Package: divvy (via r-universe)

July 23, 2024

Title Spatial Subsampling of Biodiversity Occurrence Data

Version 1.0.0.9000

Depends R (>= 4.0)

Description Divide taxonomic occurrence data into geographic regions of fair comparison, with three customisable methods to standardise area and extent. Calculate common biodiversity and range-size metrics on subsampled data. Background theory and practical considerations for the methods are described in Antell and others (2023) <doi:10.31223/X5997Z>.

License GPL (>= 3)

Encoding UTF-8

Imports iNEXT (>= 3.0.0), Rdpack, sf, terra, units, vegan

RdMacros Rdpack

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

LazyData true

Suggests knitr, rnaturalearth, rnaturalearthdata, ggplot2, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://gawainantell.github.io/divvy/, https://github.com/GawainAntell/divvy

BugReports https://github.com/GawainAntell/divvy/issues

Repository https://gawainantell.r-universe.dev

RemoteUrl https://github.com/gawainantell/divvy

RemoteRef HEAD

RemoteSha 69a3a5bcb2ffb54903c67a912f2f5370d98aedb4

Contents

`bandit` *Rarefy localities within latitudinal bands*

Description

`bandit` subsamples spatial point data to a specified number of sites within bins of equal latitude

Usage

```
bandit(
  dat,
  xy,
  iter,
  nSite,
  bin,
  centr = FALSE,
  absLat = FALSE,
  maxN = 90,
  maxS = -90,
  crs = "epsg:4326",
  output = "locs"
)
```

Arguments

`dat`	A `data.frame` or `matrix` containing the coordinate columns `xy` and any associated variables, e.g. taxon names.
`xy`	A vector of two elements, specifying the name or numeric position of columns in `dat` containing coordinates, e.g. longitude and latitude. Coordinates for any shared sampling sites should be identical, and where sites are raster cells, coordinates are usually expected to be cell centroids.
`iter`	The number of times to subsample localities within **each** latitudinal band.

nSite	The quota of unique locations to include in each subsample.
bin	A positive numeric value for latitudinal band width, in degrees.
centr	Logical: should a bin center on and cover the equator (TRUE) or should the equator mark the boundary between the lowest-latitude northern and southern bins (FALSE, default)? Ignored if absLat = TRUE.
absLat	Logical: should only the absolute values of latitude be evaluated? If absLat = TRUE, centr argument is ignored.
maxN	Optional argument to specify the northmost limit for subsampling, if less than 90 degrees.
maxS	Optional argument to specify the southmost limit for subsampling, if not -90 degrees. Should be a negative value if in the southern hemisphere.
crs	Coordinate reference system as a GDAL text string, EPSG code, or object of class crs. Default is latitude-longitude (EPSG:4326).
output	Whether the returned data should be two columns of subsample site coordinates (output = 'locs') or the subset of rows from dat associated with those coordinates (output = 'full').

Details

bandit() rarefies the number of spatial sites within latitudinal ranges of specified bin width. (Compare with cookies() and clustr(), which spatially subsample to a specified extent without regard to latitudinal position.) Cases where it could be appropriate to control for latitudinal spread of localities include characterisations of latitudinal diversity gradients (e.g. Marcot 2016) or comparisons of ecosystem parameters that covary strongly with latitude (e.g. diversity in reefal vs. non-reefal habitats). Note that the total surface area of the Earth within equal-latitudinal increments decreases from the equator towards the poles; bandit() standardises only the amount of sites/area encompassed by each subsample, not the total area that could have been available for species to inhabit.

As with all divvy subsampling functions, sites within a given regional/latitudinal subsample are selected without replacement.

To calculate an integer number of degrees into which a given latitudinal range divides evenly, the palaeoverse package (v 1.2.1) provides the palaeoverse::lat_bins() function with argument fit = TRUE.

Value

A list of subsamples, each a data.frame containing coordinates of subsampled localities (if output = 'locs') or the subset of occurrences from dat associated with those coordinates (if output = 'full'). The latitudinal bounds of each subsample are specified by its name in the list. If there are too few localities in a given interval to draw a subsample, that interval is omitted from output.

References

Allen BJ, Wignall PB, Hill DJ, Saupe EE, Dunhill AM (2020). "The latitudinal diversity gradient of tetrapods across the Permo–Triassic mass extinction and recovery interval." *Proceedings of the Royal Society B*, **287**(1929), 20201125. doi:10.1098/rspb.2020.1125.

Marcot JD, Fox DL, Niebuhr SR (2016). "Late Cenozoic onset of the latitudinal diversity gradient of North American mammals." *Proceedings of the National Academy of Sciences*, **113**(26), 7189-7194. doi:10.1073/pnas.1524750113.

See Also

cookies()

clustr()

Examples

```
# load bivalve occurrences to rasterise
library(terra)
data(bivalves)

# initialise Equal Earth projected coordinates
rWorld <- rast()
prj <- 'EPSG:8857'
rPrj <- project(rWorld, prj, res = 200000) # 200,000m is approximately 2 degrees

# coordinate column names for the current and target coordinate reference system
xyCartes <- c('paleolng','paleolat')
xyCell   <- c('centroidX','centroidY')

# project occurrences and retrieve cell centroids in new coordinate system
llOccs <- vect(bivalves, geom = xyCartes, crs = 'epsg:4326')
prjOccs <- project(llOccs, prj)
cellIds <- cells(rPrj, prjOccs)[,'cell']
bivalves[, xyCell] <- xyFromCell(rPrj, cellIds)

# subsample 20 equal-area sites within 10-degree bands of absolute latitude
n <- 20
reps <- 100
set.seed(11)
bandAbs <- bandit(dat = bivalves, xy = xyCell,
                  iter = reps, nSite = n, output = 'full',
                  bin = 10, absLat = TRUE,
                  crs = prj
)
head(bandAbs[[1]]) # inspect first subsample
names(bandAbs)[1] # degree interval (absolute value) of first subsample
#> [1] "[10,20)"
unique(names(bandAbs)) # all intervals containing sufficient data
#> [1] "[10,20)" "[20,30)" "[30,40)" "[40,50)"
# note insufficient coverage to subsample at equator or above 50 degrees

# subsample 20-degree bands, where central band spans the equator
# (-10 S to 10 N latitude), as in Allen et al. (2020)
# (An alternative, finer-grain way to divide 180 degrees evenly into an
# odd number of bands would be to set 'bin' = 4.)
bandCent <- bandit(dat = bivalves, xy = xyCell,
                   iter = reps, nSite = n, output = 'full',
```

```
                  bin = 20, centr = TRUE, absLat = FALSE,
                  crs = prj
)
unique(names(bandCent)) # all intervals containing sufficient data
#> [1] "[-50,-30)" "[10,30)" "[30,50)"
```

`bivalves` *Paleobiology Database occurrences of Pliocene fossil bivalves*

Description

A dataset containing the (palaeo)coordinates and genus identifications of 8,000 marine bivalves from the Pliocene (ca. 5.3-2.6 Ma). Records with uncertain or unaccepted taxonomic names, non-marine palaeo-environments, or missing coordinates are excluded from the original download (24 June 2022).

Usage

```
bivalves
```

Format

A data frame with 8095 rows and 9 variables:

genus Latin genus identification. Subgenera are not elevated.

paleolng, paleolat Coordinates of an occurrence, rotated to its palaeogeographic location with the tectonic plate model of GPlates

collection_no, reference_no Unique identifiers for the collection and published reference containing the occurrence

environment One of 23 marine environment categories

max_ma, min_ma Bounds of the age estimate for an occurrence

accepted_name Original identification, including subgenus and species epithet if applicable, according to the latest PBDB accepted taxonomy at time of download

Source

https://paleobiodb.org/

`classRast` *Convert point environment data to a raster of majority-environment classes*

Description

Given point occurrences of environmental categories, `classRast` generates a raster grid with cell values specifying the majority environment therein.

Usage

```
classRast(grid, dat = NULL, xy, env, cutoff)
```

Arguments

`grid`	A `SpatRaster` to use as a template for the resolution, extent, and coordinate reference system of the returned object. Values can be empty.
`dat`	Either a `data.frame` or `matrix` for which `xy` and `env` are column names, or an empty argument.
`xy`	A vector specifying the name or numeric position of columns in `dat` containing coordinates, if `dat` is supplied, or a 2-column `data.frame` or `matrix` of coordinate values.
`env`	The name or numeric position of the column in `dat` containing a categorical environmental variable, if `dat` is supplied, or a vector of environmental values.
`cutoff`	The (decimal) proportion of incidences of an environmental category above which a cell will be assigned as that category. `cutoff` must be greater than 0.5.

Details

The `cutoff` threshold is an inclusive bound: environmental incidence proportions greater than or equal to the `cutoff` will assign cell values to the majority environmental class. For instance, if category A represents 65% of occurrences in a cell and `cutoff = 0.65`, the returned value for the cell will be A. If no single category in a cell meets or exceeds the representation necessary to reach the given `cutoff`, the value returned for the cell is `indet.`, indeterminate. Cells lacking environmental occurrences altogether return `NA` values.

The `env` object can contain more than two classes, but in many cases it will be less likely for any individual class to attain an absolute majority the more finely divided classes are. For example, if there are three classes, A, B, and C, with relative proportions of 20%, 31%, and 49%, the cell value will be returned as `indet.` because no single class can attain a `cutoff` above 50%, despite class C having the largest relative representation.

Missing environment values in the point data should be coded as `NA`, not e.g. `'unknown'`. `classRast()` ignores `NA` occurrences when tallying environmental occurrences against the `cutoff`. However, `NA` occurrences still count when determining `NA` status of cells in the raster: a cell containing occurrences of only `NA` value is classified as `indet.`, not `NA`. That is, any grid cell encompassing original point data is non-`NA`.

Antell and others (2020) set a `cutoff` of 0.8, based on the same threshold Nürnberg and Aberhan (2013) used to classify environmental preferences for taxa.

The coordinates associated with points should be given with respect to the same coordinate reference system (CRS) of the target raster grid, e.g. both given in latitude-longitude, Equal Earth projected coordinates, or other CRS. The CRS of a `SpatRaster` object can be retrieved with `terra::crs()` (with the optional but helpful argument `describe = TRUE`).

Value

A raster of class `SpatRaster` defined by the `terra` package

References

Antell GT, Kiessling W, Aberhan M, Saupe EE (2020). "Marine biodiversity and geographic distributions are independent on large scales." *Current Biology*, **30**(1), 115-121. doi:10.1016/j.cub.2019.10.065.

Nürnberg S, Aberhan M (2013). "Habitat breadth and geographic range predict diversity dynamics in marine Mesozoic bivalves." *Paleobiology*, **39**(3), 360-372. doi:10.1666/12047.

Examples

```
library(terra)
# work in Equal Earth projected coordinates
prj <- 'EPSG:8857'
# generate point occurrences in a small area of Northern Africa
n <- 100
set.seed(5)
x <- runif(n,  0, 30)
y <- runif(n, 10, 30)
# generate an environmental variable with a latitudinal gradient
# more habitat type 0 (e.g. rock) near equator, more 1 (e.g. grassland) to north
env <- rbinom(n, 1, prob = (y-10)/20)
env[env == 0] <- 'rock'
env[env == 1] <- 'grass'
# units for Equal Earth are meters, so if we consider x and y as given in km,
x <- x * 1000
y <- y * 1000
ptsDf <- data.frame(x, y, env)
# raster for study area at 5-km resolution
r <- rast(resolution = 5*1000, crs = prj,
          xmin = 0, xmax = 30000, ymin = 10000, ymax = 30000)

binRast <- classRast(grid = r, dat = ptsDf, xy = c('x', 'y'),
                     env = 'env', cutoff = 0.6)
binRast

# plot environment classification vs. original points
plot(binRast, col = c('lightgreen', 'grey60', 'white'))
points(ptsDf[env=='rock', ], pch = 16, cex = 1.2) # occurrences of given habitat
points(ptsDf[env=='grass',], pch =  1, cex = 1.2)
```

```
# classRast can also accept more than 2 environmental classes:

# add a 3rd environmental class with maximum occurrence in bottom-left grid cell
newEnv <- data.frame('x' = rep(0,       10),
                     'y' = rep(10000,   10),
                     'env' = rep('new', 10))
ptsDf <- rbind(ptsDf, newEnv)
binRast <- classRast(grid = r, dat = ptsDf, xy = c('x', 'y'),
                     env = 'env', cutoff = 0.6)
plot(binRast, col = c('lightgreen', 'grey60', 'purple', 'white'))
```

---

clustr *Cluster localities within regions of nearest neighbours*

Description

Spatially subsample a dataset based on minimum spanning trees connecting points within regions of set extent, with optional rarefaction to a site quota.

Usage

```
clustr(
  dat,
  xy,
  iter,
  nSite = NULL,
  distMax,
  nMin = 3,
  crs = "epsg:4326",
  output = "locs"
)
```

Arguments

dat	A `data.frame` or `matrix` containing the coordinate columns `xy` and any associated variables, e.g. taxon names.
xy	A vector of two elements, specifying the name or numeric position of columns in `dat` containing coordinates, e.g. longitude and latitude. Coordinates for any shared sampling sites should be identical, and where sites are raster cells, coordinates are usually expected to be cell centroids.
iter	The number of spatial subsamples to return
nSite	The quota of unique locations to include in each subsample.
distMax	Numeric value for maximum diameter (km) allowed across locations in a subsample
nMin	Numeric value for the minimum number of sites to be included in every returned subsample. If `nSite` supplied, `nMin` ignored.

`crs` Coordinate reference system as a GDAL text string, EPSG code, or object of class `crs`. Default is latitude-longitude (`EPSG:4326`).

`output` Whether the returned data should be two columns of subsample site coordinates (`output = 'locs'`) or the subset of rows from `dat` associated with those coordinates (`output = 'full'`).

Details

Lagomarcino and Miller (2012) developed an iterative approach of aggregating localities to build clusters based on convex hulls, inspired by species-area curve analysis (Scheiner 2003). Close et al. (2017, 2020) refined the approach and changed the proximity metric from minimum convex hull area to minimum spanning tree length. The present implementation adapts code from Close et al. (2020) to add an option for site rarefaction after cluster construction and to grow trees at random starting points `iter` number of times (instead of a deterministic, exhaustive iteration at every unique location).

The function takes a single location as a starting (seed) point; the seed and its nearest neighbour initiate a spatial cluster. The distance between the two points is the first branch in a minimum spanning tree for the cluster. The location that has the shortest distance to any points already within the cluster is grouped in next, and its distance (branch) is added to the sum tree length. This iterative process continues until the largest distance between any two points in the cluster would exceed `distMax` km. In the rare case multiple candidate points are tied for minimum distance from the cluster, one point is selected at random as the next to include. Any tree with fewer than `nMin` points is prohibited.

In the case that `nSite` is supplied, `nMin` argument is ignored, and any tree with fewer than `nSite` points is prohibited. After building a tree as described above, a random set of `nSite` points within the cluster is taken (without replacement). The `nSite` argument makes `clustr()` comparable with `cookies()` in that it spatially standardises both extent and area/locality number.

The performance of `clustr()` is designed on the assumption `iter` is much larger than the number of unique localities. Internal code first calculates the full minimum spanning tree at every viable starting point before it then samples those trees (i.e. resamples and optionally rarefies) for the specified number of iterations. This sequence means the total run-time increases only marginally even as `iter` increases greatly. However, if there are a large number of sites, particularly a large number of densely-spaced sites, the calculations will be slow even for a small number of iterations.

Value

A list of length `iter`. Each element is a `data.frame` (or `matrix`, if `dat` is a `matrix` and `output = 'full'`). If `nSite` is supplied, each element contains `nSite` observations. If `output = 'locs'` (default), only the coordinates of subsampling locations are returned. If `output = 'full'`, all `dat` columns are returned for the rows associated with the subsampled locations.

References

Antell GT, Kiessling W, Aberhan M, Saupe EE (2020). "Marine biodiversity and geographic distributions are independent on large scales." *Current Biology*, **30**(1), 115-121. doi:10.1016/j.cub.2019.10.065.

Close RA, Benson RB, Upchurch P, Butler RJ (2017). "Controlling for the species–area effect supports constrained long-term Mesozoic terrestrial vertebrate diversification." *Nature Communications*, **8**(1), 1–11. doi:10.1038/ncomms15381.

Close RA, Benson RB, Saupe EE, Clapham ME, Butler RJ (2020). "The spatial structure of Phanerozoic marine animal diversity." *Science*, **368**(6489), 420-424. doi:10.1126/science.aay8309.

Lagomarcino AJ, Miller AI (2012). "The relationship between genus richness and geographic area in Late Cretaceous marine biotas: Epicontinental sea versus open-ocean-facing settings." *PloS One*, **7**(8), e40472. doi:10.1371/journal.pone.0040472.

Scheiner SM (2003). "Six types of species–area curves." *Global Ecology and Biogeography*, **12**(6), 441-447. doi:10.1046/j.1466822X.2003.00061.x.

See Also

`cookies()`

Examples

```
# generate occurrences: 10 lat-long points in modern Australia
n <- 10
x <- seq(from = 140, to = 145, length.out = n)
y <- seq(from = -20, to = -25, length.out = n)
pts <- data.frame(x, y)

# sample 5 sets of 4 locations no more than 400km across
clustr(dat = pts, xy = 1:2, iter = 5,
       nSite = 4, distMax = 400)
```

`collSilur` *Paleobiology Database collections of Silurian marine fossils*

Description

A dataset containing the (palaeo)coordinates and recorded marine environment of 8,000 PBDB fossil collections from the Silurian, formatted and downloaded from the Paleobiology Database on 24 June 2022.

Usage

```
collSilur
```

Format

A data frame with 8345 rows and 7 variables:

paleolng, paleolat Coordinates of a collection, rotated to its palaeogeographic location with the tectonic plate model of GPlates

collection_no, reference_no Unique identifier for the collection and its published reference

environment One of 23 marine environment categories

max_ma, min_ma Bounds of the age estimate for a collection

Source

https://paleobiodb.org/

`cookies` *Rarefy localities within circular regions of standard area*

Description

Spatially subsample a dataset to produce samples of standard area and extent.

Usage

```
cookies(
  dat,
  xy,
  iter,
  nSite,
  r,
  weight = FALSE,
  crs = "epsg:4326",
  output = "locs"
)
```

Arguments

`dat`	A `data.frame` or `matrix` containing the coordinate columns `xy` and any associated variables, e.g. taxon names.
`xy`	A vector of two elements, specifying the name or numeric position of columns in `dat` containing coordinates, e.g. longitude and latitude. Coordinates for any shared sampling sites should be identical, and where sites are raster cells, coordinates are usually expected to be cell centroids.
`iter`	The number of spatial subsamples to return
`nSite`	The quota of unique locations to include in each subsample.
`r`	Numeric value for the radius (km) defining the circular extent of each spatial subsample.
`weight`	Whether sites within the subsample radius should be drawn at random (`weight = FALSE`, default) or with probability inversely proportional to the square of their distance from the centre of the subsample region (`weight = TRUE`).
`crs`	Coordinate reference system as a GDAL text string, EPSG code, or object of class `crs`. Default is latitude-longitude (`EPSG:4326`).

`output` Whether the returned data should be two columns of subsample site coordinates (`output = 'locs'`) or the subset of rows from `dat` associated with those coordinates (`output = 'full'`).

Details

The function takes a single location as a starting (seed) point and circumscribes a buffer of `r` km around it. Buffer circles that span the antemeridian (180 degrees longitude) are wrapped as a multipolygon to prevent artificial truncation. After standardising radial extent, sites are drawn within the circular extent until a quota of `nSite` is met. Sites are sampled without replacement, so a location is used as a seed point only if it is within `r` km distance of at least `nSite-1` locations. The method is introduced in Antell et al. (2020) and described in detail in Methods S1 therein.

The probability of drawing each site within the standardised extent is either equal (`weight = FALSE`) or proportional to the inverse-square of its distance from the seed point (`weight = TRUE`), which clusters subsample locations more tightly.

For geodetic coordinates (latitude-longitude), distances are calculated along great circle arcs. For Cartesian coordinates, distances are calculated in Euclidian space, in units associated with the projection CRS (e.g. metres).

Value

A list of length `iter`. Each list element is a `data.frame` or `matrix` (matching the class of `dat`) with `nSite` observations. If `output = 'locs'` (default), only the coordinates of subsampling locations are returned. If `output = 'full'`, all `dat` columns are returned for the rows associated with the subsampled locations.

If `weight = TRUE`, the first observation in each returned subsample `data.frame` corresponds to the seed point. If `weight = FALSE`, observations are listed in the random order of which they were drawn.

References

Antell GT, Kiessling W, Aberhan M, Saupe EE (2020). "Marine biodiversity and geographic distributions are independent on large scales." *Current Biology*, **30**(1), 115-121. doi:10.1016/j.cub.2019.10.065.

See Also

`clustr()`

Examples

```
# generate occurrences: 10 lat-long points in modern Australia
n <- 10
x <- seq(from = 140, to = 145, length.out = n)
y <- seq(from = -20, to = -25, length.out = n)
pts <- data.frame(x, y)

# sample 5 sets of 3 occurrences within 200km radius
cookies(dat = pts, xy = 1:2, iter = 5,
```

```
            nSite = 3, r = 200)
```

`occSilur` *Paleobiology Database occurrences of Silurian fossil brachiopods*

Description

A dataset containing the (palaeo)coordinates and genus identifications of 13,500 marine brachiopods from the Silurian (443.1-419 Ma). Records with uncertain or unaccepted taxonomic names, non-marine palaeo-environments, or missing coordinates are excluded from the original download (29 July 2022). Taxonomic synonymisation and removal of stratigraphic outliers follows the `fossilbrush` vignette example of cross-correlation with the Sepkoski range-through database [`fossilbrush::sepkoski()`].

Usage

```
occSilur
```

Format

A data frame with 13502 rows and 11 variables:

order, family, genus Latin order, family, and genus name, as synonymised against Sepkoski database

paleolng, paleolat Coordinates of an occurrence, rotated to its palaeogeographic location with the tectonic plate model of GPlates

collection_no, reference_no Unique identifiers for the collection and published reference containing the occurrence

environment One of 23 marine environment categories

max_ma, min_ma Bounds of the age estimate for an occurrence, according to the ICS 2013 geologic time scale.

accepted_name Original identification, including subgenus and species epithet if applicable, according to the latest PBDB accepted taxonomy at time of download

Source

https://paleobiodb.org/

rangeSize *Calculate common metrics of spatial distribution*

Description

Calculate occurrence count, centroid coordinates, latitudinal range (degrees), great circle distance (km), mean pairwise distance (km), and summed minimum spanning tree length (km) for spatial point coordinates.

Usage

```
rangeSize(coords, crs = "epsg:4326")
```

Arguments

coords	2-column data.frame or matrix containing x- and y-coordinates, respectively (e.g. longitude and latitude).
crs	Coordinate reference system as a GDAL text string, EPSG code, or object of class crs. Default is latitude-longitude (EPSG:4326).

Details

Coordinates and their distances are computed with respect to the original coordinate reference system if supplied, except in calculation of latitudinal range, for which projected coordinates are transformed to geodetic ones. If crs is unspecified, by default points are assumed to be given in latitude-longitude and distances are calculated with spherical geometry.

Duplicate coordinates will be removed. If a single unique point is supplied, all distance measures returned will be NA.

Value

A 1-row, 7-column matrix

Examples

```
# generate 20 occurrences for a pseudo-species
# centred on Yellowstone National Park (latitude-longitude)
# normally distributed with a standard deviation ~110 km
set.seed(2)
x <- rnorm(20, 110.5885, 2)
y <- rnorm(20,  44.4280, 1)
pts <- cbind(x,y)

rangeSize(pts)
```

sdSumry *Calculate basic spatial coverage and diversity metrics*

Description

Summarise the geographic scope and position of occurrence data, and optionally estimate diversity and evenness

Usage

```
sdSumry(
  dat,
  xy,
  taxVar,
  crs = "epsg:4326",
  collections = NULL,
  quotaQ = NULL,
  quotaN = NULL,
  omitDom = FALSE
)
```

Arguments

dat	A data.frame or matrix containing taxon names, coordinates, and any associated variables; or a list of such structures.
xy	A vector of two elements, specifying the name or numeric position of columns in dat containing coordinates, e.g. longitude and latitude. Coordinates for any shared sampling sites should be identical, and where sites are raster cells, coordinates are usually expected to be cell centroids.
taxVar	The name or numeric position of the column containing taxonomic identifications. taxVar must be of same class as xy, e.g. a numeric column position if xy is given as a vector of numeric positions.
crs	Coordinate reference system as a GDAL text string, EPSG code, or object of class crs. Default is latitude-longitude (EPSG:4326).
collections	The name or numeric position of the column containing unique collection IDs, e.g. 'collection_no' in PBDB data downloads.
quotaQ	A numeric value for the coverage (quorum) level at which to perform coverage-based rarefaction (shareholder quorum subsampling).
quotaN	A numeric value for the quota of taxon occurrences to subsample in classical rarefaction.
omitDom	If omitDom = TRUE and quotaQ or quotaN is supplied, remove the most common taxon prior to rarefaction. The nTax and evenness returned are unaffected.

Details

`sdSumry()` compiles metadata about a sample or list of samples, before or after spatial subsampling. The function counts the number of collections (if requested), taxon presences (excluding repeat incidences of a taxon at a given site), and unique spatial sites; it also calculates site centroid coordinates, latitudinal range (degrees), great circle distance (km), mean pairwise distance (km), and summed minimum spanning tree length (km). Coordinates and their distances are computed with respect to the original coordinate reference system if supplied, except in calculation of latitudinal range, for which projected coordinates are transformed to geodetic ones. If `crs` is unspecified, by default points are assumed to be given in latitude-longitude and distances are calculated with spherical geometry.

The first two diversity variables returned are the raw count of observed taxa and the Summed Common species/taxon Occurrence Rate (SCOR). SCOR reflects the degree to which taxa are common/widespread and is decoupled from richness or abundance (Hannisdal *et al.* 2012). SCOR is calculated as the sum across taxa of the log probability of incidence, λ. For a given taxon, $\lambda = -ln(1-p)$, where p is estimated as the fraction of occupied sites. Very widespread taxa make a large contribution to an assemblage SCOR, while rare taxa have relatively little influence.

If `quotaQ` is supplied, `sdSumry()` rarefies richness at the given coverage value and returns the point estimate of richness (Hill number 0) and its 95% confidence interval, as well as estimates of evenness (Pielou's J) and frequency-distribution sample coverage (given by `iNEXT$DataInfo`). If `quotaN` is supplied, `sdSumry()` rarefies richness to the given number of occurrence counts and returns the point estimate of richness and its 95% confidence interval. Coverage-based and classical rarefaction are both calculated with `iNEXT::estimateD()` internally. For details, such as how diversity is extrapolated if sample coverage is insufficient to achieve a specified rarefaction level, consult Chao and Jost (2012) and Hsieh *et al.* (2016).

Value

A `matrix` of spatial and optional diversity metrics. If `dat` is a `list` of `data.frame` objects, output rows correspond to input elements.

References

Chao A, Jost L (2012). "Coverage-based rarefaction and extrapolation: standardizing samples by completeness rather than size." *Ecology*, **93**(12), 2533–2547. doi:10.1890/111952.1.

Hannisdal B, Henderiks J, Liow LH (2012). "Long-term evolutionary and ecological responses of calcifying phytoplankton to changes in atmospheric CO2." *Global Change Biology*, **18**(12), 3504–3516. doi:10.1111/gcb.12007.

Hsieh TC, Ma KH, Chao A (2016). "iNEXT: an R package for rarefaction and extrapolation of species diversity (Hill numbers)." *Methods in Ecology and Evolution*, **7**(12), 1451–1456. doi:10.1111/2041210X.12613.

See Also

`rangeSize()`

Examples

```
# generate occurrences
set.seed(9)
x  <- sample(rep(1:5, 10))
y  <- sample(rep(1:5, 10))
# make some species 2x or 4x as common
abund <- c(rep(4, 5), rep(2, 5), rep(1, 10))
sp <- sample(letters[1:20], 50, replace = TRUE, prob = abund)
obs <- data.frame(x, y, sp)

# minimum sample data returned
sdSumry(obs, c('x','y'), 'sp')

# also calculate evenness and coverage-based rarefaction diversity estimates
sdSumry(obs, xy = c('x','y'), taxVar = 'sp', quotaQ = 0.7)
```

---

uniqify *Find unique (taxon) occurrence records*

Description

Subset a dataset to unique spatial localities or locality-taxon combinations.

Usage

```
uniqify(dat, xy, taxVar = NULL, na.rm = TRUE)
```

Arguments

dat	A data.frame or matrix containing taxon names, coordinates, and any associated variables; or a list of such structures.
xy	A vector of two elements, specifying the name or numeric position of columns in dat containing coordinates, e.g. longitude and latitude. Coordinates for any shared sampling sites should be identical, and where sites are raster cells, coordinates are usually expected to be cell centroids.
taxVar	The name or numeric position of the column containing taxonomic identifications. taxVar must be of same class as xy, e.g. a numeric column position if xy is given as a vector of numeric positions.
na.rm	Should records missing information be removed? Default is yes.

Details

The na.rm argument applies to coordinate values and, if taxVar is supplied, to taxon values. If na.rm = FALSE, any NA values will be retained and treated as their own value. Note that divvy ignores any rows with missing coordinates for the subsampling functions cookies(), clustr(), and bandit().

Value

An object with the same class and columns as `dat`, containing the subset of rows representing unique coordinates (if only `xy` supplied) or unique taxon-site combinations (if `taxVar` is also supplied). The first record at each spatial locality is retained, or if `taxVar` is specified, the first record of each taxon at a locality.

Examples

```
# generate occurrence data
x  <- rep(1, 10)
y  <- c(rep(1, 5), 2:6)
sp <- c(rep(letters[1:3], 2),
        rep(letters[4:5], 2))
obs <- data.frame(x, y, sp)

# compare original and unique datasets:
# rows 4 and 5 removed as duplicates of rows 1 and 2, respectively
obs
uniqify(obs, taxVar = 3, xy = 1:2)

# using taxon identifications or other third variable is optional
uniqify(obs, xy = c('x', 'y'))

# caution - data outside the taxon and occurrence variables
# will be lost where associated with duplicate occurrences
obs$notes <- letters[11:20]
uniqify(obs, 1:2, 3)
# the notes 'n' and 'o' are absent in the output data
```

Index