Package 'birp'

May 16, 2025

Type Package

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

Version 0.0.3

Date 2025-05-14

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

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 2025-05-16 09:50:13 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 it is possible to use the
                        Poisson model instead of the Negative Binomial
                        (NB) model
birp                    Creating a Birp Object
birp-package            Testing for Population Trends Using Low-Cost
                        Ecological Count Data
birp_data               Creating a Birp Data Object based on counts and
```

	efforts for a single method
`birp_data_from_data_frame`	Creating a Birp Data Object based on dataframe(s)
`birp_data_from_file`	Creating a Birp Data Object based on filenames
`birp_from_command_line`	Creating a Birp Object based on output files of command-line tool
`plot.birp`	Plotting a birp object
`plot.birp_data`	Plotting a birp_data Object
`plot_epoch_pair`	Plotting posterior estimates of gamma pairs
`plot_mcmc`	Plotting the MCMC chains
`plot_trend`	Plotting posterior trends
`print.birp`	Printing a birp object
`print.birp_data`	Printing a birp_data Object
`simulate_birp`	This function simulates a birp_data object for tidy data
`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`	Summarizing a birp object
`summary.birp_data`	This function summarizes 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 it is possible to use the Poisson model instead of the Negative Binomial (NB) model*

Description

Assess whether it is possible to use the Poisson model instead of the Negative Binomial (NB) model

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`	A boolean indicating if deterministic (default) or stochastic trend model should be used
`numRep`	The number of replicates to run
`cutoff`	The fraction of replicates for which b_Pois > b_x
`plot`	A boolean indicating if the distribution of b should be plotted.
`verbose`	Logical. If `FALSE`, the console output is suppressed

Value

A list. If keepNB is TRUE, the data is overdispersed and the negative binomial model should be used to account for the overdispersion. If keepNB is FALSE, birp should be re-run using the Poisson model to gain power.

Examples

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

`birp` *Creating a Birp Object*

Description

This function creates a birp object by running the MCMC

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
`timesOfChange`	A numeric or integer vector specifying the times of change
`negativeBinomial`	A boolean indicating if Poisson (default) or negative binomial model should be used
`stochastic`	A boolean indicating if deterministic (default) or stochastic trend model should be used
`BACI`	A matrix specifying the BACI configuration. Each row of the matrix corresponds to a control/intervention group, and each column to an epoch. The very first column specifies the name of the control-intervention group and must match the groups specified in data. The values of the matrix specify which gamma to use for each group and epoch. E.g. BACI = matrix(c("A", "B", 1, 1, 1, 2), nrow = 2) corresponds to a canonical BACI design where the first row represents the control group (A) and the second row represents the intervention group (B)
`assumeTrueDetectionProbability`	A boolean indicating if provided detection probabilities are "true", i.e. meaning that they will be transform to logit and not standardized
`iterations`	The number of MCMC iterations to run
`numBurnin`	The number of burnin cycles to run
`burnin`	The number of MCMC iterations per burnin cycle
`thinning`	Integer value specifying the thinning interval for recording the MCMC trace. Only every `thinning`th iteration will be retained (e.g., `thinning` = 1 records every iteration, `thinning` = 2 records every second iteration, and so on).
`verbose`	Logical. If `FALSE`, the console output is suppressed

Value

An object of class birp

Examples

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

`birp_data` *Creating a Birp Data Object based on counts and efforts for a single method*

Description

This function creates a birp_data object

Usage

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

Arguments

`counts`	An J x K matrix of the observed counts. Each of the J rows corresponds to a location obtained at each of K times (columns)
`efforts`	An J x K matrix of the effort conducted to observe the counts
`times`	A vector giving the K time points at which counts were obtained
`CI_groups`	The name of the control-intervention (CI) group for each location. By default, all locations belong to the same group (group_1)
`location_names`	Names to distinguish the locations. By default, locations are named after their row index in counts

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` *Creating a Birp Data Object based on dataframe(s)*

Description

This function creates a birp_data object

Usage

```
birp_data_from_data_frame(data)
```

Arguments

`data`	A single dataframe or a list of data frames (one per method). Each dataframe should consist of five columns: timepoint, location, counts, effort and CI_group. The rows of the dataframe correspond to the counts and efforts obtained at one particular timepoint, location and for one particular 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` *Creating a Birp Data Object based on filenames*

Description

This function creates a birp_data object

Usage

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

Arguments

`filenames`	A vector of filenames specifying the input file(s) (one per method)
`method_names`	Names to distinguish the methods. If NA, method names will be derived from filenames
`sep`	The field separator character

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` *Creating a Birp Object based on output files of command-line tool*

Description

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

Usage

```
birp_from_command_line(path)
```

Arguments

`path`	The path where all the output files of birp are located

Value

An object of class birp

Examples

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

`plot.birp` *Plotting a birp object*

Description

Plotting 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`	Shading color for the range gamma > 0. If `NA`, shading is omitted
`shadingDecrease`	Shading color for the range gamma < 0. If `NA`, shading is omitted
`col`	Line color, one per gamma. If a single value is provided, it is recycled to match the number of gammas.
`lwd`	Line width, one per gamma. If a single value is provided, it is recycled to match the number of gammas.
`lty`	Line type, one per gamma. If a single value is provided, it is recycled to match the number of gammas.
`xlim`	The x-limits (x1, x2) of the plot. If NA, these are determined automatically
`ylim`	The y-limits (y1, y2) of the plot. If NA, these are determined automatically
`add`	If `TRUE`, posterior density is added to currently open plot. If FALSE, a new plot is opened.
`xlab`	Name of x axis
`ylab`	Name of y axis
`legend`	Add a legend to the plot. Use NA to suppress
`lineAtZero`	If `TRUE`, adds a dashed line indicating 0.
`...`	additional parameters passed to the function.

Value

No return value, called for side effects.

See Also

`birp`

Examples

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

`plot.birp_data` *Plotting a birp_data Object*

Description

This function plots the counts per unit of effort per time-point, method and location

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`	The birp data object to be printed.
`col`	A vector of colors, recycled to match the number of methods and locations
`lwd`	A vector of line width, recycled to match the number of methods and locations
`lty`	A vector of line types, recycled to match the number of methods and locations
`pch`	A vector of plotting characters, recycled to match the number of control/intervention groups
`xlab`	The label of the x-axis
`ylab`	The label of the y-axis
`legend.x`	The x coordinate to position the legend. Use legend.x=NA to omit legend
`legend.y`	The y coordinate to position the legend
`legend.bty`	The type of box to be drawn around the legend. The allowed values are "o" (the default) and "n".
`xlim`	Set the limits of the x-axis
`ylim`	Set the limits of the y-axis
`...`	Additional parameters passed to plotting functions.

Value

No return value, called for side effects

Examples

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

`plot_epoch_pair` *Plotting posterior estimates of gamma pairs*

Description

Plotting posterior estimates of gamma pairs

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`	The index of the first gamma to plot
`gamma2`	The index of the second gamma to plot
`xlab`	A label for the x axis
`ylab`	A label for the y axis

`xlim`	The x-limits (x1, x2) of the plot. Note that x1 > x2 is allowed and leads to a "reversed axis". The default value, NULL, indicates that the range of the finite values to be plotted should be used
`ylim`	The y-limits of the plot
`col`	The color for the contour lines
`diag.col`	The color of the diagonal line. Use NA to indicate that no line should be plotted
`diag.lwd`	The line width of the diagonal line
`diag.lty`	The line type of the diagonal line
`zero.col`	The color of the line at zero. Use NA to indicate that no line should be plotted
`zero.lwd`	The line width of the line at zero
`zero.lty`	The line type of the line at zero
`print.p`	If `TRUE`, add text representing the posterior probability of a trend change
`add`	Boolean indicating if a lines should be added to an existing plot
`...`	additional parameters passed to the function

Value

No return value, called for side effects.

See Also

`birp`

Examples

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

`plot_mcmc` *Plotting the MCMC chains*

Description

Plotting the MCMC chains

Usage

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

Arguments

`x`	A birp object
`col`	Color(s) used in the plot

Value

No return value, called for side effects

See Also

`birp`

Examples

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

`plot_trend` *Plotting posterior trends*

Description

Plotting posterior trends

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
`CI_group`	The index of the control-intervention (CI) group to plot. By default, the first group is plotted.
`n_points`	Number of points
`quantiles`	Which quantiles to plot
`quantile.col`	Colors of the quantiles
`quantile.border`	Define border of the quantile. NA is possible
`median.col`	Color of the median
`median.lwd`	Line width of median
`median.lty`	Line type of median
`epoch.col`	Color to represent the epochs
`epoch.lwd`	Line width to represent the epochs
`epoch.lty`	Line type to represent the epochs
`times.col`	Color to represent the times of change
`times.lwd`	Line width that represents times of change
`times.lty`	Line type that represents times of change
`log`	Plot relative densities in log
`xlab`	A label for the x axis
`ylab`	A label for the y axis
`main`	A title for the plot
`...`	additional parameters passed to the function

Value

No return value, called for side effects

See Also

`birp`

Examples

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

`print.birp` *Printing a birp object*

Description

Printing a birp object

Usage

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

Arguments

`x`	A birp object.
`...`	Additional parameters passed to print functions.

Value

No return value, called for side effects.

See Also

`birp`

Examples

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

`print.birp_data` *Printing a birp_data Object*

Description

Printing a birp_data Object

Usage

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

Arguments

`x`	The birp_data object to be printed.
`...`	Other parameters passed to function

Value

No return value, called for side effects

Examples

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

`simulate_birp` *This function simulates a birp_data object for tidy data*

Description

This function simulates a birp_data object for tidy data

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`	A vector of integers that denote the time points
`timesOfChange`	A numeric or integer vector specifying the times of change
`gamma`	A numeric vector denoting the values of gamma to simulate. If NULL, all gamma will be set to zero
`negativeBinomial`	A boolean indicating if the Poisson (default) or negative binomial model should be used
`stochastic`	A boolean indicating if the deterministic (default) or stochastic trend model should be used
`numLocations`	An integer denoting the number of locations
`numMethods`	An integer denoting the number of methods
`numCIGroups`	An integer denoting the number of control-intervention (CI) groups
`numCovariatesEffort`	An integer denoting the number of covariates for modeling the effort
`numCovariatesDetection`	An integer denoting the number of covariates for modeling the detection probabilities
`BACI`	A matrix specifying the BACI configuration. Each row of the matrix corresponds to a control/intervention group, and each column to an epoch. In addition, the very first column specifies the name of the control-intervention group. The values of the matrix specify which gamma to use for each group and epoch. E.g. BACI = matrix(c("A", "B", 1, 1, 1, 2), nrow = 2) corresponds to a canonical BACI design where the first row represents the control group (A) and the second row represents the intervention group (B)
`n_bar`	A numeric value denoting the average number of counts to be simulated
`N_0`	A numeric value denoting the expected number of observations at the first time point. If NULL, n_bar will be used instead
`a`	A single value (shared across methods) or a numeric vector (per method) used to simulate values under the negative binomial distribution
`logSigma`	A single value denoting the value of logSigma of the stochastic model to simulate. If NULL, logSigma will be set to -1
`logPhi`	A numeric vector denoting the values of logPhi of the stochastic model to simulate. If NULL, logPhi will be simulated according to the model assumptions
`covariatesEffort`	Denotes the covariates for calculating the effort. There are 3 options: 1) a single number, which is used for all covariates and locations; 2) a vector of numbers, one per covariate but the same for all location; 3) a distribution to simulate the effort from, which can be either "gamma(a, b)" or "uniform(a, b)" where a and b can be set or 4) a vector of such distributions, one per covariate
`covariatesDetection`	Denotes the covariates for calculating the detection probabilities. There are 3 options: 1) a single number, which is used for all covariates and locations; 2) a

vector of numbers, one per covariate but the same for all location; 3) a distribution to simulate the detection probabilities from, which can be either "normal(a, b)" or "uniform(a, b)" where a and b can be set or 4) a vector of such distributions, one per covariate

`proportionZeroEffort`	The proportion of effort covariates which are set to zero
`verbose`	Logical. If `FALSE`, the console output is suppressed

Value

An object of type birp_data

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`	A boolean indicating if the Poisson (default) or negative binomial model should be used

`stochastic`	A boolean indicating if the deterministic (default) or stochastic trend model should be used
`mu`	A numeric vector denoting the values of mu to be used for the negative binomial model, where the size is given by the number of method-location combinations. If NULL, all mu_i for one method i are set to the (number of locations)^(-1) for that method
`b`	A numeric vector denoting the values of b to be used for the negative binomial model (one per method). If NULL, all b_i are set to 1
`logSigma`	A single value denoting the value of logSigma of the stochastic model to simulate. If NULL, logSigma will be set to -1
`logPhi`	A numeric vector denoting the values of logPhi of the stochastic model to simulate. If NULL, logPhi will be simulated according to the model assumptions
`verbose`	Logical. If `FALSE`, the console output is suppressed

Value

An object of type birp_data

Examples

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

`summary.birp` *Summarizing a birp object*

Description

Summarizing a birp object

Usage

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

Arguments

`object`	A birp object.
`...`	Additional parameters passed to summary functions.

Value

No return value, called for side effects.

See Also

birp

Examples

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

`summary.birp_data` *This function summarizes a birp_data object*

Description

This function summarizes a birp_data object

Usage

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

Arguments

`object`	The birp_data object to be printed.
`...`	Other parameters passed to function

Value

No return value, called for side effects

Examples

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

Index