Package ‘birp’

July 7, 2025

Type Package

Title Testing for Population Trends Using Low-Cost Ecological Count Data

Version 0.0.4

Date 2025-07-06

Description A Bayesian tool to test for population trends and changes in trends under arbitrary designs, including before-after (BA), control-intervention (CI) and before-after-control-intervention (BACI) designs commonly used to assess conservation impact. It infers changes in trends jointly from data obtained with multiple survey methods, as well as from limited and noisy data not necessarily collected in standardized ecological surveys. Observed counts can be modeled as following either a Poisson or a negative binomial model, and both deterministic and stochastic trend models are available. For more details on the model see Singer et al. (2025) <doi:10.1101/2025.01.08.631844>, and the file 'AUTHORS' for a list of copyright holders and contributors.

URL `https://bitbucket.org/wegmannlab/birpr/wiki/Home`

License GPL (>= 2)

Imports Rcpp (>= 1.0.12), MASS

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

RoxygenNote 7.3.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Madleina Caduff [aut, cre],
Daniel Wegmann [aut],
Liam Singer [aut],
Raphael Eckel [ctb],
Andreas Füglistaler [ctb]

Maintainer Madleina Caduff <`madleina.caduff@unifr.ch`>

Repository CRAN

Date/Publication 2025-07-06 23:20:05 UTC

Contents

`birp-package` *Testing for Population Trends Using Low-Cost Ecological Count Data*

Description

A Bayesian tool to test for population trends and changes in trends under arbitrary designs, including before-after (BA), control-intervention (CI) and before-after-control-intervention (BACI) designs commonly used to assess conservation impact. It infers changes in trends jointly from data obtained with multiple survey methods, as well as from limited and noisy data not necessarily collected in standardized ecological surveys. Observed counts can be modeled as following either a Poisson or a negative binomial model, and both deterministic and stochastic trend models are available. For more details on the model see Singer et al. (2025) <doi:10.1101/2025.01.08.631844>, and the file 'AUTHORS' for a list of copyright holders and contributors.

Package Content

Index of help topics:

```
assess_NB               Assess whether a Poisson model can replace the
                        Negative Binomial model
birp                    Create a 'birp' Object
birp-package            Testing for Population Trends Using Low-Cost
                        Ecological Count Data
birp_data               Create a birp_data Object from Count and Effort
                        Matrices
```

`birp_data_from_data_frame`	Create a 'birp_data' Object from a Data Frame or List of Data Frames
`birp_data_from_file`	Create a birp_data Object from File(s)
`birp_from_command_line`	Create a birp Object from Command-Line Output Files
`plot.birp`	Plot posterior distributions of gamma parameters
`plot.birp_data`	Plot a birp_data Object
`plot_epoch_pair`	Plot joint posterior of two gamma parameters
`plot_mcmc`	Plot MCMC Traces and Posterior Densities
`plot_trend`	Plot Posterior Trend Estimates
`print.birp`	Print a birp object
`print.birp_data`	Print a birp_data object Prints a summary of a 'birp_data' object, including the number of methods, locations, control-intervention (CI) groups, and timepoints, as well as the names or identifiers for each.
`simulate_birp`	Simulate Data for BIRP Models Generates simulated count data using the BIRP model framework with user-defined parameters.
`simulate_birp_from_results`	This function simulates a birp_data object using all parameter estimates, dimensionality (methods, locations, timepoints) and the total number of counts nu_ij of a birp object
`summary.birp`	Summary method for birp objects
`summary.birp_data`	Summarize a birp_data object

Maintainer

Madleina Caduff <madleina.caduff@unifr.ch>

Author(s)

Madleina Caduff [aut, cre], Daniel Wegmann [aut], Liam Singer [aut], Raphael Eckel [ctb], Andreas Füglistaler [ctb]

`assess_NB` *Assess whether a Poisson model can replace the Negative Binomial model*

Description

This function tests if the Poisson model is appropriate by simulating replicate datasets under the Poisson assumption and comparing the overdispersion parameter estimates with those obtained from the Negative Binomial (NB) model fit to the original data.

Usage

```
assess_NB(
  x,
  stochastic = FALSE,
  numRep = 100,
  cutoff = 0.05,
  plot = TRUE,
  verbose = TRUE
)
```

Arguments

`x`	A `birp` object estimated under a negative binomial model.
`stochastic`	Logical; if `TRUE`, use a stochastic trend model, otherwise deterministic (default).
`numRep`	Integer; number of replicate datasets to simulate (default 100).
`cutoff`	Numeric; significance threshold for the fraction of replicates where NB overdispersion exceeds Poisson estimate (default 0.05).
`plot`	Logical; if `TRUE`, plot the distributions of overdispersion parameters from simulated Poisson replicates (default `TRUE`).
`verbose`	Logical; if `FALSE`, suppress console output (default `TRUE`).

Value

A list containing:

`keep_NB`	Logical scalar, `TRUE` if NB model should be kept (data shows overdispersion). If `FALSE`, birp should be re-run using the Poisson model to gain power.
`keep_NB_per_method`	Logical vector indicating whether NB should be kept for each method.
`frac`	Numeric vector with fractions of replicates where Poisson simulated overdispersion exceeded observed NB overdispersion.
`b_Pois`	Matrix of overdispersion parameter estimates from Poisson-simulated replicates.
`b_x`	Numeric vector of overdispersion parameter estimates from the original NB fit.

Examples

```
data <- simulate_birp()
est <- birp(data, negativeBinomial = TRUE)
res_assess <- assess_NB(est, numRep = 5)
```

`birp` *Create a* `birp` *Object*

Description

This function runs the Markov Chain Monte Carlo (MCMC) algorithm on a `birp_data` object to estimate model parameters and returns a fitted `birp` object.

Usage

```
birp(
  data,
  timesOfChange = c(),
  negativeBinomial = FALSE,
  stochastic = FALSE,
  BACI = NULL,
  assumeTrueDetectionProbability = FALSE,
  iterations = 1e+05,
  numBurnin = 10,
  burnin = 1000,
  thinning = 10,
  verbose = TRUE
)
```

Arguments

`data`	A birp_data object containing the input data.
`timesOfChange`	Numeric or integer vector specifying the times of change (change points) for the model.
`negativeBinomial`	Logical; if `TRUE`, fits a negative binomial model instead of the default Poisson model.
`stochastic`	Logical; if `TRUE`, fits a stochastic trend model instead of the default deterministic trend model.
`BACI`	Optional matrix specifying the BACI (Before-After-Control-Impact) design. Each row corresponds to a control/intervention group and each column to an epoch. The first column contains the control-intervention group names (matching those in `data`), and subsequent columns specify which gamma (rate of change) parameter to use for each group and epoch. For example, `BACI = matrix(c("A", "B", 1, 1, 1, 2), nrow = 2)` corresponds to a canonical BACI design with control group "A" and intervention group "B".
`assumeTrueDetectionProbability`	Logical; if `TRUE`, provided detection probabilities are treated as true probabilities (logit-transformed without standardization).
`iterations`	Integer; total number of MCMC iterations to run.

`numBurnin`	Integer; number of burn-in cycles to run.
`burnin`	Integer; number of MCMC iterations per burn-in cycle.
`thinning`	Integer; thinning interval for saving MCMC samples. Only every `thinning`th iteration is retained.
`verbose`	Logical; if `FALSE`, suppresses console output.

Value

An object of type `birp` containing MCMC results and model estimates.

Examples

```
data <- simulate_birp()
est <- birp(data)
```

`birp_data` *Create a birp_data Object from Count and Effort Matrices*

Description

Constructs a ’birp_data’ object from matrices of observed counts and corresponding efforts for a single method.

Usage

```
birp_data(counts, efforts, times, CI_groups = NULL, location_names = NULL)
```

Arguments

`counts`	A matrix of observed counts (J locations × K timepoints). Each row corresponds to a location and each column to a timepoint.
`efforts`	A matrix of observation effort with the same dimensions as ‘counts‘.
`times`	A vector of length K specifying the timepoints.
`CI_groups`	A vector of length J specifying the control-intervention (CI) group for each location. Defaults to a single group (‘group_1‘) if not provided.
`location_names`	Optional names for the locations. Defaults to ‘"Location_1"‘, ‘"Location_2"‘, etc.

Value

An object of type birp_data

Examples

```
data <- birp_data(c(10,20,30), c(100,200,300), c(1,2,5))
```

`birp_data_from_data_frame` *Create a 'birp_data' Object from a Data Frame or List of Data Frames*

Description

Constructs a 'birp_data' object from a single data frame or a list of data frames, with one per method.

Usage

```
birp_data_from_data_frame(data)
```

Arguments

`data`	A data frame or a list of data frames, each representing one method. Each data frame must include the following columns: 'timepoint', 'location', 'counts', 'effort', and 'CI_group'. Each row represents a survey conducted at a specific timepoint and location, and for a specific control-intervention (CI) group.

Value

An object of type birp_data

Examples

```
df <- data.frame(
  timepoint = 1:10,
  location = rep(1, 10),
  counts = runif(10, 0, 100),
  effort = rexp(10),
  CI_group = "intervention"
)
data <- birp_data_from_data_frame(df)
```

`birp_data_from_file` *Create a birp_data Object from File(s)*

Description

Constructs a 'birp_data' object from one or more input files, each representing data for a different method.

Usage

```
birp_data_from_file(filenames, method_names = NA, sep = ",")
```

Arguments

`filenames`	A character vector of file paths. Each file must contain a data frame with the columns 'timepoint', 'location', 'counts', 'effort' and 'CI_group'.
`method_names`	Optional vector of method names corresponding to the input files. If not provided, names are inferred from the file names.
`sep`	The field separator used in the files (default is comma).

Value

An object of type birp_data

Examples

```
dir <- system.file("extdata", package = "birp")
filenames <- file.path(dir, "birp_Method_1_simulated_counts.txt")
data <- birp_data_from_file(filenames = filenames, sep = "\t")
```

`birp_from_command_line` *Create a birp Object from Command-Line Output Files*

Description

This function creates a birp object by reading the output files generated by the command-line version of the birp tool.

Usage

```
birp_from_command_line(path)
```

Arguments

`path`	Character string specifying the directory path containing all birp output files.

Value

An object of type `birp` containing MCMC results and model estimates read from files.

Examples

```
est <- birp_from_command_line(file.path(system.file("extdata", package = "birp")))
```

`plot.birp` *Plot posterior distributions of gamma parameters*

Description

Plots the posterior densities of the gamma parameters estimated by a `birp` object.

Usage

```
## S3 method for class 'birp'
plot(
  x,
  shadingIncrease = NA,
  shadingDecrease = "#f2c7c7",
  col = "black",
  lwd = 1,
  lty = 1:x$num_gamma,
  xlim = NA,
  ylim = NA,
  add = FALSE,
  xlab = expression(gamma),
  ylab = "Posterior density",
  legend = x$gamma_names,
  lineAtZero = TRUE,
  ...
)
```

Arguments

`x`	A `birp` object.
`shadingIncrease`	Character or color specification; Shading color for the range where the gamma parameter is greater than 0 (`gamma > 0`). If `NA`, shading is omitted. Default is `NA`.
`shadingDecrease`	Character or color specification; Shading color for the range where the gamma parameter is less than 0 (`gamma < 0`). If `NA`, shading is omitted. Default is `"#f2c7c7"`.
`col`	Character vector or color values; Line color(s) for the density plots. If a single value is provided, it is recycled for all gamma parameters. Default is `"black"`.
`lwd`	Numeric vector; Line width(s) for the density plots. If a single value is provided, it is recycled. Default is 1.
`lty`	Numeric or character vector; Line type(s) for the density plots. If a single value is provided, it is recycled. Default is `1:x$num_gamma`.
`xlim`	Numeric vector of length 2; Optional x-axis limits. If `NA`, limits are determined automatically from the density data. Default is `NA`.

`ylim`	Numeric vector of length 2; Optional y-axis limits. If `NA`, limits are determined automatically from the density data. Default is `NA`.
`add`	Logical; If `TRUE`, adds the densities to an existing plot. Otherwise, creates a new plot. Default is `FALSE`.
`xlab`	Character; Label for the x-axis. Default is `expression(gamma)`.
`ylab`	Character; Label for the y-axis. Default is `"Posterior density"`.
`legend`	Character vector of legend labels, or `NA` to suppress the legend. Default is `x$gamma_names`.
`lineAtZero`	Logical; If `TRUE`, adds a vertical line at x = 0 to indicate no effect. Default is `TRUE`.
`...`	Additional graphical parameters passed to `lines` (when plotting the densities) and `plot` (when creating a new plot).

Value

No return value, called for side effects.

See Also

`birp`

Examples

```
data <- simulate_birp()
est <- birp(data)
plot(est)
```

`plot.birp_data` *Plot a birp_data Object*

Description

This function plots observed counts per unit of effort over time, for each method-location combination in a `birp_data` object.

Usage

```
## S3 method for class 'birp_data'
plot(
  x,
  col = 1:length(x$locations),
  lwd = 1,
  lty = 1:length(x$method_names),
  pch = 1:length(x$CI_groups),
  xlab = "time",
  ylab = "counts per unit of effort",
```

```
  legend.x = "topright",
  legend.y = NULL,
  legend.bty = "o",
  xlim = range(as.numeric(x$times)),
  ylim = NA,
  ...
)
```

Arguments

`x`	A `birp_data` object to be plotted.
`col`	A vector of colors, recycled to match the number of locations.
`lwd`	A vector of line widths, recycled to match the number of method-location combinations.
`lty`	A vector of line types, recycled to match the number of methods.
`pch`	A vector of plotting characters, recycled to match the number of control-intervention (CI) groups.
`xlab`	Label for the x-axis.
`ylab`	Label for the y-axis.
`legend.x`	The x-position for the legend. Use `NA` to omit the legend.
`legend.y`	The y-position for the legend.
`legend.bty`	Box type for the legend; either `"o"` (default) or `"n"`.
`xlim`	Numeric vector specifying the x-axis limits.
`ylim`	Numeric vector specifying the y-axis limits. If `NA`, limits are computed automatically.
`...`	Additional graphical parameters passed to `plot()` or `lines()`.

Value

No return value. Called for side effects.

Examples

```
data <- simulate_birp()
plot(data)
```

`plot_epoch_pair` *Plot joint posterior of two gamma parameters*

Description

Plots a 2D density contour for the joint posterior of two gamma parameters from a `birp` object.

Usage

```
plot_epoch_pair(
  x,
  gamma1 = 1,
  gamma2 = 2,
  xlab = .getLabelGamma.birp(x, gamma1),
  ylab = .getLabelGamma.birp(x, gamma2),
  xlim = range(x$trace_gamma[, c(gamma1, gamma2)]),
  ylim = xlim,
  col = "deeppink",
  diag.col = "black",
  diag.lwd = 1,
  diag.lty = 1,
  zero.col = "black",
  zero.lwd = 1,
  zero.lty = 2,
  print.p = TRUE,
  add = FALSE,
  ...
)
```

Arguments

`x`	A `birp` object.
`gamma1`	Integer; Index of the first gamma parameter to plot on the x-axis. Default is 1.
`gamma2`	Integer; Index of the second gamma parameter to plot on the y-axis. Default is 2.
`xlab`	Character; Label for the x-axis. Default is dynamically set based on `gamma1`.
`ylab`	Character; Label for the y-axis. Default is dynamically set based on `gamma2`.
`xlim`	Numeric vector of length 2; Optional x-axis limits. Default is the range of gamma1 and gamma2 values.
`ylim`	Numeric vector of length 2; Optional y-axis limits. Default is the same as `xlim`.
`col`	Character or color specification; Color for contour lines. Default is `"deeppink"`.
`diag.col`	Character or `NA`; Color of the diagonal line (y=x). Use `NA` to omit. Default is `"black"`.
`diag.lwd`	Numeric; Line width of the diagonal line. Default is 1.

`diag.lty`	Numeric or character; Line type of the diagonal line. Default is 1 (solid).
`zero.col`	Character or `NA`; Color of the zero reference lines (at `x=0` and `y=0`). Use `NA` to omit. Default is `"black"`.
`zero.lwd`	Numeric; Line width of the zero reference lines. Default is 1.
`zero.lty`	Numeric or character; Line type of the zero reference lines. Default is 2 (dashed).
`print.p`	Logical; If `TRUE`, adds an annotation showing the posterior probability `P(gamma1 < gamma2 | data)` or `P(gamma1 > gamma2 | data)`. Default is `TRUE`.
`add`	Logical; If `TRUE`, adds the contour plot to an existing plot. Default is `FALSE`.
`...`	Additional graphical parameters passed to `contour`.

Value

No return value; called for side effects (plotting).

See Also

`birp`

Examples

```
data <- simulate_birp(timesOfChange = 2)
est <- birp(data, timesOfChange = 2)
plot_epoch_pair(est)
```

`plot_mcmc` *Plot MCMC Traces and Posterior Densities*

Description

Visualizes the MCMC trace plots and posterior densities of the gamma parameters from a `birp` object.

Usage

```
plot_mcmc(x, col = c("black", "blue"))
```

Arguments

`x`	A `birp` object containing posterior samples.
`col`	Character vector; Colors for trace and density plots. Default is c("black", "blue").

Value

No return value; the function is called for its side effects (plotting).

See Also

`birp`

Examples

```
data <- simulate_birp()
est <- birp(data)
plot_mcmc(est)
```

`plot_trend` *Plot Posterior Trend Estimates*

Description

Visualizes posterior trends from a `birp` object by plotting the median and quantile intervals of the estimated relative densities over time. Optionally, vertical lines can be added to mark epoch boundaries and survey timepoints.

Usage

```
plot_trend(
  x,
  CI_group = 1,
  n_points = 1000,
  quantiles = c(0.99, 0.9, 0.5, 0.25),
 quantile.col = gray(seq(1, 0, length.out = length(quantiles) + 2)[2:(length(quantiles)
    + 1)]),
  quantile.border = NA,
  median.col = "deeppink",
  median.lwd = 1,
  median.lty = 1,
  epoch.col = "black",
  epoch.lwd = 1,
  epoch.lty = 1,
  times.col = "black",
  times.lwd = 1,
  times.lty = 2,
  log = FALSE,
  xlab = "Time",
  ylab = paste(c("log", "Relative Density")[c(log, TRUE)], collapse = " "),
  main = x$CI_groups[CI_group],
  ...
)
```

Arguments

`x`	A `birp` object containing MCMC trace and model outputs.
`CI_group`	Integer; Index of the control-intervention group to plot. Default is 1.
`n_points`	Integer; Number of points to evaluate the trend over time. Default is 1000.
`quantiles`	Numeric vector; Quantiles to plot as shaded intervals. Must be in (0, 1). Default is c(0.99, 0.9, 0.5, 0.25).
`quantile.col`	Character vector or color values; Fill colors for quantile polygons. Default is shades of gray.
`quantile.border`	Character or NA; Border color for quantile polygons. Use NA to omit borders. Default is NA.
`median.col`	Character; Color of the median trend line. Default is "deeppink".
`median.lwd`	Numeric; Line width for the median trend. Default is 1.
`median.lty`	Numeric or character; Line type for the median trend line. Default is 1 (solid).
`epoch.col`	Character or color specification; Color for lines representing epoch boundaries. Default is `"black"`.
`epoch.lwd`	Numeric; Line width for epoch boundary lines. Default is 1.
`epoch.lty`	Numeric or character; Line type for epoch boundary lines. Default is 1 (solid).
`times.col`	Character or color specification; Color for vertical lines representing measurement times. Default is `"black"`.
`times.lwd`	Numeric; Line width for measurement time lines. Default is 1.
`times.lty`	Numeric or character scalar; Line type for measurement time lines. Default is 2 (dashed).
`log`	Logical; If `TRUE`, plot relative densities on a logarithmic scale; otherwise plot on the original scale. Default is `FALSE`.
`xlab`	Character; Label for the x-axis. Default is `"Time"`.
`ylab`	Character; Label for the y-axis. Default dynamically set to either `"log Relative Density"` or `"Relative Density"`.
`main`	Character; Main title of the plot. Defaults to the name of the selected CI group.
`...`	Additional graphical parameters passed to the base `plot` function.

Value

No return value, called for side effects.

See Also

`birp`

Examples

```
data <- simulate_birp()
est <- birp(data)
plot_trend(est)
```

`print.birp` *Print a birp object*

Description

Prints a summary of the estimated parameters from a `birp` model.

Usage

```
## S3 method for class 'birp'
print(x, ...)
```

Arguments

`x`	A `birp` object.
`...`	Additional arguments passed to internal methods (currently unused).

Value

Invisibly returns the input `x`, called for side effects.

See Also

`birp`

Examples

```
data <- simulate_birp()
est <- birp(data)
print(est)
```

`print.birp_data` *Print a birp_data object Prints a summary of a* `birp_data` *object, including the number of methods, locations, control-intervention (CI) groups, and timepoints, as well as the names or identifiers for each.*

Description

Print a birp_data object Prints a summary of a `birp_data` object, including the number of methods, locations, control-intervention (CI) groups, and timepoints, as well as the names or identifiers for each.

Usage

```
## S3 method for class 'birp_data'
print(x, ...)
```

Arguments

`x`	A `birp_data` object to be printed.
`...`	Additional arguments passed to function.

Value

No return value; this function is called for its side effects (printing to console).

Examples

```
data <- simulate_birp()
print(data)
```

`simulate_birp` *Simulate Data for BIRP Models Generates simulated count data using the BIRP model framework with user-defined parameters.*

Description

Simulate Data for BIRP Models Generates simulated count data using the BIRP model framework with user-defined parameters.

Usage

```
simulate_birp(
  timepoints = c(1, 2, 3),
  timesOfChange = c(),
  gamma = NULL,
  negativeBinomial = FALSE,
  stochastic = FALSE,
  numLocations = 2,
  numMethods = 1,
  numCIGroups = 1,
  numCovariatesEffort = 1,
  numCovariatesDetection = 0,
  BACI = NULL,
  n_bar = 1000,
  N_0 = NULL,
  a = NULL,
  logSigma = NULL,
  logPhi = NULL,
  covariatesEffort = "gamma(1, 2)",
  covariatesDetection = "normal(0, 1)",
  proportionZeroEffort = 0,
  verbose = TRUE
)
```

Arguments

`timepoints`	Integer vector specifying time points.
`timesOfChange`	Integer vector indicating time points at which change in growth rate (gamma) occurs.
`gamma`	Numeric vector denoting the values of gamma to simulate. If NULL, all gamma will be set to zero
`negativeBinomial`	Logical; if `TRUE`, use negative binomial instead of Poisson.
`stochastic`	Logical; if `TRUE`, simulate abundance as a stochastic process instead of deterministic.
`numLocations`	Integer; number of spatial locations.
`numMethods`	Integer; number of sampling methods.
`numCIGroups`	Integer; number of control–intervention groups.
`numCovariatesEffort`	Integer; number of effort covariates.
`numCovariatesDetection`	Integer; number of detection covariates.
`BACI`	Optional matrix specifying BACI design (see Details).
`n_bar`	Expected average total observations per time point (across all locations).
`N_0`	Optional numeric; initial abundance. If NULL, n_bar will be used instead
`a`	A numeric value or vector; detection parameter(s) for the negative binomial distribution. Can be a single value (shared across methods) or a vector of values (one per method).
`logSigma`	Optional numeric; log standard deviation of abundance process in the stochastic model. If NULL, logSigma will be set to -1
`logPhi`	Optional numeric; log standard deviation of detection process in the stochastic model. If NULL, logPhi will be simulated according to the model assumptions
`covariatesEffort`	Specifies how effort is calculated for covariates. Accepts: (1) a single number used for all covariates and locations; (2) a numeric vector with one value per covariate (applied to all locations); (3) a distribution string to simulate effort from, e.g., "gamma(a, b)" or "uniform(a, b)"; or (4) a vector of such distribution strings, one per covariate.
`covariatesDetection`	Specifies how detection probabilities are calculated for covariates. Accepts: (1) a single number for all covariates and locations; (2) a numeric vector with one value per covariate (applied to all locations); (3) a distribution string, e.g., "normal(a, b)" or "uniform(a, b)"; or (4) a vector of such distribution strings, one per covariate.
`proportionZeroEffort`	Proportion of time–location–method combinations with zero effort (0 to 1).
`verbose`	Logical; if `TRUE`, print progress messages.

Details

The ‘BACI‘ matrix defines a Before-After Control-Impact experimental design. It must be a binary matrix with two columns and one row per observation. - The first column indicates the time period (‘0 = before‘, ‘1 = after‘). - The second column indicates the treatment type (‘0 = control‘, ‘1 = impact‘). This allows modeling interactions between time and treatment to isolate impact effects.

Value

An object of type birp_data containing the simulated dataset.

Examples

```
data <- simulate_birp()
```

`simulate_birp_from_results` *This function simulates a birp_data object using all parameter estimates, dimensionality (methods, locations, timepoints) and the total number of counts nu_ij of a birp object*

Description

This function simulates a birp_data object using all parameter estimates, dimensionality (methods, locations, timepoints) and the total number of counts nu_ij of a birp object

Usage

```
simulate_birp_from_results(
  x,
  negativeBinomial = FALSE,
  stochastic = FALSE,
  mu = NULL,
  b = NULL,
  logSigma = NULL,
  logPhi = NULL,
  verbose = TRUE
)
```

Arguments

`x`	An object of type `birp`.
`negativeBinomial`	Logical; if `TRUE`, simulate counts using a negative binomial distribution instead of Poisson.
`stochastic`	Logical; if `TRUE`, use a stochastic model with log-normal fluctuations.

`mu`	A numeric vector specifying values of μ for the negative binomial model, with one value per method-location combination. If `NULL`, μ_i for method i is set to $1/$number of locations.
`b`	A numeric vector specifying values of b for the negative binomial model (one per method). If `NULL`, all b_i are set to 1.
`logSigma`	A single numeric value specifying `logSigma` for the stochastic model. If `NULL`, `logSigma` is set to `-1`.
`logPhi`	A numeric vector specifying values of `logPhi` for the stochastic model. If `NULL`, values are simulated according to the model assumptions.
`verbose`	Logical; if `FALSE`, suppresses console output.

Value

An object of type birp_data

Examples

```
data  <- simulate_birp()
x <- birp(data)
data2 <- simulate_birp_from_results(x)
```

`summary.birp` *Summary method for birp objects*

Description

Provides a printed summary of model estimates for a `birp` object.

Usage

```
## S3 method for class 'birp'
summary(object, ...)
```

Arguments

`object`	A `birp` object.
`...`	Additional arguments passed to `print.birp`.

Value

Invisibly returns the input `object`, called for side effects.

See Also

`birp`

Examples

```
data <- simulate_birp()
est <- birp(data)
summary(est)
```

`summary.birp_data` *Summarize a birp_data object*

Description

Provides a printed summary of the contents of a `birp_data` object.

Usage

```
## S3 method for class 'birp_data'
summary(object, ...)
```

Arguments

`object`	A `birp_data` object to be summarized.
`...`	Additional arguments.

Value

No return value; this function is called for its side effects (printing to console).

Examples

```
data <- simulate_birp()
summary(data)
```

Index