Package: isoniche (via r-universe)

October 13, 2026

Title Calculating Density-Independent Niche Breadth Indices from Abundance Data

Version 1.0.0

Description Deriving density-independent and density-dependent niche breadth indices from abundance data of two or more habitats, including both the pairwise and n-dimensional methods to calculate isodar-adjusted inequality. Methods are described in Granot, Dubiner & Belmaker (in revision), ``Why abundance-based indices of niche breadth are biased, and what can be done to improve them", Ecology Letters.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports tibble, dplyr, stats

NeedsCompilation no

Author Shahar Dubiner [aut, cre] (ORCID: <https://orcid.org/0000-0003-1771-0087>), Itai Granot [aut], Jonathan Belmaker [aut]

Maintainer Shahar Dubiner <dubiner@mail.tau.ac.il>

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

Date/Publication 2026-08-25 21:10:12 UTC

RemoteUrl https://github.com/cran/isoniche

RemoteRef HEAD

RemoteSha 00ff115b777e839de119fa40fc515b62b52301fc

Contents

fit_isodars 2
iai 3
isoderr 4
simulate_isodars 5

Index **7**

`fit_isodars` *Fit pairwise or n-dimensional isodars using total least squares*

Description

Fits isodar relationships using total least squares (TLS), which minimizes orthogonal distances between observations and the fitted relationship.

Usage

```
fit_isodars(
  data,
  hab_cols = NULL,
  n_habitats = ncol(data),
  flip_intercept = TRUE,
  dim = "pairwise"
)
```

Arguments

`data`	A data frame containing abundance columns for the habitats.
`hab_cols`	Character vector giving the habitat abundance columns. If `NULL`, the first `n_habitats` columns are used.
`n_habitats`	Number of habitats to use when `hab_cols = NULL`.
`flip_intercept`	Logical. If TRUE, pairwise relationships with a negative intercept are reversed so that the reported intercept is positive.
`dim`	Either `"pairwise"` (default) or `"ndim"`.

Details

With `dim = "pairwise"` (default), TLS relationships are fitted between every pair of habitats.

With `dim = "ndim"`, a single multidimensional TLS hyperplane is fitted using all habitats simultaneously.

Value

For `dim = "pairwise"`, a tibble with one row for every habitat pair, containing the coefficients of the TLS regression. For `dim = "ndim"`, a tibble containing the coefficients of the n-dimensional TLS hyperplane.

Examples

```
set.seed(1)
populations1 <- simulate_isodars(1, 1.1, 1, 0, noise = 2, n = 10)
  fit_isodars(populations1, hab_cols=c("hab1","hab2","hab3"))
populations2 <- simulate_isodars(1.5, 2, 5, 2, noise = 2, n = 10)
  fit_isodars(populations2, hab_cols=c("hab1","hab2","hab3"))
```

iai *Compute the isodar-adjusted inequality (IAI) index*

Description

Reconstructs habitat abundances for one or more requested total abundances using either pairwise or n-dimensional isodars.

Usage

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

Arguments

isodars	A data frame returned by fit_isodars().
abundances	Numeric vector of total abundances. If NULL, the minimum total abundance yielding positive abundance in all habitats is searched.
weights	Weighting scheme for pairwise reconstruction. Either NULL, a numeric vector, or "1/var". Not used for dim = "ndim" (1 isodar - no variance).
method	Currently only "gini".
plot	Logical. If TRUE, plots niche breadth against chosen total abundances.
scaled	Logical. If TRUE, coerces minimum possible IAI to be zero rather than 1/n.
max_search	Maximum abundance searched when abundances = NULL.
dim	Either "pairwise" or "ndim". Should match the dim used in fit_isodars().

Details

With dim = "pairwise", the pairwise relationships are simultaneously fitted using (optionally weighted) total least squares (TLS).

With dim = "ndim", the n-dimensional TLS hyperplane is used directly. For a given total abundance, the reconstructed abundance vector is the point on the TLS hyperplane and total-abundance plane that is closest to equal allocation among habitats. Non-negative abundances are enforced using an active-set procedure.

Niche breadth is defined as 1 - Gini(x).

Value

A tibble containing total abundance, niche breadth, and the reconstructed abundance of each habitat.

Examples

```
set.seed(1)
populations1 <- simulate_isodars(1, 1.1, 1, 0, noise = 2, n = 10)
 isodars1 <- fit_isodars(populations1, hab_cols=c("hab1","hab2","hab3"))
  iai(isodars1)
   iai(isodars1,abundances=c(6,10,16,20,40,60,130,260),weights="1/var",plot=TRUE)
populations2 <- simulate_isodars(1.5, 2, 5, 2, noise = 2, n = 10)
 isodars2 <- fit_isodars(populations2, hab_cols=c("hab1","hab2","hab3"), dim="ndim")
  iai(isodars2, dim="ndim", scaled = "FALSE")
   iai(isodars2, dim="ndim")
    iai(isodars2, dim="ndim",abundances=c(26,46,60,106,160,260),plot=TRUE)
```

---

isoderr *Bootstrap uncertainty of the isodar-adjusted inequality index*

Description

Resamples the original abundance data with replacement, refits the isodars for each bootstrap replicate, and calculates the resulting isodar-adjusted niche breadth.

Usage

```
isoderr(
  data,
  hab_cols = NULL,
  n_habitats = ncol(data),
  dim = "pairwise",
  abundances = NULL,
  n_boot = 1000,
  output = "SD",
  conf_level = 0.95,
  weights = NULL,
  flip_intercept = TRUE,
  max_search = 10000,
  seed = NULL
)
```

Arguments

data	A data frame containing abundance columns for the habitats.
hab_cols	Character vector giving the habitat abundance columns. If NULL, the first n_habitats columns are used.

n_habitats	Number of habitats to use when hab_cols = NULL.
dim	Either "pairwise" (default) or "ndim".
abundances	Numeric vector of total abundances. If NULL, the abundance selected by iai() from the original data is used.
n_boot	Number of bootstrap replicates.
output	Either "SD", "SE", or "CI".
conf_level	Confidence level for CI. Default is 0.95.
weights	Weighting scheme passed to iai(). Only applicable to dim = "pairwise".
flip_intercept	Passed to fit_isodars().
max_search	Maximum abundance searched when abundances = NULL.
seed	Optional random-number seed.

Details

With dim = "pairwise", pairwise TLS isodars are refitted for every bootstrap replicate.

With dim = "ndim", an n-dimensional TLS hyperplane is refitted for every bootstrap replicate.

The requested total abundances are determined once from the original dataset and then held constant across bootstrap replicates.

Value

A tibble containing the total abundance, bootstrap estimate, and requested measure of uncertainty.

Examples

```
set.seed(1)
populations1 <- simulate_isodars(1.5, 2, 5, 2, noise = 2, n = 10)
isoderr(populations1, n_boot=100, weights=NULL,hab_cols=c("hab1","hab2","hab3"))
isoderr(populations1, n_boot=100,output="CI",hab_cols=c("hab1","hab2","hab3"))
```

---

simulate_isodars *Simulate abundances across three habitats*

Description

Simulates abundance data for three habitats across n sites using sequential linear relationships (isodars) with Gaussian noise:

- hab2 is generated from hab1 using slope1 and int1
- hab3 is generated from hab2 using slope2 and int2

Usage

```
simulate_isodars(
  slope1,
  slope2,
  int1,
  int2,
  noise = 1,
  n = 30,
  hab1_max = 100/(slope1^1.2) - int1
)
```

Arguments

slope1	Numeric. Isodar slope for hab2 ~ hab1.
slope2	Numeric. Isodar slope for hab3 ~ hab2.
int1	Numeric. Isodar intercept for hab2 ~ hab1.
int2	Numeric. Isodar intercept for hab3 ~ hab2.
noise	Numeric (>= 0). Standard deviation of Gaussian noise.
n	Integer (> 0). Number of simulated sites.
hab1_max	Integer. maximum abundance value in habitat 1 (other maxima calculated from hab1_max). Default is derived to generally suit the isodar parameters, but is arbitrary.

Details

The output includes simple per-site niche metrics (CV and Gini) computed from the three habitat abundances.

Value

A tibble with one row per simulated site containing:

- hab1, hab2, hab3: simulated abundances (nonnegative integers)
- total_abundance: hab1 + hab2 + hab3
- mean_abundance: total_abundance divided by 3
- sd: scaled standard deviation of habitat abundances
- cv: coefficient of variation (NA if mean_abundance == 0)
- gini: Gini coefficient of habitat abundances

Examples

```
set.seed(1)
simulate_isodars(1, 1.1, 1, 0, noise = 2, n = 10)
simulate_isodars(1.5, 2, 5, 2, noise = 2, n = 10)
```

Index

fit_isodars, 2

iai, 3
isoderr, 4

simulate_isodars, 5