

Package ‘occumb’

January 8, 2024

Title Site Occupancy Modeling for Environmental DNA Metabarcoding

Version 1.0.3

Description Fits multispecies site occupancy models to environmental DNA metabarcoding data collected using spatially-replicated survey design. Model fitting results can be used to evaluate and compare the effectiveness of species detection to find an efficient survey design.

Reference: Fukaya et al. (2022) <[doi:10.1111/2041-210X.13732](https://doi.org/10.1111/2041-210X.13732)>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests rmarkdown, spelling, testthat (>= 3.0.0)

Config/testthat/edition 3

Imports stats, methods, graphics, checkmate, crayon, knitr

Collate 'classes.R' 'fish.R' 'fish_raw.R' 'get_posterior.R' 'gof.R' 'occumbData.R' 'occumb.R' 'plot.R' 'show.R' 'summary.R' 'utility.R' 'utils.R'

URL <https://fukayak.github.io/occumb/>,
<https://github.com/fukayak/occumb>

Depends jagsUI, R (>= 2.10)

VignetteBuilder knitr

SystemRequirements JAGS (<http://mcmc-jags.sourceforge.net>)

BugReports <https://github.com/fukayak/occumb/issues>

Language en-US

NeedsCompilation no

Author Keiichi Fukaya [aut, cre],
Ken Kellner [cph] (summary method for occumbFit class)

Maintainer Keiichi Fukaya <fukayak99@gmail.com>

Repository CRAN

Date/Publication 2024-01-08 19:40:02 UTC

R topics documented:

eval_util_L	2
eval_util_R	5
fish	9
fish_raw	9
get_posterior	10
gof	12
list_cond_L	13
list_cond_R	14
occumb	15
occumbData	18
plot,occumbFit-method	20
plot,occumbGof-method	21
summary,occumbData-method	21
summary,occumbFit-method	22

Index	23
--------------	-----------

eval_util_L	<i>Expected utility for local species diversity assessments.</i>
-------------	--

Description

eval_util_L() evaluates the expected utility for the local species diversity assessment using Monte Carlo integration.

Usage

```
eval_util_L(
  settings,
  fit = NULL,
  z = NULL,
  theta = NULL,
  phi = NULL,
  N_rep = 1,
  cores = 1L
)
```

Arguments

settings	A data frame that specifies a set of conditions under which the utility is evaluated. It must include a column named K and N, which specifies the number of replicates per site and the sequencing depth per replicate, respectively. K and N must be numeric vectors greater than 0. When K contains a decimal, the decimal part is discarded and treated as an integer. Additional columns are ignored but may be included.
fit	An occumbFit object.

z	Sample values of site occupancy status of species stored in an array with sample \times species \times site dimension.
theta	Sample values of sequence capture probabilities of species stored in a matrix with sample \times species dimension or an array with sample \times species \times site dimension.
phi	Sample values of sequence relative dominance of species stored in a matrix with sample \times species dimension or an array with sample \times species \times site dimension.
N_rep	Controls the sample size for Monte Carlo integration. The integral is evaluated using a total of N_sample * N_rep random samples, where N_sample is the maximum size of the MCMC sample in the fit argument and the parameter sample in the z, theta, and phi arguments.
cores	The number of cores to use for parallelization.

Details

The utility of the local species diversity assessment can be defined, for a given set of sites, as the expected number of detected species per site (Fukaya et al. 2022). `eval_util_L()` evaluates this utility for arbitrary sets of sites that can potentially have different values for site occupancy status of species, z , sequence capture probabilities of species, θ , and sequence relative dominance of species, ϕ , for the combination of K and N values specified in the `conditions` argument. Such evaluations can be used to balance K and N to maximize the utility under a constant budget (possible combinations of K and N under specified budget and cost values are easily obtained using `list_cond_L()`; see the example below). It is also possible to examine how the utility varies with different K and N values without setting a budget level, which may be useful to know the satisfactory level of K and N from a purely technical point of view.

The expected utility is defined as the expected value of the conditional utility of the form:

$$U(K, N | \mathbf{r}, \mathbf{u}) = \frac{1}{J} \sum_{j=1}^J \sum_{i=1}^I \left\{ 1 - \prod_{k=1}^K \left(1 - \frac{u_{ijk} r_{ijk}}{\sum_{m=1}^I u_{mjk} r_{mjk}} \right)^N \right\}$$

where u_{ijk} is a latent indicator variable representing the inclusion of the sequence of species i in replicate k at site j and r_{ijk} is a latent variable that is proportional to the relative frequency of the sequence of species i , conditional on its presence in replicate k at site j (Fukaya et al. 2022). Expectations are taken with respect to posterior (or, possibly prior) predictive distributions of $\mathbf{r} = \{r_{ijk}\}$ and $\mathbf{u} = \{u_{ijk}\}$, which are evaluated numerically by Monte Carlo integration. The predictive distributions of \mathbf{r} and \mathbf{u} depend on the model parameters z , θ , and ϕ values. Their posterior (or prior) distribution is specified by supplying an `occumbFit` object containing their posterior samples via the `fit` argument or by supplying a matrix or array of posterior (or prior) samples of parameter values via the `z`, `theta`, and `phi` arguments. Higher approximation accuracy can be obtained by increasing the value of `N_rep`.

The `eval_util_L()` function can be executed by supplying the `fit` argument without specifying the `z`, `theta`, and `phi` arguments, by supplying the three `z`, `theta`, and `phi` arguments without the `fit` argument, or by supplying the `fit` argument and any or all of the `z`, `theta`, and `phi` arguments. If `z`, `theta`, or `phi` arguments are specified in addition to `fit`, the parameter values given in these

arguments are used preferentially to evaluate expected utility. If sample sizes differ among parameters, parameters with smaller sample sizes are resampled with replacement to align sample sizes across parameters.

The expected utility is evaluated assuming homogeneity of replicates, in the sense that θ and ϕ , the model parameters associated with the species detection process, are constant across replicates within a site. For this reason, `eval_util_L()` does not accept replicate-specific θ and ϕ . If the `occumbFit` object supplied in the `fit` argument has a replicate-specific parameter, parameter samples to be used in the utility evaluation must be provided explicitly via `theta` or `phi` argument.

Monte Carlo integration is executed in parallel on multiple CPU cores where the `cores` argument controls the degree of parallelization.

Value

A data frame with a column named `Utility` in which the estimates of expected utility are stored. This is obtained by adding the `Utility` column to the data frame provided in the `settings` argument.

References

K. Fukaya, N. I. Kondo, S. S. Matsuzaki and T. Kadoya (2022) Multispecies site occupancy modelling and study design for spatially replicated environmental DNA metabarcoding. *Methods in Ecology and Evolution* **13**:183–193. doi:10.1111/2041210X.13732

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
data <- occumbData(
  y = array(sample.int(I * J * K), dim = c(I, J, K)))

# Fitting a null model
fit <- occumb(data = data)

## Estimate expected utility
# Arbitrary K and N values
(util1 <- eval_util_L(expand.grid(K = 1:3, N = c(1E3, 1E4, 1E5)),
  fit))

# K and N values under specified budget and cost
(util2 <- eval_util_L(list_cond_L(budget = 1E5,
  lambda1 = 0.01,
  lambda2 = 5000,
  fit),
  fit))

# K values restricted
(util3 <- eval_util_L(list_cond_L(budget = 1E5,
  lambda1 = 0.01,
```

```

                                lambda2 = 5000,
                                fit,
                                K = 1:5),
                                fit))

# theta and phi values supplied
(util4 <- eval_util_L(list_cond_L(budget = 1E5,
                                lambda1 = 0.01,
                                lambda2 = 5000,
                                fit,
                                K = 1:5),
                                fit,
                                theta = array(0.5, dim = c(4000, 2, 2)),
                                phi = array(1, dim = c(4000, 2, 2))))

# z, theta, and phi values, but no fit object supplied
(util5 <- eval_util_L(list_cond_L(budget = 1E5,
                                lambda1 = 0.01,
                                lambda2 = 5000,
                                fit,
                                K = 1:5),
                                fit = NULL,
                                z = array(1, dim = c(4000, 2, 2)),
                                theta = array(0.5, dim = c(4000, 2, 2)),
                                phi = array(1, dim = c(4000, 2, 2))))

```

eval_util_R

Expected utility for regional species diversity assessments.

Description

eval_util_R() evaluates the expected utility for the regional species diversity assessment using Monte Carlo integration.

Usage

```

eval_util_R(
  settings,
  fit = NULL,
  psi = NULL,
  theta = NULL,
  phi = NULL,
  N_rep = 1,
  cores = 1L
)

```

Arguments

settings	A data frame that specifies a set of conditions under which the utility is evaluated. It must include a column named J, K, and N, which specifies the number of sites, the number of replicates per site, and the sequencing depth per replicate, respectively. J, K, and N must be numeric vectors greater than 0. When J and K contains a decimal, the decimal part is discarded and treated as an integer. Additional columns are ignored but may be included.
fit	An occumbFit object.
psi	Sample values of site occupancy probabilities of species stored in a matrix with sample \times species dimension or an array with sample \times species \times site dimension.
theta	Sample values of sequence capture probabilities of species stored in a matrix with sample \times species dimension or an array with sample \times species \times site dimension.
phi	Sample values of sequence relative dominance of species stored in a matrix with sample \times species dimension or an array with sample \times species \times site dimension.
N_rep	Controls the sample size for Monte Carlo integration. The integral is evaluated using a total of N_sample * N_rep random samples, where N_sample is the maximum size of the MCMC sample in the fit argument and the parameter sample in the psi, theta, and phi arguments.
cores	The number of cores to use for parallelization.

Details

The utility of the regional species diversity assessment can be defined as the number of species in the region of interest expected to be detected (Fukaya et al. 2022). `eval_util_R()` evaluates this utility for the region modeled in the `occumbFit` object for the combination of J, K, and N values specified in the `conditions` argument. Such evaluations can be used to balance J, K, and N to maximize the utility under a constant budget (possible combinations of J, K, and N under specified budget and cost values are easily obtained using `list_cond_R()`; see the example below). It is also possible to examine how the utility varies with different J, K, and N values without setting a budget level, which may be useful to know the satisfactory level of J, K, and N from a purely technical point of view.

The expected utility is defined as the expected value of the conditional utility of the form:

$$U(J, K, N | \mathbf{r}, \mathbf{u}) = \sum_{i=1}^I \left\{ 1 - \prod_{j=1}^J \prod_{k=1}^K \left(1 - \frac{u_{ijk} r_{ijk}}{\sum_{m=1}^I u_{mjk} r_{mjk}} \right)^N \right\}$$

where u_{ijk} is a latent indicator variable representing the inclusion of the sequence of species i in replicate k at site j and r_{ijk} is a latent variable that is proportional to the relative frequency of the sequence of species i , conditional on its presence in replicate k at site j (Fukaya et al. 2022). Expectations are taken with respect to the posterior (or, possibly prior) predictive distributions of $\mathbf{r} = \{r_{ijk}\}$ and $\mathbf{u} = \{u_{ijk}\}$, which are evaluated numerically by Monte Carlo integration. The predictive distributions of \mathbf{r} and \mathbf{u} depend on the model parameters ψ , θ , and ϕ values. Their posterior (or prior) distribution is specified by supplying an `occumbFit` object containing their posterior

samples via the `fit` argument or by supplying a matrix or array of posterior (or prior) samples of parameter values via the `psi`, `theta`, and `phi` arguments. Higher approximation accuracy can be obtained by increasing the value of `N_rep`.

The `eval_util_R()` function can be executed by supplying the `fit` argument without specifying the `psi`, `theta`, and `phi` arguments, by supplying the three `psi`, `theta`, and `phi` arguments without the `fit` argument, or by supplying the `fit` argument and any or all of the `psi`, `theta`, and `phi` arguments. If `psi`, `theta`, or `phi` arguments are specified in addition to `fit`, the parameter values given in these arguments are used preferentially to evaluate expected utility. If sample sizes differ among parameters, parameters with smaller sample sizes are resampled with replacement to align sample sizes across parameters.

The expected utility is evaluated assuming homogeneity of replicates, in the sense that θ and ϕ , the model parameters associated with the species detection process, are constant across replicates within a site. For this reason, `eval_util_R()` does not accept replicate-specific θ and ϕ . If the `occumbFit` object supplied in the `fit` argument has a replicate-specific parameter, parameter samples to be used in the utility evaluation must be provided explicitly via `theta` or `phi` argument.

If the parameters are modeled as a function of site covariates in the `fit` object, or if the `psi`, `theta`, and/or `phi` arguments have site dimension, the expected utility is evaluated to account for the site heterogeneity of the parameters. Specifically, to incorporate site heterogeneity, the parameter values for each `J` site are determined by selecting site-specific parameter values in the `fit` or that supplied in `psi`, `theta`, and `phi` via random sampling with replacement. Thus, the expected utility is evaluated by assuming the set of supplied parameter values as a statistical population of site-specific parameters.

Monte Carlo integration is executed in parallel on multiple CPU cores where the `cores` argument controls the degree of parallelization.

Value

A data frame with a column named `Utility` in which the estimates of expected utility are stored. This is obtained by adding the `Utility` column to the data frame provided in the `settings` argument.

References

K. Fukaya, N. I. Kondo, S. S. Matsuzaki and T. Kadoya (2022) Multispecies site occupancy modelling and study design for spatially replicated environmental DNA metabarcoding. *Methods in Ecology and Evolution* **13**:183–193. doi:10.1111/2041210X.13732

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
data <- occumbData(
  y = array(sample.int(I * J * K), dim = c(I, J, K)))

# Fitting a null model
fit <- occumb(data = data)
```

```

## Estimate expected utility
# Arbitrary J, K, and N values
(util1 <- eval_util_R(expand.grid(J = 1:3, K = 1:3, N = c(1E3, 1E4, 1E5)),
                      fit))

# J, K, and N values under specified budget and cost
(util2 <- eval_util_R(list_cond_R(budget = 50000,
                                  lambda1 = 0.01,
                                  lambda2 = 5000,
                                  lambda3 = 5000),
                      fit))

# K values restricted
(util3 <- eval_util_R(list_cond_R(budget = 50000,
                                  lambda1 = 0.01,
                                  lambda2 = 5000,
                                  lambda3 = 5000,
                                  K = 1:5),
                      fit))

# J and K values restricted
(util4 <- eval_util_R(list_cond_R(budget = 50000,
                                  lambda1 = 0.01,
                                  lambda2 = 5000,
                                  lambda3 = 5000,
                                  J = 1:3, K = 1:5),
                      fit))

# theta and phi values supplied
(util5 <- eval_util_R(list_cond_R(budget = 50000,
                                  lambda1 = 0.01,
                                  lambda2 = 5000,
                                  lambda3 = 5000,
                                  J = 1:3, K = 1:5),
                      fit,
                      theta = array(0.5, dim = c(4000, 2, 2)),
                      phi = array(1, dim = c(4000, 2, 2))))

# psi, theta, and phi values, but no fit object supplied
(util6 <- eval_util_R(list_cond_R(budget = 50000,
                                  lambda1 = 0.01,
                                  lambda2 = 5000,
                                  lambda3 = 5000,
                                  J = 1:3, K = 1:5),
                      fit = NULL,
                      psi = array(0.5, dim = c(4000, 2, 2)),
                      theta = array(0.5, dim = c(4000, 2, 2)),
                      phi = array(1, dim = c(4000, 2, 2))))

```

fish

Fish eDNA metabarcoding dataset

Description

A dataset of fish eDNA metabarcoding collected in the Kasumigaura watershed, Japan.

Usage

fish

Format

An `occumbData` class object containing sequence read count `y`, `mismatch` as a species covariate, and `riverbank` as a site covariate. `mismatch` represents the total number of mismatched bases in the priming region of the forward and reverse primers for each species. `riverbank` indicates whether the riverbank of each site lacks aquatic and riparian vegetation. The sequence reads were obtained from three replicates (collected from the center of the river and near the left and right riverbanks, respectively) from 50 sites across the watershed, of which read counts from 6 samples are missing. The resulting sequence counts of the 50 freshwater fish taxa detected are recorded.

Source

K. Fukaya, N. I. Kondo, S. S. Matsuzaki, T. Kadoya (2021) Data from: Multispecies site occupancy modeling and study design for spatially replicated environmental DNA metabarcoding. Dryad Digital Repository. [doi:10.5061/dryad.3bk3j9kkm](https://doi.org/10.5061/dryad.3bk3j9kkm)

fish_raw

Fish eDNA metabarcoding dataset

Description

A dataset of fish eDNA metabarcoding collected in the Kasumigaura watershed, Japan.

Usage

fish_raw

Format

A list containing an array of sequence read count `y`, a vector of the total number of mismatched bases in the priming region of the forward and reverse primers for each species `mismatch`, and a factor indicating whether the riverbank of each site lacks aquatic and riparian vegetation `riverbank`. The sequence reads were obtained from three replicates (collected from the center of the river and near the left and right riverbanks, respectively) from 50 sites across the watershed, of which read counts from 6 samples are missing. The resulting sequence counts of the 50 freshwater fish taxa detected are recorded.

Source

K. Fukaya, N. I. Kondo, S. S. Matsuzaki, T. Kadoya (2021) Data from: Multispecies site occupancy modeling and study design for spatially replicated environmental DNA metabarcoding. Dryad Digital Repository. [doi:10.5061/dryad.3bk3j9kkm](https://doi.org/10.5061/dryad.3bk3j9kkm)

get_posterior	<i>Extract posterior samples or summary of parameters from a model-fit object.</i>
---------------	--

Description

get_post_samples() extracts posterior samples of the specified parameters from a model-fit object.

get_post_summary() extracts posterior summary of the specified parameters from a model-fit object.

Usage

```
get_post_samples(
  fit,
  parameter = c("z", "pi", "phi", "theta", "psi", "alpha", "beta", "gamma",
    "alpha_shared", "beta_shared", "gamma_shared", "Mu", "sigma", "rho")
)

get_post_summary(
  fit,
  parameter = c("z", "pi", "phi", "theta", "psi", "alpha", "beta", "gamma",
    "alpha_shared", "beta_shared", "gamma_shared", "Mu", "sigma", "rho")
)
```

Arguments

fit	An occumbFit object.
parameter	A string of parameter name. See Details for possible choices and corresponding parameters.

Details

The functions return posterior samples or summary of one of the following parameters in the model, stored in the model-fit object fit:

z Site occupancy status of species.

pi Multinomial probabilities of species sequence read counts.

phi Sequence relative dominance of species.

theta Sequence capture probabilities of species.

psi Site occupancy probabilities of species.
 alpha Species-specific effects on sequence relative dominance (ϕ).
 beta Species-specific effects on sequence capture probabilities (θ).
 gamma Species-specific effects on site occupancy probabilities (ψ).
 alpha_shared Effects on sequence relative dominance (ϕ) common across species.
 beta_shared Effects on sequence capture probabilities (θ) that are common across species.
 gamma_shared Effects on site occupancy probabilities (ψ) that are common across species.
 Mu Community-level averages of the species-specific effects (alpha, beta, gamma).
 sigma Standard deviations of the species-specific effects (alpha, beta, gamma).
 rho Correlation coefficients of the species-specific effects (alpha, beta, gamma).

See [the package vignette](#) for details of these parameters.

The parameter may have dimensions corresponding to species, sites, replicates, and effects (covariates): the `dimension` and `label` attributes are added to the output object to inform these dimensions. If the sequence read count data `y` has species, site, or replicate names appended as the `dimnames` attribute (see Details in [occumbData\(\)](#)), they will be copied into the `label` attribute of the returned object.

Value

`get_post_samples()` returns a vector, matrix, or array of posterior samples of the selected parameter.

`get_post_summary()` returns a table (matrix) of posterior summary of the selected parameter. The elements of the posterior summary are the same as those obtained with the `jags()` function in the `jagsUI` package: they include mean, standard deviation, percentiles of posterior samples; the R_{hat} statistic; the effective sample size, `n.eff`; `overlap0` that checks if 0 falls in the parameter's 95% credible interval; and the proportion of the posterior with the same sign as the mean, `f`.

The `dimension` and `label` attributes of the output object give information about the dimension of the parameter.

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
y_named <- array(sample.int(I * J * K), dim = c(I, J, K))
dimnames(y_named) <- list(c("species 1", "species 2"),
                          c("site 1", "site 2"), NULL)
data_named <- occumbData(y = y_named)

# Fitting a null model
fit <- occumb(data = data_named, n.iter = 10100)

# Extract posterior samples
(post_sample_z <- get_post_samples(fit, "z"))
```

```
# Look dimensions of the parameter
attributes(post_sample_z)

# Extract posterior summary
(post_summary_z <- get_post_summary(fit, "z"))
# Look dimensions of the parameter
attributes(post_summary_z)
```

gof

*Goodness-of-fit assessment of the fitted model.***Description**

`gof()` calculates some omnibus discrepancy measures and their Bayesian p -values for the fitted model using the posterior predictive check approach.

Usage

```
gof(fit, stats = c("Freeman_Tukey", "deviance"), cores = 1L, plot = TRUE, ...)
```

Arguments

<code>fit</code>	An <code>occumbFit</code> object.
<code>stats</code>	The discrepancy statistics to be applied.
<code>cores</code>	The number of cores to use for parallelization.
<code>plot</code>	Logical, determine if draw scatter plots of the fit statistics.
<code>...</code>	Additional arguments passed to the default <code>plot</code> method.

Details

A discrepancy statistic of the fitted model is obtained using the procedure of posterior predictive checking. The following statistics are currently available:

Freeman-Tukey statistics (default) $T_{FT} = \sum_{i,j,k} \left(\sqrt{y_{ijk}} - \sqrt{E(y_{ijk} | \pi_{ijk})} \right)^2$

Deviance statistics $T_{\text{deviance}} = -2 \sum_{j,k} \log \text{Multinomial}(\mathbf{y}_{jk} | \boldsymbol{\pi}_{jk})$

where i , j , and k are the subscripts of species, site, and replicate, respectively, y_{ijk} is sequence read count data, π_{ijk} is multinomial cell probabilities of sequence read counts, $E(y_{ijk} | \pi_{ijk})$ is expected value of the sequence read counts conditional on their cell probabilities, and $\log \text{Multinomial}(\mathbf{y}_{jk} | \boldsymbol{\pi}_{jk})$ is the multinomial log-likelihood of the sequence read counts in replicate k of site j conditional on their cell probabilities.

The Bayesian p -value is estimated as the probability that the value of the discrepancy statistics of replicated dataset is more extreme than that of the observed dataset. An extreme Bayesian p -value may indicate an inadequate model fit. See, e.g., Gelman et al. (2014), Kéry and Royle (2016), and Conn et al. (2018) for more details on the procedures for posterior predictive checking.

Computations can be run in parallel on multiple CPU cores where the `cores` argument controls the degree of parallelization.

Value

A list with the following named elements:

stats The discrepancy statistics applied.

p_value Bayesian p -value.

stats_obs Discrepancy statistics for observed data.

stats_rep Discrepancy statistics for repeated data.

References

P. B. Conn, D. S. Johnson, P. J. Williams, S. R. Melin and M. B. Hooten. (2018) A guide to Bayesian model checking for ecologists. *Ecological Monographs* **88**:526–542. doi:10.1002/ecm.1314

A. Gelman, J. B. Carlin, H. S. Stern D. B. Dunson, A. Vehtari and D. B. Rubin (2013) *Bayesian Data Analysis*. 3rd edition. Chapman and Hall/CRC. <http://www.stat.columbia.edu/~gelman/book/>

M. Kéry and J. A. Royle (2016) *Applied Hierarchical Modeling in Ecology — Analysis of Distribution, Abundance and Species Richness in R and BUGS. Volume 1: Prelude and Static Models*. Academic Press. <https://www.mbr-pwrc.usgs.gov/pubanalysis/keryroylebook/>

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
data <- occumbData(
  y = array(sample.int(I * J * K), dim = c(I, J, K)),
  spec_cov = list(cov1 = rnorm(I)),
  site_cov = list(cov2 = rnorm(J),
                 cov3 = factor(1:J)),
  repl_cov = list(cov4 = matrix(rnorm(J * K), J, K))

# Fitting a null model
fit <- occumb(data = data)

# Goodness-of-fit assessment
gof_result <- gof(fit)
gof_result
```

list_cond_L

Conditions for local assessment under certain budget and cost values.

Description

list_cond_L() constructs a list of possible local species diversity assessment conditions under specified budget and cost values.

Usage

```
list_cond_L(budget, lambda1, lambda2, fit, K = NULL)
```

Arguments

budget	A numeric specifying the amount of budget. Use the currency unit consistent with lambda1 and lambda2.
lambda1	A numeric specifying the cost per sequence read for high-throughput sequencing. Use the currency unit consistent with budget and lambda2.
lambda2	A numeric specifying the cost per replicate for library preparation. Use the currency unit consistent with budget and lambda1.
fit	An occumbFit object.
K	An optional vector for manually specifying the number of replicates.

Details

This function can generate a data frame object to be given to the `settings` argument of `eval_util_L()`; see Examples of `eval_util_L()`. By default, it outputs a list of all feasible combinations of values for the number of replicates per site K and the sequencing depth per replicate N , based on the given budget and cost values and the number of sites (identified by reference to the `fit` object). The resulting N can be non-integer because it is calculated simply by assuming that we can obtain its maximum value. If you want to obtain a list for only a subset of the possible K values under a given budget and cost values, use the `K` argument to provide a vector of the desired K values.

Value

A data frame containing columns named `budget`, `lambda1`, `lambda2`, `K`, and `N`.

list_cond_R	<i>Conditions for regional assessment under certain budget and cost values.</i>
-------------	---

Description

`list_cond_R()` constructs a list of possible regional species diversity assessment conditions under specified budget and cost values.

Usage

```
list_cond_R(budget, lambda1, lambda2, lambda3, J = NULL, K = NULL)
```

Arguments

budget	A numeric specifying the amount of budget. Use the currency unit consistent with lambda1, lambda2, and lambda3.
lambda1	A numeric specifying the cost per sequence read for high-throughput sequencing. Use the currency unit consistent with budget, lambda2, and lambda3.
lambda2	A numeric specifying the cost per replicate for library preparation. Use the currency unit consistent with budget, lambda1, and lambda3.
lambda3	A numeric specifying the visiting cost per site. Use the currency unit consistent with budget, lambda1, and lambda2.
J	An optional vector for manually specifying the number of sites
K	An optional vector for manually specifying the number of replicates. For computational convenience, K values must be in ascending order.

Details

This function can generate a data frame object to be given to the `settings` argument of `eval_util_R()`; see Examples of `eval_util_R()`. By default, it outputs a list of all feasible combinations of values for the number of sites J , number of replicates per site K , and the sequencing depth per replicate N , based on the given budget and cost values. The resulting N can be non-integer because it is calculated simply by assuming that we can obtain its maximum value. If you want to obtain a list for only a subset of the possible values of J and K under a given budget and cost values, use the J and/or K arguments (in fact, it is recommended that a relatively small number of K values be specified using the K argument because the list of all conditions achievable under moderate budget and cost values can be huge, and it is rarely practical to have a vast number of replicates per site). If a given combination of J and K values is not feasible under the specified budget and cost values, that combination will be ignored and excluded from the output.

Value

A data frame containing columns named `budget`, `lambda1`, `lambda2`, `lambda3`, `J`, `K`, and `N`.

occumb

Model-fitting function.

Description

`occumb()` fits the multispecies site-occupancy model for eDNA metabarcoding (Fukaya et al. 2022) and returns a model-fit object containing posterior samples.

Usage

```
occumb(
  formula_phi = ~1,
  formula_theta = ~1,
  formula_psi = ~1,
```

```

    formula_phi_shared = ~1,
    formula_theta_shared = ~1,
    formula_psi_shared = ~1,
    prior_prec = 1e-04,
    prior_ulim = 10000,
    data,
    n.chains = 4,
    n.adapt = NULL,
    n.burnin = 10000,
    n.thin = 10,
    n.iter = 20000,
    parallel = FALSE,
    ...
)

```

Arguments

<code>formula_phi</code>	A right-hand side formula describing species-specific effects of sequence relative dominance (ϕ).
<code>formula_theta</code>	A right-hand side formula describing species-specific effects of sequence capture probability (θ).
<code>formula_psi</code>	A right-hand side formula describing species-specific effects of occupancy probability (ψ).
<code>formula_phi_shared</code>	A right-hand side formula describing effects of sequence relative dominance (ϕ) that are common across species. The intercept term is ignored (see Details).
<code>formula_theta_shared</code>	A right-hand side formula describing effects of sequence capture probability (θ) that are common across species. The intercept term is ignored (see Details).
<code>formula_psi_shared</code>	A right-hand side formula describing effects of occupancy probability (ψ) that are common across species. The intercept term is ignored (see Details).
<code>prior_prec</code>	Precision of the normal prior distribution for the community-level average of species-specific parameters and effects common across species.
<code>prior_ulim</code>	Upper limit of the uniform prior distribution for the standard deviation of species-specific parameters.
<code>data</code>	A dataset supplied as an <code>occumbData</code> class object.
<code>n.chains</code>	Number of Markov chains to run.
<code>n.adapt</code>	Number of iterations to run in the JAGS adaptive phase.
<code>n.burnin</code>	Number of iterations at the beginning of the chain to discard.
<code>n.thin</code>	Thinning rate. Must be a positive integer.
<code>n.iter</code>	Total number of iterations per chain (including burn-in).
<code>parallel</code>	If TRUE, run MCMC chains in parallel on multiple CPU cores.
<code>...</code>	Additional arguments passed to <code>jags()</code> function.

Details

`occumb()` allows the fitting of a range of the multispecies site occupancy models including covariates at different levels of the data generation process. The most general form of the model can be written as follows (notation follows the original article: see References):

Sequence read counts:

$$(y_{1jk}, \dots, y_{Ijk}) \sim \text{Multinomial}((\pi_{1jk}, \dots, \pi_{Ijk}), N_{jk}),$$

$$\pi_{ijk} = \frac{u_{ijk} r_{ijk}}{\sum_m u_{mjk} r_{mjk}},$$

Relative frequency of species sequences:

$$r_{ijk} \sim \text{Gamma}(\phi_{ijk}, 1),$$

Capture of species sequences:

$$u_{ijk} \sim \text{Bernoulli}(z_{ij}\theta_{ijk}),$$

Site occupancy of species:

$$z_{ij} \sim \text{Bernoulli}(\psi_{ij}),$$

where the variations of ϕ , θ , and ψ are modeled by specifying model formulas in `formula_phi`, `formula_theta`, `formula_psi`, `formula_phi_shared`, `formula_theta_shared`, and `formula_psi_shared`.

Each parameter may have species-specific effects and effects that are common across species, where the former is specified by `formula_phi`, `formula_theta`, and `formula_psi`, while `formula_phi_shared`, `formula_theta_shared`, and `formula_psi_shared` specify the latter. Because species-specific intercepts are specified by default, the intercept term in the `formula_phi_shared`, `formula_theta_shared`, and `formula_psi_shared` are always ignored. The covariate terms must be found in the names of the list elements stored in `spec_cov`, `site_cov`, or `repl_cov` slots in the dataset object provided with the data argument. Covariates are modeled using the log link function for ϕ and logit link function for θ and ψ .

The two arguments, `prior_prec` and `prior_ulim`, control the prior distribution of parameters. For the community-level average of species-specific effects and for effects common across species, a normal prior distribution with mean 0 and precision (i.e., the inverse of the variance) `prior_prec` is specified. For the standard deviation of species-specific effects, a uniform prior distribution with a lower limit of 0 and an upper limit of `prior_ulim` is specified. For the correlation coefficient of species-specific effects, a uniform prior distribution in the range of -1 to 1 is specified by default.

Also see [the package vignette](#) for details on model specification in `occumb()`.

The data argument requires a dataset object generated using `ocumbData()`: see the document of [ocumbData\(\)](#).

The model is fit via the `jags()` function of the `jagsUI` package, where Markov chain Monte Carlo methods are used to obtain posterior samples of parameters and latent variables. Arguments `n.chains`, `n.adapt`, `n.burnin`, `n.thin`, `n.iter`, and `parallel` are passed on to the argument of the same name in the `jags()` function. See the document of `jagsUI`'s `jags()` function for details.

Value

An S4 object of the `occumbFit` class containing the results of model fitting and the supplied dataset.

References

K. Fukaya, N. I. Kondo, S. S. Matsuzaki and T. Kadoya (2022) Multispecies site occupancy modelling and study design for spatially replicated environmental DNA metabarcoding. *Methods in Ecology and Evolution* **13**:183–193. doi:10.1111/2041210X.13732

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
data <- occumbData(
  y = array(sample.int(I * J * K), dim = c(I, J, K)),
  spec_cov = list(cov1 = rnorm(I)),
  site_cov = list(cov2 = rnorm(J),
                 cov3 = factor(1:J)),
  repl_cov = list(cov4 = matrix(rnorm(J * K), J, K)))

# Fitting a null model (includes only species-specific intercepts)
res0 <- occumb(data = data)

# Add species-specific effects of site covariates in occupancy probabilities
res1 <- occumb(formula_psi = ~ cov2, data = data) # Continuous covariate
res2 <- occumb(formula_psi = ~ cov3, data = data) # Categorical covariate
res3 <- occumb(formula_psi = ~ cov2 * cov3, data = data) # Interaction

# Add species covariate in the three parameters
# Note that species covariates are modeled as common effects
res4 <- occumb(formula_phi_shared = ~ cov1, data = data) # phi
res5 <- occumb(formula_theta_shared = ~ cov1, data = data) # theta
res6 <- occumb(formula_psi_shared = ~ cov1, data = data) # psi

# Add replicate covariates
# Note that replicate covariates can only be specified for theta and phi
res7 <- occumb(formula_phi = ~ cov4, data = data) # phi
res8 <- occumb(formula_theta = ~ cov4, data = data) # theta

# Specify the prior distribution and MCMC settings explicitly
res9 <- occumb(data = data, prior_prec = 1E-2, prior_ulim = 1E2,
              n.chains = 1, n.burnin = 1000, n.thin = 1, n.iter = 2000)
res10 <- occumb(data = data, parallel = TRUE, n.cores = 2) # Run MCMC in parallel
```

occumbData

Constructor for occumbData data class.

Description

occumbData() creates a data list compatible with the model-fitting function [occumb\(\)](#).

Usage

```
occumbData(y, spec_cov = NULL, site_cov = NULL, repl_cov = NULL)
```

Arguments

<code>y</code>	A 3-D array of sequence read counts (integer values) that may have a <code>dimnames</code> attribute. Dimensions are ordered by species, site, and replicate. Data for missing replicates must be represented by zero vectors. NAs are not allowed.
<code>spec_cov</code>	A named list of species covariates. Each covariate can be a vector of continuous (numeric or integer) or discrete (logical, factor, or character) variables whose length equals <code>dim(y)[1]</code> (i.e., the number of species). NAs are not allowed.
<code>site_cov</code>	A named list of site covariates. Each covariate can be a vector of continuous (numeric or integer) or discrete (logical, factor, or character) variables whose length equals <code>dim(y)[1]</code> (i.e., the number of sites). NAs are not allowed.
<code>repl_cov</code>	A named list of replicate covariates. Each covariate can be a matrix of continuous (numeric or integer) or discrete (logical or character) variables whose dimension equals <code>dim(y)[2:3]</code> (i.e., the number of sites \times number of replicates). NAs are not allowed.

Details

The element (i.e., covariate) names for `spec_cov`, `site_cov`, and `repl_cov` must all be unique. If `y` has a `dimnames` attribute, it is retained in the resulting `occumbData` object and can be referenced in subsequent analyses.

Value

An S4 object of the `occumbData` class.

Examples

```
# Generate the smallest random dataset (2 species * 2 sites * 2 reps)
I <- 2 # Number of species
J <- 2 # Number of sites
K <- 2 # Number of replicates
data <- occumbData(
  y = array(sample.int(I * J * K), dim = c(I, J, K)),
  spec_cov = list(cov1 = rnorm(I)),
  site_cov = list(cov2 = rnorm(J), cov3 = factor(1:J)),
  repl_cov = list(cov4 = matrix(rnorm(J * K), J, K))
)

# A case for named y (with species and site names)
y_named <- array(sample.int(I * J * K), dim = c(I, J, K))
dimnames(y_named) <- list(c("common species", "uncommon species"),
  c("good site", "bad site"), NULL)
data_named <- occumbData(
  y = y_named,
```

```

spec_cov = list(cov1 = rnorm(I)),
site_cov = list(cov2 = rnorm(J), cov3 = factor(1:J)),
repl_cov = list(cov4 = matrix(rnorm(J * K), J, K))
)

# A real data example
data(fish_raw)
fish <- occumbData(
  y = fish_raw$y,
  spec_cov = list(mismatch = fish_raw$mismatch),
  site_cov = list(riverbank = fish_raw$riverbank)
)

# Get an overview of the datasets
summary(data)
summary(data_named)
summary(fish)

```

plot,occumbFit-method *Plot method for occumbFit class.*

Description

Applies `jagsUI`'s plot method to an `occumbFit` object to draw trace plots and density plots of MCMC samples of model parameters.

Usage

```
## S4 method for signature 'occumbFit'
plot(x, y = NULL, ...)
```

Arguments

<code>x</code>	An <code>occumbFit</code> object.
<code>y</code>	NULL
<code>...</code>	Additional arguments passed to the plot method for <code>jagsUI</code> object.

Value

Returns NULL invisibly.

plot,occumbGof-method *Plot method for occumbGof class.*

Description

Draws a scatter plot of fit statistics.

Usage

```
## S4 method for signature 'occumbGof'  
plot(x, y = NULL, ...)
```

Arguments

x	An occumbGof object.
y	NULL
...	Additional arguments passed to the default plot method.

Value

Returns NULL invisibly.

summary,occumbData-method
Summary method for occumbData class.

Description

Summarizes dataset stored in an occumbData object.

Usage

```
## S4 method for signature 'occumbData'  
summary(object)
```

Arguments

object	An occumbData object.
--------	-----------------------

Value

Returns NULL invisibly.

summary,occumbFit-method

Summary method for occumbFit class.

Description

Summarizes model fitting result stored in an occumbFit object.

Usage

```
## S4 method for signature 'occumbFit'  
summary(object)
```

Arguments

object An occumbFit object.

Value

Returns NULL invisibly.

Index

* datasets

fish, [9](#)

fish_raw, [9](#)

eval_util_L, [2](#)

eval_util_R, [5](#)

fish, [9](#)

fish_raw, [9](#)

get_post_samples (get_posterior), [10](#)

get_post_summary (get_posterior), [10](#)

get_posterior, [10](#)

gof, [12](#)

jags, [11](#), [16](#), [17](#)

list_cond_L, [13](#)

list_cond_R, [14](#)

occumb, [15](#), [18](#)

occumbData, [11](#), [16](#), [17](#), [18](#)

plot, occumbFit-method, [20](#)

plot, occumbGof-method, [21](#)

summary, occumbData-method, [21](#)

summary, occumbFit-method, [22](#)