. (2014, 2018). Functional richness is expressed as the total volume of the n-dimensional hypervolume (Mammola & Cardoso, 2020). Note that the hypervolume is dimensionless, and that only hypervolumes with the same number of dimensions can be compared in terms of functional richness. Given that the density and positions of stochastic points in the hypervolume are probabilistic, the functional richness of the trait space will intimately depend on the quality of input hypervolumes (details in Mammola & Cardoso, 2020).

Value

A single value of gamma.

References

Blonder, B., Lamanna, C., Violle, C. & Enquist, B.J. (2014) The n-dimensional hypervolume. Global Ecology and Biogeography, 23: 595-609.

Blonder, B., Morrow, C.B., Maitner, B., Harris, D.J., Lamanna, C., Violle, C., ... & Kerkhoff, A.J. (2018) New approaches for delineating n-dimensional hypervolumes. Methods in Ecology and Evolution, 9: 305-319.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Examples

```
## Not run:
comm = rbind(c(1,3,2,2,2), c(0,0,0,2,2))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,5), beak = c(1,2,3,4,5))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.alpha(hv)
kernel.gamma(hv)
hvlist = kernel.build(comm, trait)
kernel.alpha(hvlist)
kernel.gamma(hvlist)

## End(Not run)
```

kernel.hotspots *Hotspots in hypervolumes.*

Description

Identify hotspots in kernel density hypervolumes based on minimum volume needed to cover a given proportion of random points.

Usage

```
kernel.hotspots(comm, prop = 0.5)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.
prop	Proportion of random points to be included.

Details

Estimates the hotspots of one or more communities using kernel density hypervolumes as in Carmona et al. (2021).

Value

A 'Hypervolume' or 'HypervolumeList' with the hotspots of each site.

References

Carmona, C.P., et al. (2021) Erosion of global functional diversity across the tree of life. Science Advances, 7: eabf2675. DOI: 10.1126/sciadv.abf2675

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,0,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
plot(hv)
kernel.alpha(hv)

hot = kernel.hotspots(hv, 0.5)
plot(hot)
kernel.alpha(hot)
```

```
hvlist = kernel.build(comm, trait)
hot = kernel.hotspots(hvlist, 0.1)
kernel.alpha(hot)

## End(Not run)
```

kernel.originality *Functional originality of observations in kernel density hypervolumes.*

Description

Average dissimilarity between a species or individual and a sample of random points within the boundaries of the n-dimensional hypervolume.

Usage

```
kernel.originality(comm, frac = 0.1, relative = FALSE)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object, preferably built using function kernel.build.
frac	A value between 0.01 and 1, indicating the fraction of random points to be used in the estimation of originality. Default is 0.1.
relative	A boolean (T/F) indicating whether originality should be relative to the most original species in the community.

Details

A measure of the originality (sensu Pavoine et al., 2005) of each observation (species or individuals) used to construct the n-dimensional hypervolume. In a probabilistic hypervolume, originality is calculated as the average distance between each observation to a sample of stochastic points within the boundaries of the n-dimensional hypervolume (Mammola & Cardoso, 2020). Originality is a measure of functional rarity (sensu Violle et al., 2017; Carmona et al., 2017) that allows to map the contribution of each observation to the divergence components of FD (Mammola & Cardoso, 2020). The number of sample points to be used in the estimation of the originality is controlled by the frac parameter. Increase frac for less deviation in the estimation, but mind that computation time also increases.

Value

A vector or matrix with the originality values of each species or individual in each site.

References

Carmona, C.P., de Bello, F., Sasaki, T., Uchida, K. & Partel, M. (2017) Towards a common toolbox for rarity: A response to Violle et al. Trends in Ecology and Evolution, 32: 889-891.

Mammola, S. & Cardoso, P. (2020) Functional diversity metrics using kernel density n-dimensional hypervolumes. Methods in Ecology and Evolution, 11: 986-995.

Pavoine, S., Ollier, S. & Dufour, A.-B. (2005) Is the originality of a species measurable? Ecology Letters, 8: 579-586.

Violle, C., Thuiller, W., Mouquet, N., Munoz, F., Kraft, N.J.B., Cadotte, M.W., ... & Mouillot, D. (2017) Functional rarity: the ecology of outliers. Trends in Ecology and Evolution, 32: 356-367.

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm[1,], trait)
kernel.originality(hv)
hvlist = kernel.build(comm, trait)
kernel.originality(hvlist)
kernel.originality(hvlist, relative = TRUE)

## End(Not run)
```

kernel.sad *Species-abundance distribution (SAD) using kernel density hypervolumes.*

Description

Fits the SAD to community abundance data based on n-dimensional hypervolumes.

Usage

```
kernel.sad(comm, octaves = TRUE, scale = FALSE, raref = 0, runs = 100)
```

Arguments

comm	A 'Hypervolume' or 'HypervolumeList' object necessarily built using function kernel.build.
octaves	a boolean indicating whether octaves should be calculated.
scale	scale y-axis to sum 1.

`raref`	An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.
`runs`	Number of resampling runs for rarefaction. If not specified, default is 100.

Details

The Species Abundance Distribution describes the commonness and rarity in ecological systems. It was recently expanded to accomodate phylegenetic and functional differences between species (Matthews et al., subm.). Classes defined as n = 1, 2-3, 4-7, 8-15, ... Rarefaction allows comparison of sites with different total abundances.

Value

A vector or matrix with the different values per class per community.

References

Matthews et al. (subm.) Phylogenetic and functional dimensions of the species abundance distribution.

Examples

```
## Not run:
comm = rbind(c(1,3,0,5,3), c(3,2,5,1,0))
colnames(comm) = c("SpA", "SpB", "SpC", "SpD", "SpE")
rownames(comm) = c("Site 1", "Site 2")

trait = data.frame(body = c(1,2,3,4,4), beak = c(1,5,4,1,2))
rownames(trait) = colnames(comm)

hv = kernel.build(comm, trait)
kernel.sad(hv, scale = TRUE)
kernel.sad(hv, octaves = FALSE)
kernel.sad(hv, raref = TRUE)

## End(Not run)
```

`kernel.similarity` *Pairwise similarity among kernel density hypervolumes.*

Description

Calculate pairwise distance metrics (centroid and minimum distance) and similarity indices (Intersection, Jaccard, Soerensen-Dice) among n-dimensional hypervolumes.

Usage

```
kernel.similarity(comm)
```

Arguments

`comm`	A ’HypervolumeList’ object, preferably built using function kernel.build.

Details

Computes a pairwise comparison between kernel density hypervolumes of multiple species or communities, based on the distance and similarity metrics implemented in hypervolume R package (Blonder et al., 2014, 2018). See Mammola (2019) for a description of the different indices, and a comparison between their performance. Note that computation time largely depends on the number of ’Hypervolume’ objects in the list, and scales almost exponentially with the number of hypervolume axes.

Value

Five pairwise distance matrices, one per each of the distance and similarity indices (in order: distance between centroids, minimum distance, Jaccard overlap, Soerensen-Dice overlap, and Intersection among hypervolumes).

References

Blonder, B., Lamanna, C., Violle, C. & Enquist, B.J. (2014) The n-dimensional hypervolume. Global Ecology and Biogeography, 23: 595-609.

Blonder, B., Morrow, C.B., Maitner, B., Harris, D.J., Lamanna, C., Violle, C., ... & Kerkhoff, A.J. (2018) New approaches for delineating n-dimensional hypervolumes. Methods in Ecology and Evolution, 9: 305-319.

Mammola, S. (2019) Assessing similarity of n-dimensional hypervolumes: Which metric to use?. Journal of Biogeography, 46: 2012-2023.

Examples

```
## Not run:
comm <- rbind(c(1,1,1,1,1), c(1,1,1,1,1), c(0,0,1,1,1),c(0,0,1,1,1))
colnames(comm) = c(”SpA”,”SpB”,”SpC”,”SpD”, ”SpE”)
rownames(comm) = c(”Site 1”,”Site 2”,”Site 3”,”Site 4”)

trait <- cbind(c(2.2,4.4,6.1,8.3,3),c(0.5,1,0.5,0.4,4),c(0.7,1.2,0.5,0.4,5),c(0.7,2.2,0.5,0.3,6))
colnames(trait) = c(”Trait 1”,”Trait 2”,”Trait 3”,”Trait 4”)
rownames(trait) = colnames(comm)

hvlist = kernel.build(comm, trait)
kernel.similarity(hvlist)
hvlist = kernel.build(comm, trait, axes = 0.9)
kernel.similarity(hvlist)

## End(Not run)
```

linnean *Create Linnean tree.*

Description

Creates a Linnean tree from taxonomic hierarchy.

Usage

```
linnean(taxa, distance = NULL)
```

Arguments

taxa	A taxonomic matrix with columns ordered according to linnean hierarchy starting with the highest.
distance	A vector with distances between levels starting with the highest. If not provided distances will be evenly distributed from 1 to 0.

Value

An hclust with all species.

Examples

```
family <- c("Nemesiidae", "Nemesiidae", "Zodariidae", "Zodariidae")
genus <- c("Iberesia", "Nemesia", "Zodarion", "Zodarion")
species <- c("Imachadoi", "Nungoliant", "Zatlanticum", "Zlusitanicum")
taxa <- cbind(family, genus, species)
par(mfrow = c(1, 2))
plot(linnean(taxa))
plot(linnean(taxa, c(2, 0.5, 0.3)))
```

mixture *Mixture model.*

Description

Mixture model by Hilario et al. subm.

Usage

```
mixture(
  comm,
  tree,
  q = 0,
  precision = 0.1,
  replace = TRUE,
  alpha = 0.05,
  param = TRUE,
  runs = 1000
)
```

Arguments

`comm` A sites x species matrix, with abundance data.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree. Will only be used if q = 0, in which case phylogenetic or functional richness are calculated instead of species richness.

`q` Hill number order: q(0) = species richness, q(1) ~ Shannon diversity, q(2) ~ Simpson diversity.

`precision` Precision of the proportion of each habitat type to be tested.

`replace` Boolean indicating whether simulations should be with or without (default) replacement.

`alpha` alpha value for significance level.

`param` Value is calculated with parametric or non-parametric method. The later is preferable when distribution of estimated values is not normally distributed.

`runs` Number of runs for the bootstrap providing confidence limits.

Details

A tool to assess biodiversity in landscapes containing varying proportions of n environments.

Value

A matrix with expected diversity at each proportion of different habitats in a landscape.

Author(s)

Renato Hilario & Pedro Cardoso

References

Chao et al. (2019) Proportional mixture of two rarefaction/extrapolation curves to forecast biodiversity changes under landscape transformation. Ecology Letters, 22: 1913-1922. https://doi.org/10.1111/ele.13322

Hilario et al. (subm.) Function ‘mixture’: A new tool to quantify biodiversity change under landscape transformation.

Examples

```
comm <- matrix(c(20,20,20,20,20,9,1,0,0,0,1,1,1,1,1), nrow = 3, ncol = 5, byrow = TRUE)
tree = hclust(dist(1:5))

hill(comm)
alpha(comm, tree)

mixture(comm, runs = 10)
mixture(comm, tree, replace = TRUE, runs = 10)
```

---

optim.alpha *Optimization of alpha diversity sampling protocols.*

Description

Optimization of alpha diversity sampling protocols when different methods and multiple samples per method are available.

Usage

```
optim.alpha(comm, tree, methods, base, seq = FALSE, runs = 1000, prog = TRUE)
```

Arguments

`comm`	A samples x species x sites array, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`methods`	A data.frame with the method names (1st column), number of samples per method (2nd column), base cost per method (3rd column, those costs that are fixed once a method is decided), and sample cost per method (those costs that add with each sample of the method, 4th column). If the last two columns are not provided base = 0 and sample = 1. The order of methods must be the same as in comm and the sum of the samples must be the same as nrow(comm).
`base`	A vector defining a base protocol from which to build upon (complementarity analysis) (length must be equal to number of methods).
`seq`	By default all combinations will be tested. If TRUE, a sequential approach will be taken, where methods are added based on the previous step. The method added will be the one providing the highest efficiency as quantified by the slope of the accumulation curve.
`runs`	Number of random permutations to be made to the sample order. Default is 1000.
`prog`	Present a text progress bar in the R console.

Details

Often a combination of methods allows sampling maximum plot diversity with minimum effort, as it allows sampling different sub-communities, contrary to using single methods. Cardoso (2009) proposed a way to optimize the number of samples per method when the target is to maximize sampled alpha diversity. It is applied here for TD, PD and FD, and for one or multiple sites simultaneously. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A matrix of samples x methods (values being optimum number of samples per method). The last column is the average alpha diversity value, rescaled to 0-1 if made for several sites, where 1 is the true diversity of each site.

References

Cardoso, P. (2009) Standardization and optimization of arthropod inventories - the case of Iberian spiders. Biodiversity and Conservation, 18, 3949-3962.

Examples

```
comm1 <- matrix(c(1,1,0,2,4,0,0,1,2,0,0,3), nrow = 4, ncol = 3, byrow = TRUE)
comm2 <- matrix(c(2,2,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm <- array(c(comm1, comm2), c(4,3,2))
colnames(comm) <- c("Sp1","Sp2","Sp3")

methods <- data.frame(method = c("Met1","Met2","Met3"),
           nSamples = c(1,2,1), fixcost = c(1,1,2), varCost = c(1,1,1))
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels <- colnames(comm)

## Not run:
  optim.alpha(comm,,methods)
  optim.alpha(comm,,methods, seq = TRUE)
  optim.alpha(comm, tree, methods)
  optim.alpha(comm,, methods = methods, seq = TRUE, base = c(0,1,1))

## End(Not run)
```

---

optim.alpha.stats *Efficiency statistics for alpha-sampling.*

Description

Average alpha diversity observed with a given number of samples per method.

Usage

```
optim.alpha.stats(comm, tree, methods, samples, runs = 1000)
```

Arguments

comm A samples x species x sites array, with either abundance or incidence data.

tree A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

methods A data.frame with the method names (1st column) and number of samples per method (2nd column). The order of methods must be the same as in comm and the sum of the samples must be the same as nrow(comm).

samples A vector with the number of samples per method to test.

runs Number of random permutations to be made to the sample order. Default is 1000.

Details

Different combinations of samples per method allow sampling different sub-communities. This function allows knowing the average TD, PD or FD values for a given combination, for one or multiple sites simultaneously. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A single average alpha diversity value. Rescaled to 0-1 if made for several sites, where 1 is the true diversity of each site.

Examples

```
comm1 <- matrix(c(1,1,0,2,4,0,0,1,2,0,0,3), nrow = 4, ncol = 3, byrow = TRUE)
comm2 <- matrix(c(2,2,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm <- array(c(comm1, comm2), c(4,3,2))
colnames(comm) <- c(”Sp1”,”Sp2”,”Sp3”)

tree <- hclust(dist(c(1:3), method=”euclidean”), method=”average”)
tree$labels <- colnames(comm)

methods <- data.frame(method = c(”Met1”,”Met2”,”Met3”), nSamples = c(1,2,1))

optim.alpha.stats(comm,, methods, c(0,0,1))
optim.alpha.stats(comm, tree, methods, c(0,1,1), runs = 100)
```

---

optim.beta *Optimization of beta diversity sampling protocols.*

Description

Optimization of beta diversity sampling protocols when different methods and multiple samples per method are available.

Usage

```
optim.beta(
  comm,
  tree,
  methods,
  base,
  seq = FALSE,
  abund = TRUE,
  runs = 1000,
  prog = TRUE
)
```

Arguments

`comm`	A samples x species x sites array, with either abundance or incidence data.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`methods`	A data.frame with the method names (1st column), number of samples per method (2nd column), base cost per method (3rd column, those costs that are fixed once a method is decided), and sample cost per method (those costs that add with each sample of the method, 4th column). If the last two columns are not provided base = 0 and sample = 1. The order of methods must be the same as in comm and the sum of the samples must be the same as nrow(comm).
`base`	Allows defining a base mandatory protocol from which to build upon (complementarity analysis). It should be a vector with length = number of methods.
`seq`	By default all combinations will be tested. If TRUE, a sequential approach will be taken, where methods are added based on the previous step. The method added will be the one providing the highest efficiency as quantified by the slope of the accumulation curve.
`abund`	A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis.
`runs`	Number of random permutations to be made to the sample order. Default is 1000.
`prog`	Present a text progress bar in the R console.

Details

Often, comparing differences between sites or the same site along time (i.e. measure beta diversity) it is not necessary to sample exhaustively. A minimum combination of samples targeting different sub-communities (that may behave differently) may be enough to perceive such differences, for example, for monitoring purposes. Cardoso et al. (in prep.) introduce and differentiate the concepts of alpha-sampling and beta-sampling. While alpha-sampling optimization implies maximizing local diversity sampled (Cardoso 2009), beta-sampling optimization implies minimizing differences in beta diversity values between partially and completely sampled communities. This function uses as beta diversity measures the Btotal, Brepl and Brich partitioning framework (Carvalho et al. 2012) and respective generalizations to PD and FD (Cardoso et al. 2014). PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A matrix of samples x methods (values being optimum number of samples per method). The last column is precision = (1 - average absolute difference from real beta).

References

Cardoso, P. (2009) Standardization and optimization of arthropod inventories - the case of Iberian spiders. Biodiversity and Conservation, 18, 3949-3962.

Cardoso, P., Rigal, F., Carvalho, J.C., Fortelius, M., Borges, P.A.V., Podani, J. & Schmera, D. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41, 749-761.

Cardoso, P., et al. (in prep.) Optimal inventorying and monitoring of taxon, phylogenetic and functional diversity.

Carvalho, J.C., Cardoso, P. & Gomes, P. (2012) Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. Global Ecology and Biogeography, 21, 760-771.

Examples

```
comm1 <- matrix(c(1,1,0,2,4,0,0,1,2,0,0,3), nrow = 4, ncol = 3, byrow = TRUE)
comm2 <- matrix(c(2,2,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm3 <- matrix(c(2,0,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm <- array(c(comm1, comm2, comm3), c(4,3,3))
colnames(comm) <- c("sp1","sp2","sp3")

methods <- data.frame(method = c("Met1","Met2","Met3"),
           nSamples = c(1,2,1), fixcost = c(1,1,2), varCost = c(1,1,1))
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels <- colnames(comm)

## Not run:
  optim.beta(comm,,methods)
  optim.beta(comm,,methods, seq = TRUE)
  optim.beta(comm, tree, methods)
  optim.alpha(comm,, methods = methods, seq = TRUE, base = c(0,1,1))

## End(Not run)
```

optim.beta.stats *Efficiency statistics for beta-sampling.*

Description

Average absolute difference between sampled and real beta diversity when using a given number of samples per method.

Usage

```
optim.beta.stats(comm, tree, methods, samples, abund = TRUE, runs = 1000)
```

Arguments

comm	A samples x species x sites array, with either abundance or incidence data.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
methods	A data.frame with the method names (1st column) and number of samples per method (2nd column). The order of methods must be the same as in comm and the sum of the samples must be the same as nrow(comm).
samples	A vector with the number of samples per method to test.
abund	A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis.
runs	Number of random permutations to be made to the sample order. Default is 1000.

Details

Different combinations of samples per method allow sampling different sub-communities. This function allows knowing the average absolute difference between sampled and real beta diversity for a given combination, for one or multiple sites simultaneously. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A single precision value = (1 - average absolute beta diversity difference value).

Examples

```
comm1 <- matrix(c(1,1,0,2,4,0,0,1,2,0,0,3), nrow = 4, ncol = 3, byrow = TRUE)
comm2 <- matrix(c(2,2,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm3 <- matrix(c(2,0,0,3,1,0,0,0,5,0,0,2), nrow = 4, ncol = 3, byrow = TRUE)
comm <- array(c(comm1, comm2, comm3), c(4,3,3))
colnames(comm) <- c("sp1","sp2","sp3")

tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels <- colnames(comm)

methods <- data.frame(method = c("Met1","Met2","Met3"), nSamples = c(1,2,1))

optim.beta.stats(comm,,methods, c(1,2,1)) #a complete sample will have 0 difference
optim.beta.stats(comm, tree, methods = methods, samples = c(0,1,1), runs = 100)
```

`optim.spatial` *Optimization of spatial sampling.*

Description

Optimization of sampling site distribution in space based on environmental (or other) variables.

Usage

```
optim.spatial(layers, n, latlong = TRUE, clusterMap = TRUE)
```

Arguments

`layers`	A SpatRaster object from package terra.
`n`	The number of intended sampling sites (clusters).
`latlong`	Boolean indicating whether latitude and longitude should be taken into account when clustering.
`clusterMap`	Boolean indicating whether to build a new raster with clusters.

Details

Optimizing the selection of sampling sites often requires maximizing the environmental diversity covered by them. One possible solution to this problem, here adopted, is performing a k-means clustering using environmental data and choosing the sites closest to the multidimensional environmental centroid of each cluster for sampling (Jimenez-Valverde & Lobo 2004)

Value

Either a matrix of cells x clusters (also indicating distance to centroid, longitude and latitude of each cell) or a list with such matrix plus the clusterMap.

References

Jimenez-Valverde, A., & Lobo, J. M. (2004) Un metodo sencillo para seleccionar puntos de muestreo con el objetivo de inventariar taxones hiperdiversos: el caso practico de las familias Araneidae y Thomisidae (Araneae) en la comunidad de Madrid, Espana. Ecologia, 18: 297-305.

originality *Phylogenetic/functional originality of species or individuals.*

Description

Average dissimilarity between a species or individual and all others in a community.

Usage

```
originality(comm, tree, distance, abund = FALSE, relative = FALSE)
```

Arguments

comm	A sites x species matrix, with either abundance or incidence data. If missing, the originality using the full tree or distance matrix is calculated.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
distance	A dist object representing the phylogenetic or functional distance between species.
abund	A boolean (T/F) indicating whether originality should be calculated per individual (T) or species (F).
relative	A boolean (T/F) indicating whether originality should be relative to the maximum distance between any two species in the tree or distance matrix.

Details

This is the originality measure of Pavoine et al. (2005) without replacement.

Value

A matrix of sites x species values.

References

Pavoine, S., Ollier, S. & Dufour, A.-B. (2005) Is the originality of a species measurable? Ecology Letters, 8: 579-586.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,1,1,1), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method="euclidean")
tree = hclust(distance)

originality(tree = tree)
originality(distance = distance)
originality(comm, tree)
originality(comm, tree, abund = TRUE)
originality(comm, tree, relative = TRUE)
```

`phylotree` *Taxonomic tree for 338 species of spiders (surrogate for phylogeny)*

Description

A dataset representing an approximation to the phylogenetic tree for 338 species of spiders captured in Portugal. The tree is based on the linnean hierarchy, with different suborders separated by 1 unit, families by 0.75, genera by 0.5 and species by 0.25.

Usage

```
data(phylotree)
```

Format

An hclust object with 338 species.

`r2` *Model R2.*

Description

Calculates R2 from the summed squared differences between observed and estimated values.

Usage

```
r2(obs, est = NULL, param = 0)
```

Arguments

`obs`	Either a model or a vector with observed values.
`est`	A vector with estimated values. Only used if obs is not a model.
`param`	Number of parameters in the model to calculate the adjusted R2 if > 0. If obs is a model param will be ignored and the number of parameters will be calculated from the model.

Details

Useful for models or functions that do not provide r2 values.

Value

The r2 value.

Examples

```
obs = c(1,4,5,6)
est = c(0,1,4,7)

#example using values
r2(obs, est)
r2(obs, est, param = 1)

#example using model
mod = lm(obs ~ est)
r2(mod)
summary(mod)$r.squared
r2(mod, param = 1)
summary(mod)$adj.r.squared
```

---

rao *Rao quadratic entropy.*

Description

Rao quadratic entropy for Phylogenetic or Functional richness.

Usage

```
rao(comm, tree, distance, raref = 0, runs = 100)
```

Arguments

`comm`	A sites x species matrix, with abundance data.
`tree`	A phylo or hclust object (used only for PD or FD).
`distance`	A dist object representing the phylogenetic or functional distance between species or alternatively a species x traits matrix or data.frame to calculate distances.
`raref`	An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.
`runs`	Number of resampling runs for rarefaction. If not specified, default is 100.

Details

Rao quadratic entropy (Rao, 1982) measures diversity based on the abundance of species and the dissimilarity between them.

Value

A matrix of sites x diversity values (either "Rao" OR "Mean, Median, Min, LowerCL, UpperCL and Max").

References

Rao, C.R. (1982). Diversity and dissimilarity coefficients: a unified approach. Theoretical Population Biology, 21: 24-43.

Examples

```
comm <- matrix(c(1,1,1,1,1,0,100,1,2,0), nrow = 2, ncol = 5, byrow = TRUE)
distance = dist(1:5)
rao(comm)
rao(comm, , distance)
rao(comm, hclust(distance), raref = 1)
```

`raster.alpha` *Maps of alpha diversity (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Observed alpha diversity using rasters of species distributions (presence/absence).

Usage

```
raster.alpha(layers, tree)
```

Arguments

`layers`	A SpatRaster object of species distributions from package terra.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

Details

TD is equivalent to species richness. Calculations of PD and FD are based on Faith (1992) and Petchey & Gaston (2002, 2006), which measure PD and FD of a community as the total branch length of a tree linking all species represented in such community. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in layers must be the same as in tree.

Value

A SpatRaster object representing richness in space.

References

Faith, D.P. (1992) Conservation evaluation and phylogenetic diversity. Biological Conservation, 61, 1-10.

Petchey, O.L. & Gaston, K.J. (2002) Functional diversity (FD), species richness and community composition. Ecology Letters, 5, 402-411.

Petchey, O.L. & Gaston, K.J. (2006) Functional diversity: back to basics and looking forward. Ecology Letters, 9, 741-758.

Examples

```
sp1 <- terra::rast(matrix(c(NA,1,1,1,1,0,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
sp2 <- terra::rast(matrix(c(0,0,0,0,1,1,1,1,1), nrow = 3, ncol = 3, byrow = TRUE))
sp3 <- terra::rast(matrix(c(0,0,0,1,1,1,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
spp <- c(sp1, sp2, sp3)
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels = c("Sp1", "Sp2", "Sp3")
names(spp) = tree$labels
raster.alpha(spp)
raster.alpha(spp, tree)
```

`raster.beta` *Maps of beta diversity (Taxon, Phylogenetic or Functional Diversity - TD, PD, FD).*

Description

Observed beta diversity using rasters of species distributions (presence/absence or abundance).

Usage

```
raster.beta(layers, tree, func = "jaccard", neighbour = 8, abund = FALSE)
```

Arguments

`layers`	A SpatRaster object of species distributions from package terra.
`tree`	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
`func`	Partial match indicating whether the Jaccard or Soerensen family of beta diversity measures should be used. If not specified, default is Jaccard.
`neighbour`	Either 8 (default) or 4 cells considered to calculate beta diversiy of each focal cell.
`abund`	A boolean (T/F) indicating whether abundance data should be used (TRUE) or converted to incidence (FALSE) before analysis.

Details

The beta diversity measures used here follow the partitioning framework independently developed by Podani & Schmera (2011) and Carvalho et al. (2012) and later expanded to PD and FD by Cardoso et al. (2014), where Btotal = Brepl + Brich. Btotal = total beta diversity, reflecting both species replacement and loss/gain; Brepl = beta diversity explained by replacement of species alone; Brich = beta diversity explained by species loss/gain (richness differences) alone. PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric). The path to the root of the tree is always included in calculations of PD and FD. The number and order of species in layers must be the same as in tree.

Value

A SpatRaster object with three layers representing Btotal, Brepl and Brich in space.

References

Cardoso, P., Rigal, F., Carvalho, J.C., Fortelius, M., Borges, P.A.V., Podani, J. & Schmera, D. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41, 749-761.

Carvalho, J.C., Cardoso, P. & Gomes, P. (2012) Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns. Global Ecology and Biogeography, 21, 760-771.

Gotelli, N.J. & Colwell, R.K. (2001) Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. Ecology Letters, 4, 379-391.

Podani, J. & Schmera, D. (2011) A new conceptual and methodological framework for exploring and explaining pattern in presence-absence data. Oikos, 120, 1625-1638.

Examples

```
sp1 <- terra::rast(matrix(c(NA,1,1,1,1,0,1,1,0), nrow = 3, ncol = 3, byrow = TRUE))
sp2 <- terra::rast(matrix(c(0,0,0,1,1,1,1,1,1), nrow = 3, ncol = 3, byrow = TRUE))
sp3 <- terra::rast(matrix(c(0,0,0,1,1,1,1,1,0), nrow = 3, ncol = 3, byrow = TRUE))
spp <- c(sp1, sp2, sp3)
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels = c("Sp1", "Sp2", "Sp3")
names(spp) = tree$labels
raster.beta(spp)
raster.beta(spp, tree)
```

`raster.dispersion` *Maps of phylogenetic/functional dispersion of species or individuals.*

Description

Average dissimilarity between any two species or individuals randomly chosen in a community using rasters of species distributions (presence/absence or abundance).

Usage

```
raster.dispersion(
  layers,
  tree,
  distance,
  func = "originality",
  abund = FALSE,
  relative = FALSE
)
```

Arguments

layers	A SpatRaster object of species distributions from package terra.
tree	A phylo or hclust object or alternatively a species x traits matrix or data.frame to build a functional tree.
distance	A dist object representing the phylogenetic or functional distance between species.
func	Calculate dispersion using originality (default), uniqueness or contribution.
abund	A boolean (T/F) indicating whether dispersion should be calculated using individuals (T) or species (F).
relative	A boolean (T/F) indicating whether dispersion should be relative to the maximum distance between any two species in the tree or distance matrix.

Details

If abundance data is used and a tree is given, dispersion is the quadratic entropy of Rao (1982). If abundance data is not used but a tree is given, dispersion is the phylogenetic dispersion measure of Webb et al. (2002). Note that cells with less than two species cannot have dispersion values.

Value

A SpatRaster object representing dispersion in space.

References

Rao, C.R. (1982) Diversity and dissimilarity coefficients: a unified approach. Theoretical Population Biology, 21: 24-43.

Webb, C.O., Ackerly, D.D., McPeek, M.A. & Donoghue, M.J. (2002) Phylogenies and community ecology. Annual Review of Ecology and Systematics, 33: 475-505.

Examples

```
sp1 <- terra::rast(matrix(c(NA,1,1,1,1,0,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
sp2 <- terra::rast(matrix(c(0,0,0,0,1,1,1,1,1), nrow = 3, ncol = 3, byrow = TRUE))
sp3 <- terra::rast(matrix(c(0,0,0,1,1,1,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
spp <- c(sp1, sp2, sp3)
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels = c("Sp1", "Sp2", "Sp3")
names(spp) = tree$labels
raster.dispersion(spp, tree)
```

raster.evenness *Maps of phylogenetic/functional evenness of species or individuals.*

Description

Regularity of distance and abundance between any two species in a community using rasters of species distributions (presence/absence or abundance).

Usage

```
raster.evenness(
  layers,
  tree,
  distance,
  method = "expected",
  func = "camargo",
  abund = TRUE
)
```

Arguments

`layers`	A SpatRaster object of species distributions from package terra.
`tree`	A phylo or hclust object or alternatively a species x traits matrix or data.frame to build a functional tree.
`distance`	A dist object representing the phylogenetic or functional distance between species.
`method`	Calculate dispersion using "expected" values (default) or values based on "contribution" of species to the tree.
`func`	Calculate dispersion using "Camargo" (1993; default) or "Bulla" (1994) index.
`abund`	A boolean (T/F) indicating whether evenness should be calculated using abundance data.

Details

If no tree or distance is provided the result is the original index of Bulla with correction. Note that cells with less than two species cannot have evenness values.

Value

A SpatRaster object representing evenness in space.

References

Bulla, L. (1994) An index of evenness and its associated diversity measure. Oikos, 70: 167-171.

Camargo, J.A. (1993) Must dominance increase with the number of subordinate species in competitive interactions? Journal of Theoretical Biology, 161: 537-542.

Examples

```
sp1 <- terra::rast(matrix(c(NA,1,1,1,1,0,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
sp2 <- terra::rast(matrix(c(0,0,0,0,1,1,1,1,1), nrow = 3, ncol = 3, byrow = TRUE))
sp3 <- terra::rast(matrix(c(0,0,0,1,1,1,0,0,0), nrow = 3, ncol = 3, byrow = TRUE))
spp <- c(sp1, sp2, sp3)
tree <- hclust(dist(c(1:3), method="euclidean"), method="average")
tree$labels = c("Sp1", "Sp2", "Sp3")
names(spp) = tree$labels
raster.evenness(spp)
raster.evenness(spp, tree)
```

---

sad *Species-abundance distribution (SAD).*

Description

Fits the SAD to community abundance data, also using trees and with possible rarefaction.

Usage

```
sad(comm, tree, octaves = TRUE, scale = FALSE, raref = 0, runs = 100)
```

Arguments

`comm` Either a vector with the abundance per species, or a sites x species matrix.

`tree` A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.

`octaves` a boolean indicating whether octaves should be calculated.

`scale` scale y-axis to sum 1.

`raref` An integer specifying the number of individuals for rarefaction (individual based). If raref < 1 no rarefaction is made. If raref = 1 rarefaction is made by the minimum abundance among all sites. If raref > 1 rarefaction is made by the abundance indicated. If not specified, default is 0.

`runs` Number of resampling runs for rarefaction. If not specified, default is 100.

Details

The Species Abundance Distribution describes the commonness and rarity in ecological systems. It was recently expanded to accomodate phylegenetic and functional differences between species (Matthews et al., subm.). Classes defined as n = 1, 2-3, 4-7, 8-15, ... Rarefaction allows comparison of sites with different total abundances.

Value

A vector or matrix with the different values per class per community.

References

Matthews et al. (subm.) Phylogenetic and functional dimensions of the species abundance distribution.

Examples

```
comm1 <- c(20,1,3,100,30)
comm2 <- c(1,2,12,0,45)
comm <- rbind(comm1, comm2)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
sad(comm1)
sad(comm)
sad(comm, octaves = FALSE)
sad(comm, tree, scale = TRUE)
sad(comm, raref = 1)
```

---

sar *Species-area relationship (SAR).*

Description

Fits and compares several of the most supported models for the species (or PD, or FD) -area relationship.

Usage

```
sar(comm, tree, area)
```

Arguments

comm	Either a vector with the diversity values per site, or a sites x species matrix.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree (used only to fit the PD or FD-area relationships, requires comm to be a sites x species matrix).
area	A vector with the area per site.

Details

Larger areas (often islands) usually carry more species. Several formulas were proposed in the past to describe this relationship (Arrhenius 1920, 1921; Gleason 1922). Recently, the same approach began to be used for other measures of diversity, namely phylogenetic (PD) and functional (FD) diversity (Whittaker et al. 2014). The function compares some of the most commonly used and theoretically or empirically suported models. The relationships for PD and FD are calculated based on a tree (hclust or phylo object, no need to be ultrametric).

Value

A matrix with the different model parameters and explanatory power.

References

Arrhenius, O. (1920) Distribution of the species over the area. Meddelanden fran Vetenskapsakademiens Nobelinstitut, 4: 1-6.

Arrhenius, O. (1921) Species and area. Journal of Ecology, 9: 95-99.

Gleason, H.A. (1922) On the relation between species and area. Ecology, 3: 158-162.

Whittaker, R.J., Rigal, F., Borges, P.A.V., Cardoso, P., Terzopoulou, S., Casanoves, F., Pla, L., Guilhaumon, F., Ladle, R. & Triantis, K.A. (2014) Functional biogeography of oceanic islands and the scaling of functional diversity in the Azores. Proceedings of the National Academy of Sciences USA, 111: 13709-13714.

Examples

```
div <- c(1,2,3,4,4)
comm <- matrix(c(2,0,0,0,3,1,0,0,2,4,5,0,1,3,2,5,1,1,1,1), nrow = 5, ncol = 4, byrow = TRUE)
tree <- hclust(dist(c(1:4), method="euclidean"), method="average")
area <- c(10,40,80,160,160)
sar(div,,area)
sar(comm,,area)
sar(comm,tree,area)
```

ses   *Standard Effect Size.*

Description

Calculates the standard effect size from observed and estimated values.

Usage

```
ses(obs, est, param = TRUE, p = TRUE)
```

Arguments

`obs` A single observed value.

`est` A vector with estimated values.

`param` Value is calculated with parametric or non-parametric method. Because standardized effect sizes may lead to biased conclusions if null values show an asymmetric distribution or deviate from normality, non-parametric effect sizes use probit transformed p-values (Lhotsky et al., 2016).

`p` Boolean indicating whether the p-value should be returned.

Value

The ses value or a vector with ses and p-value.

References

Lhotsky et al. (2016) Changes in assembly rules along a stress gradient from open dry grasslands to wetlands. Journal of Ecology, 104: 507-517.

Examples

```
est = rnorm(1000, 500, 100)

ses(100, est)
ses(100, est, param = FALSE)
ses(500, est)
ses(500, est, param = FALSE)
ses(900, est, p = TRUE)
ses(900, est, param = FALSE, p = TRUE)
```

`sim.plot` *Plots of simulated species spatial distributions.*

Description

Plots individuals from artificial communities with given SAD and spatial clustering.

Usage

```
sim.plot(comm, sad = FALSE, s = 0)
```

Arguments

`comm`	artificial community data from function sim.spatial.
`sad`	boolean indicating if the SAD plot should also be shown. Default is FALSE.
`s`	number of species to plot simultaneously. Default is the number of species in comm.

Details

Function useful for visualizing the results of sim.spatial.

Examples

```
comm <- sim.spatial(1000, 24)
sim.plot(comm)
sim.plot(comm, sad = TRUE)
sim.plot(comm, s = 9)
```

sim.sad *Simulation of species abundance distributions (SAD).*

Description

Creates artificial communities following given SADs.

Usage

```
sim.sad(n, s, sad = "lognormal", sd = 1)
```

Arguments

n	total number of individuals.
s	number of species.
sad	The SAD distribution type (lognormal, uniform, broken stick or geometric). Default is lognormal.
sd	The standard deviation of lognormal distributions. Default is 1.

Details

Species Abundance Distributions may take a number of forms. A lognormal SAD probably is the most supported by empirical data, but we include other common types useful for testing multiple algorithms including several of the functions in BAT.

Value

A matrix of species x abundance per species.

Examples

```
comm1 <- sim.sad(10000, 100)
comm2 <- sim.sad(10000, 100, sd = 2)
comm3 <- sim.sad(10000, 100, sad = "uniform")
par(mfrow=c(1,3))
hist(log(comm1$Freq))
hist(log(comm2$Freq))
hist(log(comm3$Freq))
```

sim.sample *Simulation of sampling from artificial communities.*

Description

Simulates a sampling process from artificial communities.

Usage

```
sim.sample(comm, cells = 100, samples = 0)
```

Arguments

comm	simulated community data from function sim.spatial.
cells	number of cells to divide the simulated space into. Default is 100.
samples	number of samples (cells) to randomly extract. Default is the number of cells (the entire community).

Details

The space will be divided in both dimensions by sqrt(cells).

Function useful for simulating sampling processes from the results of sim.spatial.

May be used as direct input to other functions (e.g. alpha, alpha.accum, beta, beta.accum) to test the behavior of multiple descriptors and estimators.

Value

A matrix of samples x species (values are abundance per species per sample).

Examples

```
comm <- sim.spatial(1000, 10)
sim.sample(comm)
sim.sample(comm, cells = 10, samples = 5)
```

sim.spatial *Simulation of species spatial distributions.*

Description

Creates artificial communities with given SAD and spatial clustering.

Usage

```
sim.spatial(
  n,
  s,
  sad = "lognormal",
  sd = 1,
  distribution = "aggregated",
  clust = 1
)
```

Arguments

n	total number of individuals.
s	number of species.
sad	The SAD distribution type (lognormal, uniform, broken stick or geometric). Default is lognormal.
sd	The standard deviation of lognormal distributions. Default is 1.
distribution	The spatial distribution of individual species populations (aggregated, random, uniform or gradient). Default is aggregated.
clust	The clustering parameter if distribution is either aggregated or gradient (higher values create more clustered populations). Default is 1.

Details

The spatial distribution of individuals of given species may take a number of forms. Competitive exclusion may cause overdispersion, specific habitat needs or cooperation may cause aggregation and environmental gradients may cause abundance gradients.

Value

A matrix of individuals x (species, x coords and y coords).

Examples

```
par(mfrow = c(3 ,3))
comm = sim.spatial(100, 9, distribution = "uniform")
for(i in 1:9){
sp <- comm[comm[1] == paste("Sp", i, sep = ""), ]
plot(sp$x, sp$y, main = paste("Sp", i), xlim = c(0,1), ylim = c(0,1))
}
```

sim.tree *Simulation of phylogenetic or functional tree.*

Description

Simulates a random tree.

Usage

```
sim.tree(s, m = 100)
```

Arguments

s	number of species.
m	a structural parameter defining the average difference between species. Default is 100. Lower numbers create trees dominated by increasingly similar species, higher numbers by increasingly dissimilar species.

Details

A very simple tree based on random genes/traits.

Value

An hclust object.

Examples

```
tree <- sim.tree(10)
plot(as.dendrogram(tree))
tree <- sim.tree(100,10)
plot(as.dendrogram(tree))
tree <- sim.tree(100,1000)
plot(as.dendrogram(tree))
```

slope *Slope of accumulation curves.*

Description

This is similar to the first derivative of the curves at each of its points.

Usage

```
slope(accum)
```

Arguments

`accum` A matrix resulting from the alpha.accum or beta.accum functions (sampling units x diversity values).

Details

Slope is the expected gain in diversity when sampling a new individual. The slope of an accumulation curve, of either observed or estimated diversity, allows verifying if the asymptote has been reached (Cardoso et al. 2011). This is an indication of either the completeness of the inventory (low final slopes of the observed curve indicate high completeness) or reliability of the estimators (stability of the slope around a value of 0 along the curve indicates reliability).

Value

A matrix of sampling units x slope values.

References

Cardoso, P., Pekar, S., Jocque, R. & Coddington, J.A. (2011) Global patterns of guild composition and functional diversity of spiders. PLoS One, 6, e21710.

Examples

```
comm1 <- matrix(c(2,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,0,2,2), nrow = 4, ncol = 5, byrow = TRUE)
comm2 <- matrix(c(1,1,0,0,0,0,2,1,0,0,0,0,2,1,0,0,0,0,2,1), nrow = 4, ncol = 5, byrow = TRUE)
tree <- hclust(dist(c(1:5), method="euclidean"), method="average")
acc.alpha = alpha.accum(comm1)
slope(acc.alpha)
acc.beta = beta.accum(comm1, comm2, tree)
slope(acc.beta)
```

`standard` *Standardize variables.*

Description

Standardize (or normalize) variables in different ways.

Usage

```
standard(trait, method = "standard", convert = NULL)
```

Arguments

`trait` A species x traits matrix or data.frame.

`method` One of "standard" (standardize to mean = 0 and sd = 1, i.e., use z-score), "range" (rescale with range 0-1), or "rank" (rescale with range 0-1 after ranking).

`convert` A vector of column numbers to be standardized. If NULL all will be standardized.

Details

Standardizing values allows to directly compare variables of interest with inherently different ranges, avoiding artificial distortions of distances between observations.

Value

A matrix with variables standardized.

Examples

```
trait = data.frame(body = c(20,40,60,30,10), beak = c(NA,4,6,3,1))
standard(trait)
standard(trait, method = "range")
standard(trait, method = "rank")
```

tree.build *Build functional tree.*

Description

Builds a functional tree from trait or distance data.

Usage

```
tree.build(
  trait,
  distance = "gower",
  func = "nj",
  fs = 0,
  convert = NULL,
  weight = NULL,
  root = NULL
)
```

Arguments

`trait` A species x traits matrix or data.frame or, alternatively, a dist object.

`distance` One of "gower" or "euclidean". Not used if trait is already a dist object.

`func` One of "upgma", "mst", "nj", "bionj" or "best".

`fs` Only used for func = "nj" OR "bionj". Argument s of the agglomerative criterion: it is coerced as an integer and must at least equal to one.

`convert` A vector of column numbers, usually categorical variables, to be converted to dummy variables. Not used if trait is already a dist object.

`weight` A vector of column numbers with weights for each variable. Its length must be equal to the number of columns in trait. Not used if trait is already a dist object.

`root` A numeric or character specifying the functional outgroup to root the tree.

Details

The tree will be built using one of four algorithms after traits are dummyfied (if needed) and standardized (always): If func = "upgma" uses average linkage clustering (UPGMA, Cardoso et al. 2014). If func = "mst" uses minimum spanning trees, equivalent to single linkage clustering (Gower & Ross 1969). If func = "nj" uses the original neighbor-joining algorithm of Saitou & Nei (1987). If func = "bionj" uses the modified neighbor-joining algorithm of Gascuel (1997). Any of the neighbor-joining options is usually preferred as they keep distances between species better than UPGMA or MST (Cardoso et al. subm.). If func = "best", chooses the best of the options above based on maximum tree.quality values. If NJ trees are built, the root will be set at the node closest to the midpoint between the two most dissimilar species in the tree or, if root not NULL, at the node provided in parameter root (Podani et al. 2000). Gower distance (Pavoine et al. 2009) allows continuous, ordinal, categorical or binary variables, with possible weighting. NAs are allowed as long as each pair of species has at least one trait value in common. For fs > 0 even if this condition is not met the Q* criterion by Criscuolo & Gascuel (2008) is used to fill missing data. If convert is given the algorithm will convert these column numbers to dummy variables. Otherwise it will convert all columns with factors or characters as values.

Value

A phylo object representing a functional tree.

References

Cardoso et al. (2014) Partitioning taxon, phylogenetic and functional beta diversity into replacement and richness difference components. Journal of Biogeography, 41: 749-761.

Cardoso et al. (subm.) Using neighbor-joining trees for functional diversity analyses.

Criscuolo & Gascuel (2008) Fast NJ-like algorithms to deal with incomplete distance matrices. BMC Bioinformatics, 9: 166.

Gascuel (1997) BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Molecular Biology and Evolution, 14: 685–695.

Gower & Ross (1969) Minimum spanning trees and single linkage cluster analysis. Journal of the Royal Statistical Society, 18: 54-64.

Pavoine et al. (2009) On the challenge of treating various types of variables: application for improving the measurement of functional diversity. Oikos, 118: 391-402.

Podani et al. (2000) Additive trees in the analysis of community data. Community Ecology, 1, 33–41.

Saitou & Nei (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular Biology and Evolution, 4, 406–425.

Examples

```
trait = data.frame(body = c(NA,2,3,4,4), beak = c(1,1,1,1,2))
plot(tree.build(trait))
plot(tree.build(trait, func = "bionj", fs = 1, weight = c(1, 0)), "u")
plot(tree.build(trait, func = "best", root = 4))
```

tree.quality *Quality of tree.*

Description

Assess the quality of a functional tree.

Usage

```
tree.quality(distance, tree)
```

Arguments

`distance`	A dist object representing the initial distances between species.
`tree`	A phylo or hclust object.

Details

The algorithm calculates the inverse of mean squared deviation between initial and cophenetic distances (Maire et al. 2015) after standardization of all values between 0 and 1 for simplicity of interpretation. A value of 1 corresponds to maximum quality of the functional representation. A value of 0 corresponds to the expected value for a star tree, where all pairwise distances are 1.

Value

A single value of quality.

References

Maire et al. (2015) How many dimensions are needed to accurately assess functional diversity? A pragmatic approach for assessing the quality of functional spaces. Global Ecology and Biogeography, 24: 728:740.

Examples

```
trait = data.frame(body = c(1,2,3,4,4), beak = c(1,1,1,1,2))
distance = gower(trait)

tree = tree.build(trait)
tree.quality(distance, tree)

tree = tree.build(trait, func = "bionj")
tree.quality(distance, tree)

tree = tree.build(trait, func = "upgma")
tree.quality(distance, tree)

tree = tree.build(trait, func = "mst")
tree.quality(distance, tree)
```

```
tree = tree.build(trait, func = "best")

distance1 = distance
distance1[] = 1
tree = hclust(distance1)
tree.quality(distance, tree)
```

tree.zero *Convert negative branches of tree.*

Description

Converts negative branch lengths of any tree to zero.

Usage

```
tree.zero(tree)
```

Arguments

`tree`	A phylo object.

Details

Converts branches with negative values to zero while shortening only the two branches immediately below it by the same absolute amount to ensure the tree remains with tips at same distances and there are no polytomies.

Value

A phylo object.

Examples

```
par(mfrow = c(1,2))
tree <- ape::read.tree(text='(((A:3, B:3):1,
(G:6, (H:5, I:5):1):-2):3, ((C:1, D:1):2, (E:4, F:4):-1):4);')
plot(tree)

tree = tree.zero(tree)
plot(tree)
```

uniqueness *Phylogenetic/functional uniqueness of species.*

Description

Dissimilarity between each species and the single closest in a community.

Usage

```
uniqueness(comm, tree, distance, relative = FALSE)
```

Arguments

comm	A sites x species matrix, with either abundance or incidence data. If missing, the uniqueness using the full tree or distance matrix is calculated.
tree	A phylo or hclust object (used only for PD or FD) or alternatively a species x traits matrix or data.frame to build a functional tree.
distance	A dist object representing the phylogenetic or functional distance between species.
relative	A boolean (T/F) indicating whether uniqueness should be relative to the maximum distance between any two species in the tree or distance matrix.

Details

This is equivalent to the originality measure of Mouillot et al. (2013).

Value

A matrix of sites x species values.

References

Mouillot, D., Graham, N.A., Villeger, S., Mason, N.W. & Bellwood, D.R. (2013) A functional approach reveals community responses to disturbances. Trends in Ecology and Evolution, 28: 167-177.

Examples

```
comm <- matrix(c(1,2,0,0,0,1,1,0,0,0,0,2,2,0,0,0,0,1,0,1), nrow = 4, byrow = TRUE)
distance <- dist(c(1:5), method="euclidean")
tree <- hclust(distance, method="average")

uniqueness(tree = tree)
uniqueness(distance = distance)
uniqueness(comm, tree)
```

Index