Package ‘nichetools’

June 7, 2024

Type Package

Title Complementary Package to ‘nicheROVER’ and ‘SIBER’

Version 0.2.0

Description Provides functions complementary to packages ‘nicheROVER’ and ‘SIBER’ allowing the user to extract Bayesian estimates from data objects created by the packages ‘nicheROVER and ‘SIBER’. Please see the following publications for detailed methods on ‘nicheROVER’ and ‘SIBER’

Depends R (>= 2.10)

Imports cli, dplyr, ellipse, lifecycle, nicheROVER, purrr, rlang, tibble, tidyr

License CC0

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

URL https://benjaminhlina.github.io/nichetools/

Suggests ggplot2, ggtext, ggh4x, janitor, knitr, patchwork, rmarkdown, SIBER, stringr, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Benjamin L. Hlina [aut, cre]

Maintainer Benjamin L. Hlina <benjamin.hlina@gmail.com>

Repository CRAN

Date/Publication 2024-06-07 14:20:02 UTC
extract_mu

Contents

extract_mu ................................................................. 2
extract_niche_size ................................................... 3
extract_overlap .......................................................... 4
extract_sigma ............................................................. 5
mu_est_long .............................................................. 6
niche_ellipse ............................................................. 7
niw_fish_post ............................................................ 8
over_stat ................................................................. 8
post_sam_siber ........................................................... 9
sigma_est_wide ........................................................... 9

Index 10

extract_mu extract \( \mu \)

Description

Extract Bayesian estimates of \( \mu \) from data objects created by nicheROVER or SIBER.

Usage

extract_mu(
  data,
  pkg = NULL,
  isotope_a = NULL,
  isotope_b = NULL,
  data_format = NULL
)

Arguments

data a list created by the function niw.post() or siberMVN() in the package nicheROVER or SIBER, respectfully.
pkg a character string that is the name of the package that you’re using. Defaults to “nicheROVER”. Alternatively the user can supply the argument with “SIBER”.
isotope_a a character string to change the column name of the first isotope used in the analysis. Defaults to "d13c".
isotope_b a character string to change the name of second isotope used in the analysis. Defaults to "d15n".
data_format a character string that decides whether the returned object is in long or wide format. Default is "long", with the alternative supplied being "wide".
**extract_niche_size**

**Value**

Returns a tibble of extracted estimates of $\mu$ created by the function `niw.post()` or `siberMVN()` in the packages `nicheROVER` and `SIBER`.

The tibble will contain five columns in the following order, `metric`, `sample_name`, `sample_number`, and the names of the isotope columns supplied to `niw.post()` or `siberMVN()` (e.g., `d13c` and `d15n`).

**See Also**

`nicheROVER::niw.post()` and `SIBER::siberMVN()`

**Examples**

```r
extract_mu(
  data = niw_fish_post
)
extract_mu(
  data = post_sam_siber,
  pkg = "SIBER"
)
```

---

### Description

Extract niche size based on elliptical niche region of Bayesian estimates of sigma created by function `niw.post()` in the package `nicheROVER`. This function is a wrapper around `nicheROVER::niche.size`.

### Usage

```r
extract_niche_size(data, name = NULL, prob = NULL)
```

### Arguments

- **data**: a list created by the function `niw.post()` in the package `nicheROVER`.
- **name**: a character string that will be assigned as the column name for groups. Default is `sample_name`.
- **prob**: a numeric bound by 0 and 1 indicating the probabilistic niche size. Default is 0.95.

### Value

A tibble containing three rows, `sample_name`, `id`, and `niche_size`. 
See Also

`nicheROVER::niche.size()` and `nicheROVER::niw.post()`

Examples

```r
evaluate_overlap(data = niw_fish_post)
```

---

**Description**

Extract Bayesian estimates of similarities among groups produced by the following function `overlap()` in the package `nicheROVER`.

**Usage**

```r
evaluate_overlap(data, name_a = NULL, name_b = NULL)
```

**Arguments**

- **data**: a `data.frame` object containing matrices created by the function `overlap()` in the package `nicheROVER`.
- **name_a**: character string to supply for the first sample_name used in `overlap()`. Defaults to "sample_name_a".
- **name_b**: character string to supply for the second sample_name used in `overlap()`. Defaults to "sample_name_b".

**Value**

A `tibble` containing five rows, `sample_name_a`, `id`, `sample_name_b`, `sample_number`, and `niche_overlap`.

**See Also**

`nicheROVER::overlap()`

**Examples**

```r
evaluate_overlap(data = over_stat)
```
**extract_sigma**  

**Description**  

Extract Bayesian estimates of \( \Sigma \) from data objects created by `nicheROVER` or `SIBER`.

**Usage**  

```r  
exttract_sigma(  
data,  
pkg = NULL,  
isotope_a = NULL,  
isotope_b = NULL,  
data_format = NULL  
)  
```

**Arguments**  

- **data**: a list created by the function `niw.post()` or `siberMVN()` in the package `nicheROVER` or `SIBER`, respectfully.
- **pkg**: a character string that is the name of the package that you're using. Defaults to "nicheROVER". Alternately the user can supply the argument with "SIBER".
- **isotope_a**: a character string to change the column name of the first isotope used in the analysis. Defaults to "d13c".
- **isotope_b**: a character string to change the name of second isotope used in the analysis. Defaults to "d15n".
- **data_format**: a character string that decides whether the returned object is in long or wide format. Default is "wide", with the alternative supplied being "long".

**Value**  

Returns a tibble of extracted estimates of \( \Sigma \) created by the function `niw.post()` or `siberMVN()` in the packages `nicheROVER`, and `SIBER`.

The returned object will contain five columns in the following order when `data_format` is set to "wide", `metric`, `id`, `sample_name`, `isotope`, `sample_number`, and the posterior sample for \( \Sigma \) (e.g., `d13c` and `d15n`).

**See Also**  

`nicheROVER::niw.post()` and `SIBER::siberMVN()`
Examples

```r
evaluate_sigma(
  data = niw_fish_post
)

evaluate_sigma(
  data = post_sam_siber,
  pkg = "SIBER"
)
```

---

**mu_est_long**

A data.frame containing posterior estimates of \( \mu \)

### Description

Posterior estimates of \( \mu \) using fish data set from nicheROVER, using Normal-Inverse-Wishart (NIW) priors.

### Usage

```r
mu_est_long
```

### Format

data.frame containing 8,000 rows and 7 variables

- **metric**: name of the metric extracted from `niw.post()`
- **species**: species abbreviation
- **sample_number**: sample number from 1-1000
- **isotope**: column with isotope name
- **mu_est**: estimate of \( \mu \) produced from `niw.post()`
- **element**: isotopic element used in labelling
- **neutron**: neutron number used in labelling
**niche_ellipse**

Create ellipses based on Bayesian estimates of $\mu$ and $\Sigma$

### Description

This function allows the user to supply Bayesian estimates of $\mu$ and $\Sigma$ to create estimated Bayesian ellipse for niche region.

### Usage

```r
niche_ellipse(
  dat_mu,
  dat_sigma,
  isotope_a = NULL,
  isotope_b = NULL,
  p_ell = NULL,
  random = NULL,
  set_seed = NULL,
  n = NULL,
  message = TRUE
)
```

### Arguments

- **dat_mu**: a data.frame containing $\mu$ Bayesian estimates. This data.frame needs to be in long format with each $\mu$ estimate for each isotope stacked on top of each other. This can be produced using `extract_mu()`.
- **dat_sigma**: a data.frame containing $\Sigma$ Bayesian estimates. This data.frame needs be in wide format, that is $\Sigma$ (covariance) matrices stacked on top of each other. See example of how to convert to wide format. This can be produced using `extract_sigma()`.
- **isotope_a**: character string that is the column name of the first isotope used in `dat_sigma`. Defaults to "d13c".
- **isotope_b**: character string that is the column name of the second isotope used in `dat_sigma`. Defaults to "d15n".
- **p_ell**: is the confidence interval of each ellipse estimate. Default is 0.95 (i.e., 95% confidence interval). This value is bound by 0 and 1 and has to be a numeric.
- **random**: logical value indicating whether or not to randomly sample posterior distributions for $\mu$ and $\Sigma$ to create a sub-sample of ellipse. Default is `TRUE`.
- **set_seed**: numerical value to set seed for random sampling. Default is a random value. To consistently sample the same subsample, please supply a numerical value (e.g., 4). It is highly suggested to use `set_seed` to make the function results when randomly sampling reproducible.
- **n**: numerical value that controls the number of random samples. Default is 10.
- **message**: control whether the time processing is displayed after the end of the function. Default is `TRUE`. 

---

The `niche_ellipse` function allows users to create Bayesian ellipses based on Bayesian estimates of $\mu$ and $\Sigma$. The function requires the user to supply data frames containing Bayesian estimates of $\mu$ and $\Sigma$, as well as the names of the isotopes used to calculate the covariance matrices. The function also allows for customization of the confidence interval and the number of random samples to be used for creating the ellipses. The function's arguments include `dat_mu` and `dat_sigma` for Bayesian estimates of $\mu$ and $\Sigma$, respectively, `isotope_a` and `isotope_b` for the column names of the first and second isotopes used in `dat_sigma`, `p_ell` for the confidence interval of each ellipse, `random` for controlling whether to randomly sample posterior distributions, `set_seed` for setting the seed for random sampling, `n` for controlling the number of random samples, and `message` for controlling the display of processing time. The function is designed to be flexible, allowing users to customize their ellipse calculations according to their specific needs.
Value

A tibble containing sample_name, sample_number, and the isotopes that were used in the estimation of ellipse (i.e., and d13c and d15n).

See Also

nicheROVER::niw.post(), SIBER::siberMVN(), extract_mu(), and extract_sigma()

Examples

niche_ellipse(dat_mu = mu_est_long,
              dat_sigma = sigma_est_wide)

niw_fish_post       A list of the posterior estimates of μ and Σ from {nicheROVER}

Description

Posterior estimates of μ and Σ using the fish data set from nicheROVER, using Normal-Inverse-Wishart (NIW) priors. This list is produced using the function niw.post() from nicheROVER.

Usage

niw_fish_post

Format

A list with elements μ and Σ of sizes c(nsamples, length(lambda)) and c(dim(Psi)).

over_stat         A data.frame containing the estimates of percentage of overlap among groups

Description

Estimates of the percentage of overlap among example species used in nicheROVER.

Usage

over_stat

Format

A array containing matrices of the percent overlap for each group used in Bayesian estimates of μ and Σ using Normal-Inverse-Wishart (NIW) priors calculated in niw.post().
**post_sam_siber**

A list of the posterior estimates of $\mu$ and $\Sigma$ from \{SIBER\}

---

**Description**

Posterior estimates of $\mu$ and $\Sigma$ using the demo.siber.data data set from \texttt{SIBER}. This list is produced using the function \texttt{siberMVN()} from \texttt{SIBER}.

**Usage**

post_sam_siber

**Format**

A list with estimates of $\mu$ and $\Sigma$ for each species and group.

---

**sigma_est_wide**

A data.frame containing posterior estimates of $\Sigma$

---

**Description**

Posterior estimates of $\Sigma$ using fish data set from \texttt{nicheROVER}, using Normal-Inverse-Wishart (NIW) priors

**Usage**

sigma_est_wide

**Format**

data.frame containing 8,000 rows and 6 variables

- **metric** name of the metric extracted from \texttt{niew.post()}
- **species** species abbreviation
- **isotope** column with isotope name
- **sample_number** sample number from 1-1000
- **d15n** estimate of $\Sigma$ for d15n produced from \texttt{niew.post()}
- **d13c** estimate of $\Sigma$ for d13c produced from \texttt{niew.post()}
Index

* datasets
  - mu_est_long, 6
  - niw_fish_post, 8
  - over_stat, 8
  - post_sam_siber, 9
  - sigma_est_wide, 9

extract_mu, 2
extract_mu(), 8
extract_niche_size, 3
extract_overlap, 4
extract_sigma, 5
extract_sigma(), 8

mu_est_long, 6

niche_ellipse, 7
nicheROVER::niche.size(), 4
nicheROVER::niw.post(), 3–5, 8
nicheROVER::overlap(), 4
niw_fish_post, 8

over_stat, 8

post_sam_siber, 9

SIBER::siberMVN(), 3, 5, 8
sigma_est_wide, 9