

Package ‘isoniche’

May 10, 2026

Title Calculating Density-Independent Niche Breadth Indices from Abundance Data

Version 0.1.1

Description Deriving isodar-based niche breadth indices from abundance data of two or more habitats, including several methods based on pairwise isodars, multidimensional isodars, and isodar-adjusted inequality.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports tibble, dplyr, lmodel2

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 CRAN

Date/Publication 2026-05-10 19:00:02 UTC

Contents

fit_isodar	2
isodar_adj_niche	3
ndim_isoniche	4
pairwise_isoniche	5
simulate_isodars	6

Index	8
--------------	----------

fit_isodar

Fit pairwise isodars between two or more habitats

Description

Fits pairwise isodar relationships between all habitat pairs using `lmodel2::lmodel2()` (Model II regression). Each output row represents the relationship:

$$\text{habitat}_y = \text{intercept} + \text{slope} \times \text{habitat}_x$$

Usage

```
fit_isodar(
  data,
  hab_cols = NULL,
  n_habitats = ncol(data),
  flip_intercept = TRUE
)
```

Arguments

`data` A data frame of abundance data with one column per habitat.

`hab_cols` A character vector of length `n_habitats`, detailing the abundance columns. Currently defaults to first columns.

`n_habitats` Integer (≥ 2). Number of habitat columns to use, starting at column 1.

`flip_intercept` Logical. If TRUE (default), flip axes when intercept < 0 .

Details

It is crucial to either place the columns with abundance for each habitat as the FIRST columns, or to explicitly state the habitat columns in `hab_cols`.

If the fitted intercept is negative, the relationship is flipped (axes swapped) to provide a biologically interpretable representation (negative intercepts are meaningless in this context) and model refitted.

The `sd` column is the residual SD from vertical residuals in the reported equation.

Value

A tibble with one row per habitat pair containing:

- `habitat_x`, `habitat_y`
- `slope`, `intercept`
- `sd`: residual SD in the reported equation
- `p_slope`, `p_intercept`
- `flipped`: TRUE if axes were flipped (due to `int < 0`), FALSE otherwise

Examples

```
set.seed(1)
isod <- simulate_isodars(1, 2, 5, 1, noise = 2, n = 10)
fit_isodar(isod, n_habitats = 3)
```

isodar_adj_niche	<i>Compute the isodar-adjusted inequality index for one or more total abundances</i>
------------------	--

Description

Reconstructs habitat abundances for each requested total abundance by solving a weighted least-squares system defined by the pairwise isodars:

$$\text{habitat}_y = \text{intercept} + \text{slope} \times \text{habitat}_x$$

Usage

```
isodar_adj_niche(
  isodars,
  abundances = NULL,
  weights = NULL,
  alpha = 0.1,
  sig_weights = c(sig = 1, nonsig = 0),
  method = c("gini"),
  plot = FALSE,
  max_search = 10000
)
```

Arguments

isodars	A data frame returned by <code>fit_isodar()</code>
abundances	Numeric vector of total abundances. If <code>NULL</code> , the minimal total abundance yielding occupancy in all habitats ("baseline") is chosen when possible; otherwise a default exploratory sequence is used
weights	Weighting scheme. One of: <ul style="list-style-type: none"> • <code>NULL</code>: equal weights • numeric vector of length <code>nrow(isodars)</code> • <code>"1/var"</code>: weights proportional to $1/\text{sd}^2$ • <code>"sig"</code>: weights based on significance of intercepts, using <code>sig_weights</code>
alpha	Numeric in $(0, 1)$. Significance threshold used when <code>weights = "sig"</code>
sig_weights	Numeric vector of length 2 giving weights for significant and non-significant isodars when <code>weights = "sig"</code> . Can be named <code>c(sig = , nonsig =)</code> or unnamed <code>c(sig, nonsig)</code> . Default is <code>c(sig = 1, nonsig = 0)</code> .

method	Character. Currently only method = "gini" available
plot	Logical. If TRUE, plots adjusted niche breadth versus total abundance
max_search	Integer. Maximum total abundance to search when abundances = NULL, in order to find baseline n

Details

The reconstruction is constrained to be nonnegative and to sum to the requested total abundance (active-set heuristic). Niche breadth is then computed from the reconstructed habitat vector; currently defined as:

$$\text{nichebreadth} = 1 - Gini(x)$$

of isodar-reconstructed abundances at a given N.

To emphasize the intercept component, use baseline abundances provided by abundances = NULL. To emphasize the slope component, we recommend to set abundances to the highest n in your dataset. To see how niche breadth changes as a function of n, enter a vector of increasing abundances.

Value

A tibble with one row per total abundance containing:

- total_abundance
- niche_breadth
- one column per habitat with reconstructed abundance

Examples

```
set.seed(1)
isod <- simulate_isodars(1, 2, 5, 1, noise = 2, n = 10)
INB <- fit_isodar(isod, n_habitats = 3)
isodar_adj_niche(INB, max_search = 100) # automatically checks only baseline abundance
isodar_adj_niche(INB, abundances = c(15, 30, 45), max_search = 100) # set specific values
```

ndim_isoniche

Multidimensional isodar-based niche breadth indices

Description

Computes vector- and orthogonal isodar-based niche breadth components from abundance data across all habitats simultaneously, generalizing the slope (density-dependent) and intercept/baseline (density-independent) components of isodars to n dimensions.

Usage

```
ndim_isoniche(data, n_habitats = ncol(data), scale = TRUE, weights = TRUE)
```

Arguments

data	A data frame of abundance data (one column per habitat).
n_habitats	Integer (≥ 2). Number of habitat columns to use, starting at column 1.
scale	Logical. If TRUE, results are raised to the power of $n_habitats - 1$ to compensate for reduced sensitivity due to added dimensions.
weights	Logical. If TRUE, habitats are weighted by inverse residual variance. If FALSE, uniform weights are used.

Details

Both indices return values between 0 and 1, where 1 indicates a perfect generalist and 0 a specialist. Habitat axes can optionally be weighted by inverse residual variance.

Value

A tibble with vector, orthogonal, and n_habitats.

Examples

```
set.seed(1)
isod <- simulate_isodars(1, 2, 5, 1, noise = 2, n = 10)
```

pairwise_isoniche *Pairwise isodar-based niche breadth indices*

Description

Computes niche breadth indices from a set of pairwise isodars (slopes and intercepts) in the format of the output table of `fit_isodar()`. Two methods are available: a weighted mean of isodar components and a weighted inverse variance of the isodars' divergence from neutrality.

Usage

```
pairwise_isoniche(  
  data,  
  method = c("inverse_variance", "mean"),  
  weights = NULL,  
  alpha = 0.1,  
  sig_weights = c(sig = 1, nonsig = 0)  
)
```

Arguments

data	A data frame returned by <code>fit_isodar()</code> .
method	Character. One of "mean" or "inverse_variance".
weights	Weighting scheme for pairwise isodars. One of: <ul style="list-style-type: none"> • NULL: uniform weights • numeric vector of length <code>nrow(data)</code> • "1/var": weights proportional to $1/\text{sd}^2$ (sd from <code>fit_isodar</code>) • "sig": weights based on significance of intercepts, using <code>sig_weights</code>
alpha	Numeric in (0, 1). Significance threshold used when <code>weights = "sig"</code> .
sig_weights	Numeric vector of length 2 giving weights for significant and non-significant isodars when <code>weights = "sig"</code> . Can be named <code>c(sig = ..., nonsig = ...)</code> or unnamed <code>c(sig, nonsig)</code> . Default is <code>c(sig = 1, nonsig = 0)</code> .

Value

A one-row tibble with columns `intercept`, `slope`, and `method`.

Examples

```
set.seed(1)
isod <- simulate_isodars(1, 2, 5, 1, noise = 2, n = 10)
INB <- fit_isodar(isod, n_habitats = 3)
pairwise_isoniche(INB, method = "inverse_variance", weights = "1/var")
pairwise_isoniche(INB, method = "inverse_variance", weights = "sig",
  sig_weights = c(sig = 1, nonsig = 0.25), alpha = 0.1)
```

simulate_isodars	<i>Simulate abundances across three habitats</i>
------------------	--

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, 2, 5, 1, noise = 2, n = 10)
```

Index

`fit_isodar`, 2

`isodar_adj_niche`, 3

`ndim_isoniche`, 4

`pairwise_isoniche`, 5

`simulate_isodars`, 6