Package 'isoniche'

November 12, 2026

Type Package

Title Isodar-Adjusted, Abundance-Independent Niche Breadth

Version 1.1.0

Description Estimates habitat niche breadth corrected for the effects of abundance, using isodars (the relationships between a species' densities in pairs of habitats under density-dependent habitat selection). Fits pairwise and n-dimensional isodars, including data where sites lack some habitats, and computes the isodar-adjusted inequality (IAI) at any total abundance, with density-independent and density-dependent niche breadth, bootstrap confidence intervals and diagnostics. Method described in Granot, Dubiner and Belmaker (in revision, Ecology Letters).

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports stats, graphics, tibble

Suggests glmmTMB, testthat (>= 3.1.5), knitr, rmarkdown

LazyData true

VignetteBuilder knitr

Config/testthat/edition 3

URL https://itai811.github.io/isoniche/, https://github.com/itai811/isoniche

BugReports https://github.com/itai811/isoniche/issues

RoxygenNote 7.3.2

NeedsCompilation no

Author Itai Granot [aut, cre] (ORCID: <https://orcid.org/0000-0002-2168-3440>, Contributed equally with Shahar Dubiner), Shahar Dubiner [aut] (ORCID: <https://orcid.org/0000-0003-1771-0087>, Contributed equally with Itai Granot), Jonathan Belmaker [aut] (ORCID: <https://orcid.org/0000-0002-5618-7359>)

Maintainer Itai Granot <itai811@gmail.com>

Repository CRAN

Date/Publication 2026-11-12 16:00:02 UTC

Contents

check_coverage 2
community_to_long 3
fit_isodars 4
group_sites 8
iai 9
iai_curve 11
isoderr 12
moorea_fish 14
niche_breadth 15
pairwise_niche 18
plot.iai_curve 20
plot.niche_breadth 21
radius_sensitivity 22
simulate_isodars 23
summary.niche_breadth 24

Index **26**

`check_coverage` *Sampling coverage of each species and habitat*

Description

Summarises, for each species and habitat, how many of the species' sites sampled the habitat, how often the species was found there, and whether `fit_isodars()` would treat the habitat as used, absent or undersampled. Run it before fitting, to decide which species and habitats can be analysed and whether sites should be grouped (`group_sites()`).

Usage

```
check_coverage(
  data,
  species = NULL,
  site,
  habitat,
  abundance,
  effort = NULL,
  absent_max_count = 0,
  absent_min_sites = 5
)
```

Arguments

`data`	Long-format data: one row per species x site x habitat.
`species`, `site`, `habitat`, `abundance`, `effort`	Long format: names of the columns holding species (optional; without it the data are one species), site (or other replicate), habitat, abundance (counts) and sampling effort (optional; without it every row has effort 1).
`absent_max_count`, `absent_min_sites`	Thresholds for `absent = "auto"`; see Details.

Value

A tibble with one row per species x habitat: `species` (if given), `habitat`, `n_sites` (sites where the species was recorded in any habitat), `n_complete` (of those, sites where every habitat was sampled), `n_sampled` (sites where this habitat was sampled), `n_occupied` (sites where the species was recorded in this habitat), `total_count` and `status` (`"used"`, `"absent"` or `"undersampled"`).

See Also

`fit_isodars()`, `group_sites()`

Examples

```
cov <- check_coverage(moorea_fish, species = "species", site = "site",
                      habitat = "habitat", abundance = "abundance")
cov[cov$status != "used", ]
```

`community_to_long` *Convert a community matrix to long format*

Description

Turns a community matrix (one row per sample, one column per species, as used by vegan and most community-ecology software) and a table describing the samples into the long format used by `check_coverage()`, `fit_isodars()` and `niche_breadth()`: one row per species x site x habitat.

Usage

```
community_to_long(comm, samples, site, habitat, effort = NULL, keep = NULL)
```

Arguments

`comm`	A matrix or data frame of counts: one row per sample, one column per species (column names are the species names). Counts must be non-negative and not missing.
`samples`	A data frame with one row per row of `comm`, in the same order, giving the site, habitat and (optionally) effort of each sample.

site, habitat	Names of the site and habitat columns of samples.
effort	Optional name of the effort column of samples (area, transect length, trap nights...; positive).
keep	Optional names of other site-level columns of samples to carry over, such as longitude and latitude for group_sites(). They must have one value per site.

Details

Each row of comm is one sample (a transect, quadrat, trap, count...) and the matching row of samples says at which site and in which habitat it was taken. Samples taken at the same site and habitat are pooled: counts are summed, and so are efforts. Without an effort column, every sample has effort 1, so the effort of a site x habitat is its number of samples; sites and habitats sampled unevenly are then still compared as densities per sample.

Every sampled site x habitat appears for every species, with 0 where the species was not recorded (which is what makes zeros informative). Species never recorded are dropped.

Value

A tibble with columns species, the site and habitat columns (named as in samples), abundance, effort (named as in samples, or "effort" = number of samples) and any keep columns, ready for niche_breadth(x, species = "species", site = ..., habitat = ..., abundance = "abundance", effort = ...). Pass the effort column as effort: without it, every pooled site x habitat counts as equal effort (fit_isodars() and niche_breadth() warn when it is left out).

See Also

check_coverage(), niche_breadth(), vignette("data-preparation")

Examples

```
comm <- matrix(c(5, 0, 2,
                 3, 1, 0,
                 0, 4, 6,
                 1, 2, 2), nrow = 4, byrow = TRUE,
               dimnames = list(NULL, c("sp_a", "sp_b", "sp_c")))
samples <- data.frame(site = c("s1", "s1", "s1", "s2"),
                      habitat = c("reef", "reef", "lagoon", "reef"))
community_to_long(comm, samples, site = "site", habitat = "habitat")
```

---

fit_isodars *Fit isodars*

Description

Fits isodars to abundances (or densities) recorded in two or more habitats at a set of sites (or other replicates, such as years). Data are either in wide format (one species; one row per site, one column per habitat) or in long format (one or many species; one row per species x site x habitat).

Usage

```
fit_isodars(
  data,
  hab_cols = NULL,
  n_habitats = NULL,
  dim = c("pairwise", "ndim"),
  flip_intercept = TRUE,
  ndim_method = c("auto", "pca", "latent"),
  absent = "auto",
  absent_max_count = 0,
  absent_min_sites = 5,
  species = NULL,
  site = NULL,
  habitat = NULL,
  abundance = NULL,
  effort = NULL,
  reference_effort = NULL
)
```

Arguments

`data`	A data frame. Wide format: one row per site and one numeric column per habitat; `NA` marks a habitat that is absent from, or was not sampled at, a site. Long format: one row per species x site x habitat (see Details).
`hab_cols`	Wide format: character vector of habitat column names.
`n_habitats`	Alternatively, the number of habitat columns, taken from the start of `data`. One of `hab_cols` or `n_habitats` is required.
`dim`	`"pairwise"` or `"ndim"`.
`flip_intercept`	Pairwise only. If `TRUE`, an isodar with a negative intercept and positive slope is expressed in the reverse direction (habitats swapped), so that intercepts are non-negative.
`ndim_method`	`"auto"`, `"pca"` or `"latent"`; see Details.
`absent`	Habitats the species does not use: `"auto"` (detected from the data, see Details), `"none"`, or a character vector of habitat names (`character(0)` is the same as `"none"`).
`absent_max_count`, `absent_min_sites`	Thresholds for `absent = "auto"`; see Details.
`species`, `site`, `habitat`, `abundance`, `effort`	Long format: names of the columns holding species (optional; without it the data are one species), site (or other replicate), habitat, abundance (counts) and sampling effort (optional; without it every row has effort 1).
`reference_effort`	Long format: the sampling unit in which densities are expressed (default: median effort); see Details.

Details

Input formats. In wide format, give `hab_cols` (or `n_habitats`); the values are used as they are, and the result is a single fit. In long format, give the column names `site`, `habitat` and `abundance` (and optionally `species` and `effort`); the result is an `isodar_fits` table with one row per species, which `niche_breadth()` and `iai_curve()` take. Replicates may be sites or time steps (for example, years at one site); in long format they go in the `site` column. A site x habitat combination that appears for any species is taken as sampled, and a species without a row there has 0 individuals there; a habitat missing at a site simply has no rows. Sites where a species has no individuals in any habitat carry no information on its habitat selection and are dropped for that species.

Effort and densities. Isodars relate densities. In long format, density = abundance / effort x `reference_effort`. Effort may be an area, a transect length (constant width) or a unitless relative effort, but must be the same kind of measure in every row; it must be the same for every row of a site x habitat combination. `reference_effort` (default: the median effort of sampled site x habitat combinations) is the sampling unit in which abundances are expressed: "one individual" in the density-independent niche breadth, the spillover abundance and the x-axis of `iai_curve()` all mean one individual per `reference_effort`. Density-independent niche breadth and spillover abundance therefore depend on this unit; slopes and density-dependent niche breadth do not. The default depends on how samples happen to be pooled, so set `reference_effort` to a natural unit (for example, the area of one standard transect) and use the same value for every species you compare.

In wide format, rows are used as given: unlike long format, a site with 0 in every habitat is kept (it pulls the isodars toward the origin), so remove such rows if they mean the species was absent from the site.

Pairwise isodars (`dim = "pairwise"`) are fitted for every pair of habitats by total least squares (TLS), which is identical to major-axis regression in two dimensions. Each pair uses every site where both habitats were recorded (non-missing), so sites lacking some habitats still contribute. When many sites are incomplete, both pairwise isodars and the latent line (below) use all of them; fitting both is a useful check.

The n-dimensional isodar (`dim = "ndim"`) is a line. Under the ideal free distribution, equal fitness across n habitats imposes $n - 1$ equalities, so abundances across habitats change along a single line as total abundance changes, whatever the number of habitats. The line is described by a point (`mean`) and a unit direction vector (`direction`), with one component per habitat. Each component is the rate at which that habitat gains individuals as total abundance rises (the analogue of the isodar slope); where the line enters each habitat reflects its density-independent preference (the analogue of the intercept). Every pairwise isodar can be derived from it, and with two habitats it is the classic isodar.

With `ndim_method = "pca"` the line is the first principal axis of the covariance matrix of complete sites (the TLS line). With `ndim_method = "latent"` it is estimated from all sites with at least two recorded habitats, as a one-factor model, $N_{hs} = \mu_h + v_h t_s + \varepsilon_{hs}$, where t_s is the position of site s along the isodar (requires **glmmTMB**). `ndim_method = "auto"` uses `"pca"` when every site is complete and `"latent"` otherwise.

Absent habitats. A habitat that the species never uses carries no information about habitat selection, and an isodar cannot be fitted to it. Such a habitat is excluded from the fit and fixed at 0 abundance; `iai()` still counts it, as 0, in the inequality, so it lowers niche breadth. With `absent = "auto"` a habitat is declared absent if the species was recorded in it no more than `absent_max_count` times in total (in the units of wide `data`; as counts, before conversion to densities, in long format), and the habitat was sampled at no fewer than `absent_min_sites` sites where the species was present in

another habitat. A habitat with (almost) no records but fewer such sites is *undersampled*: it is kept in the fit, with a warning, because its zeros may reflect too little sampling rather than avoidance. A habitat used at low density and abandoned at high density is not absent; it is fitted, and `iai()` clips it to 0 at totals where the isodars predict negative abundance. With the default `absent_max_count = 0`, a single individual makes a habitat used, which can change the spillover abundance a lot; raise `absent_max_count` if a few stray individuals should not count.

Absent habitats lower niche breadth, so species should be compared over the same set of habitats, namely the habitats available to them.

Value

For `dim = "pairwise"`, a tibble of class `isodars_pairwise` with one row per pair of used habitats: `habitat_x`, `habitat_y`, `intercept`, `slope`, `sd` (SD of perpendicular residuals), `n_sites`, and `flipped` (whether the habitats were swapped). For `dim = "ndim"`, a tibble of class `isodar_line` with one row per habitat: `habitat`, `mean`, `direction` (both 0 for absent habitats); attributes `var_explained` (share of variance along the line, `"pca"` only), `eigenvalues`, `n_sites` and `ndim_method`. Both have attributes `habitats` (all habitats), `absent` and `undersampled`.

In long format, a tibble of class `isodar_fits` with one row per species: `species` (if given), `fit` (the fit as above, or `NULL` if it failed), `counts` and `effort` (site x habitat matrices), `n_sites` and `note` (warnings and errors from fitting). One summary warning is given if any species could not be fitted.

References

Morris, D.W. (1987). Spatial scale and the cost of density-dependent habitat selection. *Evolutionary Ecology*, 1, 379–388.

See Also

`niche_breadth()`, `iai()`, `isoderr()`

Examples

```
set.seed(1)
sim <- simulate_isodars(1.5, 1.5, 5, 5, noise = 1, n = 30)
fit_isodars(sim, hab_cols = c("hab1", "hab2", "hab3"))
fit_isodars(sim, hab_cols = c("hab1", "hab2", "hab3"), dim = "ndim")

# long format, several species
fish <- moorea_fish[moorea_fish$species %in% c("Acanthurus nigrofuscus",
  "Scarus schlegeli", "Zebrasoma scopas"), ]
fits <- fit_isodars(fish, species = "species", site = "site",
                    habitat = "habitat", abundance = "abundance")
fits
fits$fit[[which(fits$species == "Scarus schlegeli")]]
```

`group_sites` *Group nearby sites into sampling units*

Description

Merges sites within a given distance of each other into one sampling unit, summing counts and efforts. Grouping helps when single sites lack some habitats or hold too few individuals, at the cost of fewer units.

Usage

```
group_sites(
  data,
  site,
  lon,
  lat,
  radius_km,
  linkage = "complete",
  species = NULL,
  habitat,
  abundance,
  effort = NULL
)
```

Arguments

`data`	Long-format data: one row per species x site x habitat.
`lon`, `lat`	Names of the longitude and latitude columns (decimal degrees; one location per site).
`radius_km`	Grouping radius in km: one number, or a named vector with one radius per species.
`linkage`	`"complete"` or `"single"`.
`species`, `site`, `habitat`, `abundance`, `effort`	Long format: names of the columns holding species (optional; without it the data are one species), site (or other replicate), habitat, abundance (counts) and sampling effort (optional; without it every row has effort 1).

Details

Sites are clustered by complete linkage on great-circle distance, so every pair of sites in a group lies within `radius_km` of each other (single linkage, `linkage = "single"`, can chain distant sites together). With `radius_km = 0` only sites with identical coordinates are merged.

Counts are summed over the sites of a group, and so are efforts: the effort of a group in a habitat is the summed effort of the sites that sampled that habitat. Without an `effort` column, every site x habitat has effort 1, so the group's effort is the number of sites that sampled the habitat. Groups are named after their sites (joined with `"+"`).

The radius is not chosen automatically: choosing it to maximise completeness or niche breadth would be circular. Use `radius_sensitivity()` to see how the number of units, their completeness and the niche breadth estimates change with the radius.

Value

A tibble in long format with the original column names: the species (if given), site (the group name), habitat, abundance and effort (named `"effort"` if there was no effort column), plus `lon` and `lat` (mean of the sites) and `n_sites` (number of sites merged). Every sampled group x habitat appears for every species, with 0 where the species was not recorded.

See Also

`radius_sensitivity()`, `check_coverage()`

Examples

```
set.seed(1)
d <- simulate_isodars(1.5, 1.5, 5, 5, n = 12, format = "long")
sites <- unique(d$site)
xy <- data.frame(site = sites, lon = stats::runif(12, 0, 0.05),
                 lat = stats::runif(12, 0, 0.05))
d <- merge(d[c("site", "habitat", "abundance")], xy)
g <- group_sites(d, site = "site", lon = "lon", lat = "lat", radius_km = 2,
                 habitat = "habitat", abundance = "abundance")
head(g)
```

---

iai *Isodar-adjusted inequality (IAI)*

Description

Computes niche breadth from fitted isodars by allocating a given total abundance among habitats according to the isodars and measuring the inequality of that allocation (1 - Gini). Evaluating it at several totals gives a rarefaction-like curve of niche breadth against abundance.

Usage

```
iai(
  isodars,
  abundances = NULL,
  weights = NULL,
  scaled = TRUE,
  min_occupancy = 1,
  max_search = 10000,
  dim = NULL,
  plot = FALSE,
  method = "gini"
)
```

Arguments

isodars	Output of fit_isodars().
abundances	Numeric vector of total abundances, or NULL for the density-independent value.
weights	Pairwise only: NULL (equal weights), "1/var" (inverse variance of each isodar's vertical residuals, $1/(sd^2(1+b^2))$), or a numeric vector with one weight per isodar.
scaled	If TRUE, 1 - Gini is rescaled to 0-1, so that a species found in a single habitat scores 0 and an even spread scores 1.
min_occupancy	Individuals required in every habitat for the density-independent value.
max_search	Largest total searched for the density-independent value.
dim	Optional; inferred from isodars. Kept for compatibility.
plot	If TRUE and several abundances are given, plots the curve.
method	Inequality measure; only "gini" is currently implemented.

Details

Allocation. For pairwise isodars, the predicted abundances are those that minimise the (optionally weighted) sum of squared deviations from all pairwise isodar equations, $y - bx = a$, subject to summing to the total, with no habitat below zero. This constrained least-squares problem is solved exactly: if some habitats would be negative, the solution keeps the best set of habitats at zero (an active-set method, which can bring back a habitat dropped in an earlier step). For an n-dimensional isodar (a line), the allocation is the single point at which the line reaches the total; negative habitats are set to zero and the others rescaled in proportion to reach the total (so a clipped allocation is no longer on the line).

Density-independent niche breadth (abundances = NULL) is evaluated at the smallest total (searched over integers) at which every used habitat holds at least min_occupancy individuals. If no total up to max_search achieves this, a single row with NA is returned.

Absent habitats (see fit_isodars()) are fixed at 0 and count as 0 in the inequality, so they lower niche breadth; with a single used habitat, niche breadth is 0.

Density-dependent niche breadth is the value at a high total abundance, supplied through abundances.

Value

A tibble with total_abundance, niche_breadth, one column per habitat (predicted abundance), and note.

See Also

fit_isodars(), isoderr(), pairwise_niche()

Examples

```
set.seed(1)
sim <- simulate_isodars(1.5, 1.5, 5, 5, noise = 1, n = 30)
pw <- fit_isodars(sim, hab_cols = c("hab1", "hab2", "hab3"))
iai(pw)                          # density-independent
```

```
iai(pw, abundances = c(50, 500)) # density-dependent
ln <- fit_isodars(sim, hab_cols = c("hab1", "hab2", "hab3"), dim = "ndim")
iai(ln, abundances = 500)
```

`iai_curve` *Niche breadth as a function of total abundance*

Description

Computes the isodar-adjusted inequality along a range of total abundances for each species, with optional bootstrap bands, and the points needed to plot it: the density-independent value at each species' spillover abundance, the asymptote, preference reversals and the totals at which the curves of two species cross.

Usage

```
iai_curve(
  fit,
  totals = NULL,
  n_boot = 0,
  conf_level = 0.95,
  weights = "1/var",
  scaled = TRUE,
  min_occupancy = 1,
  seed = NULL,
  max_search = 10000
)
```

Arguments

`fit`	Output of `niche_breadth()` (whose settings are reused), the long-format output of `fit_isodars()`, or a single fit from `fit_isodars()` (no bootstrap).
`totals`	Optional numeric vector of total abundances.
`n_boot`	Number of bootstrap replicates for the bands (0 for none).
`conf_level`	Confidence level of the bands.
`weights`, `scaled`, `min_occupancy`, `max_search`	As in `niche_breadth()`; ignored when `fit` is a `niche_breadth` result.
`seed`	Optional random-number seed.

Details

By default each species' curve runs from its spillover abundance T_0 to $1000T_0$ (see `niche_breadth()`), usually far beyond the abundances observed; the plot draws the part beyond the largest observed site total as a thinner line. `totals` sets a common range instead; totals below a species' T_0 are allowed (some used habitat then holds less than one individual), so curves, reversals and crossings can lie below T_0. Totals are in individuals per reference effort (see `fit_isodars()`).

The bands are pointwise percentile intervals at each fixed total. They differ from the intervals of `nb_di` in `niche_breadth()`, where the spillover abundance is re-estimated in every bootstrap replicate; report those for density-independent niche breadth.

Where the curves of two species cross, their ranking by niche breadth depends on the abundance at which they are compared. These rank reversals are returned in attribute `"crossings"`, using the totals where both curves are defined.

Value

A tibble of class `iai_curve` with `species` (if given), `total_abundance`, `relative_abundance` (total / T_0), `niche_breadth`, and `lower` and `upper` (bootstrap band; `NA` without bootstrap). Attributes: `"points"` (per species: `spillover_abundance`, `nb_di`, `nb_dd` at the asymptote, and `max_observed`, the largest site total in the data), `"reversals"` (preference reversals within the range) and `"crossings"` (`species_1`, `species_2`, `total_abundance`, `niche_breadth`).

See Also

`plot.iai_curve()`, `niche_breadth()`

Examples

```
three <- moorea_fish[moorea_fish$species %in% c("Acanthurus nigrofuscus",
  "Scarus schlegeli", "Siganus argenteus"), ]
fits <- fit_isodars(three, species = "species", site = "site",
                    habitat = "habitat", abundance = "abundance", dim = "ndim")
curve <- iai_curve(fits, n_boot = 50, seed = 1)
plot(curve)
plot(curve, x_axis = "relative")
attr(curve, "crossings")
```

---

isoderr *Bootstrap uncertainty of the isodar-adjusted inequality*

Description

Resamples sites with replacement (keeping all habitats of a site together), refits the isodars and recomputes the IAI for each resample. Absent habitats are identified once, on the full data, and kept fixed in every resample.

Usage

```
isoderr(
  data,
  hab_cols = NULL,
  n_habitats = NULL,
  dim = "pairwise",
  abundances = NULL,
  n_boot = 1000,
```

```
  output = c("SD", "SE", "CI"),
  conf_level = 0.95,
  weights = NULL,
  flip_intercept = TRUE,
  min_occupancy = 1,
  max_search = 10000,
  seed = NULL,
  ndim_method = "auto",
  absent = "auto",
  absent_max_count = 0,
  absent_min_sites = 5
)
```

Arguments

`data`	A data frame. Wide format: one row per site and one numeric column per habitat; `NA` marks a habitat that is absent from, or was not sampled at, a site. Long format: one row per species x site x habitat (see Details).
`hab_cols`	Wide format: character vector of habitat column names.
`n_habitats`	Alternatively, the number of habitat columns, taken from the start of `data`. One of `hab_cols` or `n_habitats` is required.
`dim`	`"pairwise"` or `"ndim"`.
`abundances`	Numeric vector of total abundances, or `NULL` for the density-independent value.
`n_boot`	Number of bootstrap replicates.
`output`	`"SD"` or `"SE"` (both the bootstrap standard deviation, which is the standard error of the estimate) or `"CI"` (percentile interval).
`conf_level`	Confidence level for `output = "CI"`.
`weights`	Pairwise only: `NULL` (equal weights), `"1/var"` (inverse variance of each isodar's vertical residuals, $1/(sd^2(1+b^2))$), or a numeric vector with one weight per isodar.
`flip_intercept`	Pairwise only. If `TRUE`, an isodar with a negative intercept and positive slope is expressed in the reverse direction (habitats swapped), so that intercepts are non-negative.
`min_occupancy`	Individuals required in every habitat for the density-independent value.
`max_search`	Largest total searched for the density-independent value.
`seed`	Optional random-number seed.
`ndim_method`	`"auto"`, `"pca"` or `"latent"`; see Details.
`absent`	Habitats the species does not use: `"auto"` (detected from the data, see Details), `"none"`, or a character vector of habitat names (`character(0)` is the same as `"none"`).
`absent_max_count`, `absent_min_sites`	Thresholds for `absent = "auto"`; see Details.

Value

A tibble with `total_abundance`, `niche_breadth` (estimate from the full data), `uncertainty` (SD/SE) or `lower` and `upper` (CI), and `n_valid` (bootstrap replicates that produced a value).

Examples

```
set.seed(1)
sim <- simulate_isodars(1.5, 1.5, 5, 5, noise = 1, n = 30)
isoderr(sim, hab_cols = c("hab1", "hab2", "hab3"), abundances = 100,
        n_boot = 50, seed = 1)
```

---

moorea_fish *Reef fishes of Moorea in three habitats*

Description

Counts of 325 reef fish species in three reef habitats at 13 sites around Moorea, French Polynesia: the data of Granot, Dubiner & Belmaker (Table S2). Fish were counted on belt transects by CRIOBE between 2004 and 2018; there is one count per species, site and habitat. Only sites where a species was recorded are included for that species; all three habitats were surveyed at every site, and zeros are explicit.

Usage

```
moorea_fish
```

Format

A tibble with one row per species x site x habitat:

species Species name.

site Site (13 sites).

habitat `"barrier_reef"`, `"fringing_reef"` or `"outer_slope"`.

abundance Number of individuals counted.

Source

"MCR LTER Coral Reef Long-term Population and Community Dynamics: Fishes" in BioTIME 2.0: Dornelas, M., Antão, L. H., Bates, A. E. et al. (2025). BioTIME 2.0: expanding and improving a database of biodiversity time series. *Global Ecology and Biogeography*, 34(5), e70003. doi: 10.1111/geb.70003

Examples

```
head(moorea_fish)
length(unique(moorea_fish$species))
```

niche_breadth *Density-independent and density-dependent niche breadth*

Description

Fits isodars (unless given fitted ones) and returns, for each species, the density-independent (DI) and density-dependent (DD) niche breadth with bootstrap confidence intervals, the spillover abundance, preference reversals and diagnostic flags. With a `species` column, everything runs per species and the result has one row per species; otherwise the data are one species.

Usage

```
niche_breadth(
  fit_or_data,
  ...,
  dependent_at = "asymptote",
  min_occupancy = 1,
  reference_effort = NULL,
  absent = "auto",
  n_boot = 999,
  conf_level = 0.95,
  scaled = TRUE,
  weights = "1/var",
  diagnostics = list(),
  seed = NULL,
  max_search = 10000
)
```

Arguments

`fit_or_data`	Data in long or wide format (see `fit_isodars()`), or the long-format output of `fit_isodars()` (an `isodar_fits` table).
`...`	Passed to `fit_isodars()` when `fit_or_data` is data: the column names (`species`, `site`, `habitat`, `abundance`, `effort`) or `hab_cols`, and `dim`, `ndim_method`, `flip_intercept`, `absent_max_count`, `absent_min_sites`.
`dependent_at`	`"asymptote"`, `"max_observed"` or a total abundance; see Details.
`min_occupancy`	Individuals required in every used habitat at the spillover abundance.
`reference_effort, absent`	Passed to `fit_isodars()` (data only).
`n_boot`	Number of bootstrap replicates (0 for none).
`conf_level`	Confidence level of the bootstrap intervals.
`scaled`	If `TRUE`, niche breadth is scaled to 0-1 (see `iai()`).
`weights`	Pairwise isodars only: `"1/var"` (default, as in the paper) or `NULL`; see `iai()`.
`diagnostics`	Named list of thresholds for the flags; see Details.

seed	Optional random-number seed (bootstrap and null model). Each species then gets its own random streams, so its results do not depend on n_boot or on the other species; your random-number state is restored afterwards.
max_search	Largest total searched for the spillover abundance.

Details

Niche breadth is the isodar-adjusted inequality (iai()): total abundance is allocated among habitats as the isodars predict and niche breadth is 1 - Gini of that allocation (scaled to 0-1 by default). Absent habitats count as 0 (see fit_isodars()).

Spillover abundance T_0 is the smallest total abundance (an integer, in individuals per reference effort; see fit_isodars()) at which every used habitat holds at least min_occupancy individuals: the point at which the species starts to spill over into its least-favoured used habitat. DI niche breadth is the value at T_0. It depends on the sampling unit (reference_effort); DD niche breadth does not.

DD niche breadth. With dependent_at = "asymptote" (default) it is the limit as total abundance $T \to \infty$, where the allocation is proportional to the direction of the isodar line (or follows the pairwise slopes) and intercepts no longer matter; it is computed directly. A number gives the value at that total, and "max_observed" the value at the largest site total in the data.

The curve between them. If no two habitats change rank and no habitat is abandoned anywhere above T_0 (no reversal or abandoned flag), niche breadth follows

$$NB(T) = NB_{DD} + (NB_{DI} - NB_{DD})T_0/T,$$

and is halfway between the two values at $T = 2T_0$. Otherwise it has this form only piecewise, with different constants between successive reversals. Where the predicted abundances of two habitats cross, the ranking of habitats changes and the curve can be non-monotonic; these preference-reversal abundances are returned in reversals (totals from T_0 upward) and flagged. A habitat used at low density whose predicted abundance falls to 0 at high totals is listed in abandoned (it still counts for DI niche breadth). Use iai_curve() to compute and plot whole curves.

Bootstrap. Whole sites are resampled with replacement and the isodars refitted, keeping the absent habitats (and the line method) of the full data. Intervals are percentile intervals.

Diagnostics. Each flag_* column is TRUE when:

- few_sites: fewer than min_sites sites;
- few_pair_sites: some pair of used habitats shares fewer than min_pair_sites sites;
- low_abundance: the median site total is below min_median_count individuals (counts, not densities), or more than max_zero_share of the sampled cells of used habitats are zeros;
- di_extrapolation: T_0 lies outside the observed range of site totals by more than a factor extrapolation_ratio;
- dd_extrapolation: for the asymptote, the largest site total is below extrapolation_ratio x T_0 (with the default 2, the data do not reach the halfway point of the curve); for a numeric dependent_at, it exceeds the largest site total by more than that factor;
- abandoned: some used habitat is abandoned at high density (informational);
- reversal: habitats change rank along the curve (informational);

- poor_fit: the line explains less than min_var_explained of the variance (ndim_method = "pca"), or the pairwise isodars are inconsistent: for some three habitats i, j, k the slopes $b_{ij}b_{jk}$ and b_{ik} differ in sign or by more than a factor max_slope_ratio;
- disconnected: habitats are not connected through shared sites;
- intercept_noise: no isodar intercept differs from those expected from sampling alone. Following the paper (section 1.2.3), n_null data sets are simulated without density-dependent habitat selection: each site's individuals are redistributed among its sampled habitats in proportion to the species' overall density in each habitat (multinomial). Two-sided p-values of the pairwise TLS intercepts are Holm-corrected within species; intercept_p is the smallest, and the flag is set when it exceeds alpha. The test needs integer counts (NA otherwise). Set n_null = 0 to skip it;
- unstable_bootstrap: more than max_failed_share of bootstrap replicates failed, or a confidence interval is wider than max_ci_width;
- undersampled: a habitat had no records at too few sites to be declared absent (see fit_isodars());
- no_di: no total up to max_search occupies every used habitat;
- fit_failed: the isodars could not be fitted (see note).

Defaults: min_sites = 10, min_pair_sites = 5, min_median_count = 5, max_zero_share = 0.5, extrapolation_ratio = 2, min_var_explained = 0.5, max_slope_ratio = 2, n_null = 199, alpha = 0.05, max_failed_share = 0.1, max_ci_width = 0.5. Change any of them through diagnostics, e.g. diagnostics = list(min_sites = 8).

Comparing species. Absent habitats lower niche breadth, so species compared with each other should be assessed over the same set of habitats: the habitats available to them.

Value

A tibble of class niche_breadth, one row per species, with species (if given), method, n_sites, nb_di, nb_di_lower, nb_di_upper, spillover_abundance, spillover_lower, spillover_upper, nb_dd, nb_dd_lower, nb_dd_upper, dependent_at (Inf for the asymptote), var_explained (line, "pca" only), intercept_p, absent and abandoned (habitats, comma-separated), reversals (list of data frames: total_abundance, habitat_1 (more abundant before), habitat_2, niche_breadth), n_boot_valid, flags (names of the flags raised, comma-separated), one logical flag_* column per diagnostic, and note. Printing it adds one line giving the number of flagged species.

See Also

iai_curve(), fit_isodars(), check_coverage()

Examples

```
three <- moorea_fish[moorea_fish$species %in% c("Acanthurus nigrofuscus",
  "Scarus schlegeli", "Siganus argenteus"), ]
nb <- niche_breadth(three, species = "species", site = "site",
                    habitat = "habitat", abundance = "abundance",
                    n_boot = 99, seed = 1)
nb
nb[c("species", "nb_di", "spillover_abundance", "nb_dd", "flags")]
```

`pairwise_niche` *Niche breadth indices from pairwise isodar parameters*

Description

Summarises the intercepts and slopes of the pairwise isodars directly (Appendix S1 of Granot, Dubiner & Belmaker), as an alternative to the isodar-adjusted inequality (`iai()`), and checks that the isodars are consistent with each other. A perfect generalist has intercepts of 0 and slopes of 1 between all habitat pairs.

Usage

```
pairwise_niche(
  isodars,
  weights = NULL,
  scale = c("log", "linear"),
  cycle_tolerance = log(1.5),
  log_slope = NULL
)
```

Arguments

`isodars`	Pairwise isodars from `fit_isodars(dim = "pairwise")`, or a data frame with columns `habitat_x`, `habitat_y`, `intercept`, `slope` (and `sd`, the SD of perpendicular residuals, for `weights = "1/var"`).
`weights`	`NULL` (equal weights), `"1/var"` (inverse variance of each isodar; see Details), or a numeric vector with one non-negative weight per isodar.
`scale`	`"log"` (default; orientation-invariant indices) or `"linear"` (the earlier indices); see Details.
`cycle_tolerance`	Largest acceptable $\lvert\log(product)\rvert$ of slopes around a cycle of three habitats (default `log(1.5)`).
`log_slope`	Deprecated: use `scale`. `TRUE` is `scale = "log"` and `FALSE` is `scale = "linear"`.

Details

Orientation invariance. An isodar between habitats i and j can be written with either habitat on the x-axis: $N_j = a + bN_i$ or, equivalently, $N_i = -a/b + (1/b)N_j$. With `scale = "log"` (default) every index is the same whichever way each isodar is written:

- slope deviations are $(\log b)^2$, since $\log(1/b) = -\log b$;
- intercept deviations use the perpendicular distance of the isodar from the origin, $d = |a|/\sqrt{1+b^2}$, which is the same for both ways of writing the line.

The variance-based indices are

$$1/\sqrt{\sum w(\log b)^2/\sum w}$$

for slopes and

$$1/\sqrt{\sum wd^2/\sum w}$$

for intercepts, where w are the isodar weights. They measure deviations from the values that indicate no habitat preference ($b = 1$, $d = 0$). Larger values indicate broader niches; extreme specialists approach 0, and a perfect generalist gives `Inf`. The mean indices are the weighted means of $|\log b|$ and of d.

With `weights = "1/var"` and `scale = "log"`, each isodar is weighted by $1/sd^2$, the inverse variance of its perpendicular residuals, which is also orientation-invariant.

Non-positive slopes (for example, a habitat that loses individuals as another gains them) have no logarithm. Such isodars are left out of the slope indices, with a warning naming them, but still used in the intercept indices.

Linear scale. `scale = "linear"` gives the earlier indices: each isodar is first expressed so that its intercept is non-negative (habitats swapped if needed), then the mean intercept and mean slope, and $1/\sqrt{\sum w(b-1)^2/\sum w}$ and $1/\sqrt{\sum wa^2/\sum w}$, with `"1/var"` weights $1/(sd^2(1+b^2))$. These depend on how each isodar is oriented, and slopes of 2 and 0.5 (preferences of equal strength in opposite directions) count as different deviations.

Consistency check. If the isodars describe one consistent pattern of habitat selection, the slopes around any cycle of three habitats multiply to 1: $b_{AB}b_{BC}b_{CA} = 1$, where b_{AB} is the slope of habitat B on habitat A. Each cycle's product is returned in `cycles`, and the result is flagged as inconsistent when $|\log(product)|$ exceeds `cycle_tolerance` for any cycle (or the product is not positive).

Units. The intercept indices are in units of abundance per sampling unit (see `reference_effort` in `fit_isodars()`), so they depend on survey effort and are comparable among species only when surveys are standardised. The slope indices are unitless.

Value

An object of class `pairwise_niche`, a list with

- `indices`: a one-row tibble. For `scale = "log"`: `mean_abs_log_slope`, `mean_distance`, `slope_niche_breadth`, `intercept_niche_breadth`, `n_isodars`, `n_slope` (isodars used for the slope indices) and `note`. For `scale = "linear"`: `mean_intercept`, `mean_slope`, `slope_niche_breadth`, `intercept_niche_breadth`, `n_isodars` and `note`.
- `cycles`: a tibble with one row per cycle of three habitats with isodars between all three pairs: `habitat_1`, `habitat_2`, `habitat_3`, `product` ($b_{12}b_{23}b_{31}$), `log_product` and `inconsistent`.
- `inconsistent`: `TRUE` if any cycle is inconsistent.
- `isodars`: the isodars used, with their `weight`, and for `scale = "log"` the `distance` d and `log_slope`; for `scale = "linear"` they are oriented to non-negative intercepts.
- `scale`, `cycle_tolerance`.

See Also

`fit_isodars()`, `iai()`

Examples

```
set.seed(1)
sim <- simulate_isodars(1.5, 1.5, 5, 5, noise = 1, n = 30)
pw <- fit_isodars(sim, hab_cols = c("hab1", "hab2", "hab3"))
pn <- pairwise_niche(pw, weights = "1/var")
pn
pn$cycles
pairwise_niche(pw, scale = "linear")$indices
```

`plot.iai_curve` *Plot niche breadth curves*

Description

Plots the curves from `iai_curve()`: one line per species with bootstrap bands (dotted), a point at each species' density-independent value (at its own spillover abundance), the asymptote (density-dependent value) as a short horizontal dash at the right, and preference reversals (crosses). Beyond the largest total observed at any site of a species, the curve is drawn as a thinner line: there it extrapolates the fitted isodars.

Usage

```
## S3 method for class 'iai_curve'
plot(
  x,
  species = NULL,
  x_axis = c("log_abundance", "relative"),
  abundances = NULL,
  crossings = FALSE,
  legend = TRUE,
  col = NULL,
  ...
)
```

Arguments

`x`	Output of `iai_curve()`.
`species`	Optional species to plot (default: all).
`x_axis`	`"log_abundance"` (total abundance, log scale) or `"relative"` (total abundance / spillover abundance, log scale; every species' DI value is then at 1).
`abundances`	Optional total abundances to mark with vertical lines (`"log_abundance"` only).
`crossings`	If `TRUE`, marks the totals where two species' curves cross (open circles; `"log_abundance"` only).
`legend`	`TRUE` (default) adds a legend (up to 12 species) in a corner free of curves and bands, or above the plot if there is none; a position such as `"topleft"` or `"bottomright"` (see `graphics::legend()`), or `"outside"` (above the plot), places it there; `FALSE` omits it.

col	Colours, one per species (default: the current palette).
...	Passed to graphics::plot().

Value

x, invisibly.

See Also

iai_curve()

---

plot.niche_breadth *Plot niche breadth against abundance*

Description

Computes the niche breadth curves of a niche_breadth() result with iai_curve() and plots them with plot.iai_curve().

Usage

```
## S3 method for class 'niche_breadth'
plot(x, n_boot = 0, seed = NULL, ...)
```

Arguments

x	Output of niche_breadth().
n_boot, seed	Passed to iai_curve() (bootstrap bands; default none).
...	Passed to plot.iai_curve(), e.g. species or x_axis.

Value

The iai_curve() result, invisibly.

See Also

iai_curve(), plot.iai_curve()

Examples

```
three <- moorea_fish[moorea_fish$species %in% c("Acanthurus nigrofuscus",
  "Scarus schlegeli", "Siganus argenteus"), ]
nb <- niche_breadth(three, species = "species", site = "site",
                    habitat = "habitat", abundance = "abundance",
                    n_boot = 0, diagnostics = list(n_null = 0))
plot(nb)
```

`radius_sensitivity` *Sensitivity of niche breadth to the grouping radius*

Description

Groups sites with `group_sites()` at each radius and estimates niche breadth with `niche_breadth()`, showing how the number of sampling units, their completeness and the estimates change with the radius.

Usage

```
radius_sensitivity(
  data,
  species = NULL,
  site,
  habitat,
  abundance,
  effort = NULL,
  lon,
  lat,
  radii_km = c(0, 0.5, 1, 2, 5),
  linkage = "complete",
  ...
)
```

Arguments

`data`	Long-format data: one row per species x site x habitat.
`species`, `site`, `habitat`, `abundance`, `effort`	Long format: names of the columns holding species (optional; without it the data are one species), site (or other replicate), habitat, abundance (counts) and sampling effort (optional; without it every row has effort 1).
`lon`, `lat`	Names of the longitude and latitude columns (decimal degrees; one location per site).
`radii_km`	Radii (km) to compare.
`linkage`	`"complete"` or `"single"`.
`...`	Passed to `niche_breadth()` (for example `dim`, `n_boot`; the default here is `n_boot = 0`).

Value

A tibble with one row per radius x species: `radius_km`, `species` (if given), `n_units` (sampling units with the species), `complete_share` (share of those units in which every habitat was sampled), `nb_di`, `spillover_abundance`, `nb_dd` and `flags`.

See Also

group_sites()

Examples

```
set.seed(1)
d <- simulate_isodars(1.5, 1.5, 5, 5, n = 20, p_missing = 0.3,
                      format = "long")
xy <- data.frame(site = unique(d$site), lon = stats::runif(20, 0, 0.1),
                 lat = stats::runif(20, 0, 0.1))
d <- merge(d, xy)
radius_sensitivity(d, site = "site", lon = "lon", lat = "lat",
                   habitat = "habitat", abundance = "abundance",
                   radii_km = c(0, 2, 5), diagnostics = list(n_null = 0))
```

---

simulate_isodars *Simulate abundances in habitats linked by isodars*

Description

Simulates abundances at n sites along a chain of isodars: habitat 1 is drawn uniformly between 0 and hab1_max, and each following habitat $k+1$ = intercepts[k] + slopes[k] x habitat k + noise. Values are rounded to integers and negative values set to 0. The default is three habitats, set by slope1, slope2, int1 and int2; the sequence of random numbers is the same as in version 1.0.0.

Usage

```
simulate_isodars(
  slope1 = 1,
  slope2 = 1,
  int1 = 0,
  int2 = 0,
  noise = 1,
  n = 30,
  hab1_max = 100/(slopes[1]^1.2) - intercepts[1],
  slopes = c(slope1, slope2),
  intercepts = c(int1, int2),
  p_missing = 0,
  min_habitats = 2,
  format = c("wide", "long")
)
```

Arguments

slope1, int1	Isodar of habitat 2 on habitat 1.
slope2, int2	Isodar of habitat 3 on habitat 2.

noise	Standard deviation of the Gaussian noise.
n	Number of sites.
hab1_max	Upper bound of abundance in habitat 1. The default, 100 / slopes[1]^1.2 - intercepts[1], keeps total abundances in a comparable range across parameter values.
slopes, intercepts	Any number of habitats: vectors of length $H-1$ giving the isodar of each habitat on the previous one (default: c(slope1, slope2) and c(int1, int2)).
p_missing	Probability that a habitat is missing (not sampled) at a site; at least min_habitats habitats are kept at every site.
min_habitats	Minimum number of habitats recorded per site.
format	"wide" (one column per habitat, NA where missing) or "long" (columns site, habitat, abundance; missing habitats have no row).

Value

Wide: a tibble with columns hab1, ..., habH and total_abundance. Long: a tibble with site, habitat and abundance.

Examples

```
set.seed(1)
simulate_isodars(1, 1.1, 1, 0, noise = 2, n = 10)
simulate_isodars(slopes = c(1.5, 1, 0.8), intercepts = c(5, 0, 2), n = 5,
                 p_missing = 0.3)
```

---

summary.niche_breadth *Summarise niche breadth results and their diagnostics*

Description

Summarises a niche_breadth() result: how many species have estimates, the range of niche breadth values, and for each diagnostic flag how many species raised it, what it means and what to do about it.

Usage

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

## S3 method for class 'summary.niche_breadth'
print(x, ...)
```

Arguments

object	The output of niche_breadth().
...	Ignored.
x	The output of summary().

Value

An object of class `summary.niche_breadth`: a list with `n_species`, `n_estimated` (species with DI and DD niche breadth), `n_clean` (estimated species with no warning flags), `clean` (their names, or row numbers without a `species` column), `estimates` (quartiles of `nb_di`, `spillover_abundance` and `nb_dd`), `flags` (a data frame with `flag`, `type`, `n`, `meaning` and `advice`, for the flags raised) and `settings`. Flags of type `"info"` (abandoned habitats, preference reversals) describe the species and are not warnings.

See Also

`niche_breadth()`, `vignette("diagnostics")`

Examples

```
three <- moorea_fish[moorea_fish$species %in% c("Acanthurus nigrofuscus",
  "Scarus schlegeli", "Siganus argenteus"), ]
nb <- niche_breadth(three, species = "species", site = "site",
                    habitat = "habitat", abundance = "abundance",
                    n_boot = 0, seed = 1)
summary(nb)
```

Index

∗ **datasets**
moorea_fish, 14

check_coverage, 2
check_coverage(), *3, 4*, *9*, *17*
community_to_long, 3

fit_isodars, 4
fit_isodars(), *2, 3*, *10, 11*, *15–17*, *19*

graphics::legend(), *20*
graphics::plot(), *21*
group_sites, 8
group_sites(), *2–4*, *22, 23*

iai, 9
iai(), *6, 7*, *15, 16*, *18, 19*
iai_curve, 11
iai_curve(), *6*, *16, 17*, *20, 21*
isoderr, 12
isoderr(), *7*, *10*

moorea_fish, 14

niche_breadth, 15
niche_breadth(), *3, 4*, *6, 7*, *11, 12*, *21, 22*, *24, 25*

pairwise_niche, 18
pairwise_niche(), *10*
plot.iai_curve, 20
plot.iai_curve(), *12*, *21*
plot.niche_breadth, 21
print.summary.niche_breadth *(*summary.niche_breadth*)*, 24

radius_sensitivity, 22
radius_sensitivity(), *9*

simulate_isodars, 23
summary.niche_breadth, 24