Package ‘occumb’

July 22, 2025

Title Site Occupancy Modeling for Environmental DNA Metabarcoding

Version 1.2.1

Description Fits community 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>,
Fukaya and Hasebe (2025) <doi:10.1002/1438-390X.12219>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests covr, rmarkdown, spelling, testthat (>= 3.0.0), tibble, vdiffr

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' 'predict.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),
Mika Takahashi [aut]

Maintainer Keiichi Fukaya <fukayak99@gmail.com>

Repository CRAN

Date/Publication 2025-07-16 09:30:09 UTC

Contents

`eval_util_L` *Expected utility for local species diversity assessments.*

Description

eval_util_L() evaluates the expected utility of a local species diversity assessment by 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 utility is evaluated. It must include columns named K and N, which specify 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 value, it 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 × species × site dimensions.
theta	Sample values of sequence capture probabilities of species stored in a matrix with sample × species dimensions or an array with sample × species × site dimensions.
phi	Sample values of sequence relative dominance of species stored in a matrix with sample × species dimensions or an array with sample × species × site dimensions.
N_rep	Controls the sample size for the Monte Carlo integration. The integral is evaluated using 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 local species diversity assessment for a given set of sites can be defined 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 a 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 for determining a 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 in the form:

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

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 the posterior (or possibly prior) predictive distributions of $\boldsymbol{r} = \{r_{ijk}\}$ and $\boldsymbol{u} = \{u_{ijk}\}$, which are evaluated numerically using Monte Carlo integration. The predictive distributions of $\boldsymbol{r}$ and $\boldsymbol{u}$ 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 the fit, the parameter values given in these arguments are used preferentially to evaluate the expected utility. If the sample sizes differ among parameters, parameters with smaller sample sizes are resampled with replacements to align the 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, the parameter samples to be used in the utility evaluation must be provided explicitly via the theta or phi arguments.

The 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 the 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

```
set.seed(1)

# Generate a random dataset (20 species * 2 sites * 2 reps)
I <- 20 # 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, I, J)),
                      phi = array(1, dim = c(4000, I, J))))

# 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, I, J)),
                      theta = array(0.5, dim = c(4000, I, J)),
                      phi = array(1, dim = c(4000, I, J))))
```

eval_util_R *Expected utility for regional species diversity assessments.*

Description

eval_util_R() evaluates the expected utility of a 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 utility is evaluated. It must include columns named J, K, and N, which specify the number of sites, number of replicates per site, and sequencing depth per replicate, respectively. J, K, and N must be numeric vectors greater than 0. When J and K contain decimal values, they are discarded and treated as integers. Additional columns are ignored, but may be included.

fit — An occumbFit object.

psi — Sample values of the site occupancy probabilities of species stored in a matrix with sample × species dimensions or an array with sample × species × site dimensions.

theta — Sample values of sequence capture probabilities of species stored in a matrix with sample × species dimensions or an array with sample × species × site dimensions.

phi — Sample values of sequence relative dominance of species stored in a matrix with sample × species dimensions or an array with sample × species × site dimensions.

N_rep — Controls the sample size for the 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 a regional species diversity assessment can be defined as the number of species expected to be detected in the region of interest (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 a 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 in determining the satisfactory levels 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 in the form:

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

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 $\boldsymbol{r} = \{r_{ijk}\}$ and $\boldsymbol{u} = \{u_{ijk}\}$, which are evaluated numerically using Monte Carlo integration. The predictive distributions of $\boldsymbol{r}$ and $\boldsymbol{u}$ 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 the `psi`, `theta`, or `phi` arguments are specified in addition to the `fit`, the parameter values given in these arguments are preferentially used to evaluate the expected utility. If the sample sizes differed among parameters, parameters with smaller sample sizes are resampled with replacements to align the 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, the parameter samples to be used in the utility evaluation must be provided explicitly via the `theta` or `phi` arguments.

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 dimensions, the expected utility is evaluated to account for the site heterogeneity of the parameters. To incorporate site heterogeneity, the parameter values for each `J` site are determined by selecting site-specific parameter values in the `fit`, or those supplied in `psi`, `theta`, and `phi` via random sampling with replacement. Thus, expected utility is evaluated by assuming a set of supplied parameter values as a statistical population of site-specific parameters.

The 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 the 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

```
set.seed(1)
```

```
# Generate a random dataset (20 species * 2 sites * 2 reps)
I <- 20 # 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, I, J)),
                      phi = array(1, dim = c(4000, I, J))))

# 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(0.9, dim = c(4000, I, J)),
                      theta = array(0.9, dim = c(4000, I, J)),
                      phi = array(1, dim = c(4000, I, J))))
```

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 the sequence read count `y`, a species covariate `mismatch`, and a site covariate `riverbank`. `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 at each site lacks aquatic and riparian vegetation. Sequence reads were obtained from three replicates (collected from the center of the river and near the left and right riverbanks) from 50 sites across the watershed, of which read counts from six samples were missing. The resulting sequence counts of 50 freshwater fish taxa were 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

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 at each site lacked aquatic and riparian vegetation, riverbank. Sequence reads were obtained from three replicates (collected from the center of the river and near the left and right riverbanks) from 50 sites across the watershed, of which read counts from six samples were missing. The resulting sequence counts of 50 freshwater fish taxa detected were 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

---

get_posterior *Extract posterior samples or summary of parameters from a model-fit object.*

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"),
  output_dataframe = FALSE
)

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

Arguments

fit An occumbFit object.

parameter A string of parameter name. See Details for possible choices and corresponding parameters.

`output_dataframe`
If `TRUE`, results are returned in data frame format.

Details

The functions return posterior samples or a 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 (`phi`).

`beta` Species-specific effects on sequence capture probabilities (`theta`).

`gamma` Species-specific effects on site occupancy probabilities (`psi`).

`alpha_shared` Effects on sequence relative dominance (`phi`) common across species.

`beta_shared` Effects on sequence capture probabilities (`theta`) that are common across species.

`gamma_shared` Effects on site occupancy probabilities (`psi`) that are common across species.

`Mu` Community-level averages of species-specific effects (`alpha`, `beta`, `gamma`).

`sigma` Standard deviations of 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), and when `output_dataframe = FALSE`, the `dimension` and `label` attributes are added to the output object to inform these dimensions. If the sequence read count data `y` have species, site, or replicate names appended as the `dimnames` attribute (see Details in `occumbData()`), they are copied into the `label` attribute of the returned object.

Value

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

`get_post_summary()` returns, by default, a table (matrix) of the posterior summary of the selected parameters. The elements of the posterior summary are the same as those obtained with the `jags()` function in the `jagsUI` package: they include the mean, standard deviation, percentiles of posterior samples; the `Rhat` statistic; the effective sample size, `n.eff`; `overlap0`, which 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 provide information regarding the dimensions of the parameter.

When `output_dataframe = TRUE`, the results are returned in data frame format where the attributes obtained when `output_dataframe = FALSE` are incorporated into the table.

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 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", "chi_squared"),
  cores = 1L,
  plot = TRUE,
  ...
)
```

Arguments

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

Details

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

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

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

Chi-squared statistics $T_{\chi^2} = \sum_{i,j,k} \frac{(y_{ijk} - E(y_{ijk}|\pi_{ijk}))^2}{E(y_{ijk}|\pi_{ijk})}$

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, $E(y_{ijk} \mid \pi_{ijk})$ is expected value of the sequence read counts conditional on their cell probabilities, and $\log \text{Multinomial}(\boldsymbol{y}_{jk} \mid \boldsymbol{\pi}_{jk})$ 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 the replicated dataset is more extreme than that of the observed dataset. An extreme Bayesian p-value may indicate inadequate model fit. See Gelman et al. (2014), Kéry and Royle (2016), and Conn et al. (2018) for further details on the procedures used 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. `https://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 the specified budget and cost values.

Usage

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

Arguments

budget	A numeric specifying budget amount. The currency unit is arbitrary but must be consistent with that of lambda1 and lambda2.
lambda1	A numeric specifying the cost per sequence read for high-throughput sequencing. The currency unit is arbitrary but must be consistent with that of budget and lambda2.
lambda2	A numeric specifying the cost per replicate for library preparation. The currency unit is arbitrary but must be consistent with that of 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, cost values, and number of sites (identified by reference to the fit object). The resulting N can be a non-integer because it is calculated simply by assuming that the maximum value can be obtained. To obtain a list for only a subset of the possible K values under a given budget and cost value, the K argument is used 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` *Conditions for regional assessment under certain budget and cost values.*

Description

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

Usage

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

Arguments

`budget` A numeric specifying budget amount. The currency unit is arbitrary but must be consistent with that of `lambda1`, `lambda2`, and `lambda3`.

`lambda1` A numeric specifying the cost per sequence read for high-throughput sequencing. The currency unit is arbitrary but must be consistent with that of `budget`, `lambda2`, and `lambda3`.

`lambda2` A numeric specifying the cost per replicate for library preparation. The currency unit is arbitrary but must be consistent with that of `budget`, `lambda1`, and `lambda3`.

`lambda3` A numeric specifying the visiting cost per site. The currency unit is arbitrary but must be consistent with that of `budget`, `lambda1`, and `lambda2`.

`J` An optional vector for manually specifying the number of sites

`K` An optional vector used to specify the number of replicates manually. For computational convenience, the `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 sequencing depth per replicate `N` based on the given budget and cost values. The resulting `N` can be a non-integer because it is calculated simply by assuming that the maximum value can be obtained. If one wants to obtain a list for only a subset of the possible values of `J` and `K` under a given budget and cost value, 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 large, 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, the 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 community 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

`formula_phi` A right-hand side formula describing species-specific effects of sequence relative dominance (ϕ).

`formula_theta` A right-hand side formula describing species-specific effects of sequence capture probability (θ).

`formula_psi` A right-hand side formula describing species-specific effects of occupancy probability (ψ).

`formula_phi_shared`
A right-hand side formula describing effects of sequence relative dominance (ϕ) that are common across species. The intercept term is ignored (see Details).

`formula_theta_shared`
A right-hand side formula describing effects of sequence capture probability (θ) that are common across species. The intercept term is ignored (see Details).

`formula_psi_shared`
A right-hand side formula describing effects of occupancy probability (ψ) that are common across species. The intercept term is ignored (see Details).

`prior_prec` Precision of normal prior distribution for the community-level average of species-specific parameters and effects common across species.

`prior_ulim` Upper limit of uniform prior distribution for the standard deviation of species-specific parameters.

`data` A dataset supplied as an `occumbData` class object.

`n.chains` Number of Markov chains to run.

`n.adapt` Number of iterations to run in the JAGS adaptive phase.

`n.burnin` Number of iterations at the beginning of the chain to discard.

`n.thin` Thinning rate. Must be a positive integer.

`n.iter` Total number of iterations per chain (including burn-in).

`parallel` If TRUE, run MCMC chains in parallel on multiple CPU cores.

`...` Additional arguments passed to `jags()` function.

Details

`occumb()` allows the fitting of a range of community 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 (the notation follows that of the original article; see Fukaya et al. (2022) or the package vignette).

Sequence read counts:

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

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

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`, whereas `formula_phi_shared`, `formula_theta_shared`, and `formula_psi_shared` specify the latter. As species-specific intercepts are specified by default, the intercept terms in `formula_phi_shared`, `formula_theta_shared`, and `formula_psi_shared` are always ignored. Covariate terms must be found in the names of the list elements stored in the `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 effects common across species, a normal prior distribution with a mean of 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 zero 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.

See the package vignette for details on the model specifications in occumb().

The data argument requires a dataset object to be generated using ocumbData(); see the document of occumbData().

The model is fit using the jags() function of the jagsUI package, where Markov chain Monte Carlo (MCMC) methods are used to obtain posterior samples of the parameters and latent variables. Arguments n.chains, n.adapt, n.burnin, n.thin, n.iter, and parallel are passed on to arguments of the same name in the jags() function. See the document of jagsUI's jags() function for details. A set of random initial values is used to perform an MCMC run.

Value

An S4 object of the occumbFit class containing the results of the 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(). 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.

Usage

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

Arguments

y
: A 3-D array or a dataframe of sequence read counts (integer values). An array's dimensions are ordered by species, site, and replicate, and may have a dimnames attribute. A dataframe's columns are ordered by species, site, replicate, and sequence read counts. The data for missing replicates are represented by zero vectors. NAs are not allowed.

spec_cov
: 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 is dim(y)[1] (i.e., the number of species). The order of the species of the covariate values must correspond to that of the species dimension of y. NAs are not allowed.

site_cov
: 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 is dim(y)[1] (i.e., the number of sites). The order of the sites of the covariate values must correspond to that of the site dimension of y. NAs are not allowed.

repl_cov
: A named list of replicate covariates. Each covariate can be a matrix of continuous (numeric or integer) or discrete (logical or character) variables with dimensions equal to dim(y)[2:3] (i.e., number of sites × number of replicates). The order of the sites and replicates of the covariate values must correspond to that of the site and replicate dimensions of y. NAs are not allowed.

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

`x`	An `occumbFit` object.
`y`	`NULL`
`...`	Additional arguments passed to the plot method for jagsUI 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.

predict,occumbFit-method

Predict method for occumbFit class.

Description

Obtain predictions of parameters related to species occupancy and detection from an occumbFit model object.

Usage

```
## S4 method for signature 'occumbFit'
predict(
  object,
  newdata = NULL,
  parameter = c("phi", "theta", "psi"),
  scale = c("response", "link"),
  type = c("quantiles", "mean", "samples"),
  output_dataframe = FALSE
)
```

Arguments

object
: An occumbFit object.

newdata
: An optional occumbData object with covariates to be used for prediction. If omitted, the fitted covariates are used.

parameter
: The parameter to be predicted.

scale
: The scale on which the prediction is made. type = "response" returns the prediction on the original scale of the parameter. type = "link" returns the prediction on the link scale of the parameter.

type
: The type of prediction. type = "quantiles" returns 50% quantile as the posterior median of the prediction in addition to 2.5 and 97.5% quantiles as the lower and upper limits of the 95% credible interval of the prediction. type = "mean" returns the posterior mean of the prediction. type = "samples" returns the posterior samples of the prediction.

output_dataframe
: If TRUE, results are returned in data frame format.

Details

Applying predict() to an occumbFit object generates predictions for the specified parameter (phi, theta, or psi) based on the estimated effects and the given covariates. It is important to recognize that the predictions are specific to the individual species being modeled since they depend on the estimated species-specific effects (i.e., alpha, beta, and gamma; see the package vignette for details). When providing newdata, it must thus be assumed that the set of species contained in newdata is the same as that of the data being fitted.

Value

Predictions are obtained as a matrix or array that can have dimensions corresponding to statistics (or samples), species, sites, and replicates. 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.

When `output_dataframe = TRUE`, the results are returned in data frame format where the attributes obtained when `output_dataframe = FALSE` are incorporated into the table.

`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