

Package: ecoGLMM (via r-universe)

October 4, 2026

Type Package

Title Reproducible Ecological Generalized Linear Mixed Model Pipelines

Version 0.1.4

Description Fits and compares generalized linear mixed models for multiple ecological responses and environmental predictors. The package supports additive and temporal-interaction candidate models, AICc model selection, likelihood-ratio tests, coefficient extraction, Nakagawa R-squared, simulation-based diagnostics, figures, and spreadsheet exports. Model selection follows Burnham and Anderson (2002, ISBN:9780387953649); marginal and conditional R-squared follow Nakagawa and Schielzeth (2013) <doi:10.1111/j.2041-210x.2012.00261.x>.

License MIT + file LICENSE

URL <https://github.com/andre-fcsantos/ecoGLMM>

BugReports <https://github.com/andre-fcsantos/ecoGLMM/issues>

Encoding UTF-8

Depends R (>= 4.1.0)

Imports DHARMA, ggeffects, ggplot2, glmmTMB, MuMIn, performance, stats, writexl

Suggests knitr, readxl, remotes, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libssl-dev zlib1g-dev

Repository <https://andre-fcsantos.r-universe.dev>

Date/Publication 2026-10-04 12:03:53 UTC

RemoteUrl <https://github.com/andre-fcsantos/ecoglmm>

RemoteRef HEAD

RemoteSha 5c3330c2f1e85e1451082ebfef1dbbd8f56cf04f

Contents

ecoGLMM-package	2
adjust_beta	2
compare_models	3
diagnose_models	3
export_ecoglmm	4
extract_results	6
plot_coefficients	7
plot_effect	8
plot_selection	9
prepare_acoustic_indices	10
prepare_ecodata	11
run_ecoglmm	12
test_interactions	13
Index	15

ecoGLMM-package	<i>Reproducible ecological GLMM pipelines</i>
-----------------	---

Description

Fits, compares, diagnoses, plots, and exports candidate generalized linear mixed models for multiple ecological responses and environmental predictors.

adjust_beta	<i>Keep proportions inside the open interval (0, 1)</i>
-------------	---

Description

Keep proportions inside the open interval (0, 1)

Usage

adjust_beta(x, epsilon = 0.001)

Arguments

- | | |
|---------|-----------------------|
| x | Numeric vector. |
| epsilon | Small boundary value. |

Value

A numeric vector with boundary values adjusted.

Examples

adjust_beta(c(0, 0.25, 0.5, 0.75, 1))

compare_models	<i>Compare a candidate set using corrected Akaike information criterion</i>
----------------	---

Description

Compare a candidate set using corrected Akaike information criterion

Usage

```
compare_models(models, competitive_delta = 2)
```

Arguments

models	Named list of fitted models.
competitive_delta	Maximum delta AICc defining competitive models.

Value

Model-selection data frame.

Examples

```
dat <- data.frame(  
  y = c(1.0, 1.3, 1.2, 1.7, 1.5, 2.0, 1.8, 2.2, 2.1, 2.5, 2.4, 2.8),  
  x = rep(1:6, 2),  
  period = factor(rep(c("early", "late"), each = 6))  
)  
models <- list(  
  x_Add = stats::lm(y ~ x + period, data = dat),  
  x_Int = stats::lm(y ~ x * period, data = dat)  
)  
compare_models(models)
```

diagnose_models	<i>Run DHARMA diagnostics for selected models</i>
-----------------	---

Description

Run DHARMA diagnostics for selected models

Usage

```
diagnose_models(object, nsim = 1000, seed = NULL)
```

Arguments

object	An 'ecoglmm_fit' object.
nsim	Number of simulations.
seed	Optional random seed.

Value

List containing a summary table and simulated residual objects.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglmm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
diagnostics <- diagnose_models(fit, nsim = 20, seed = 1)
diagnostics$summary
```

export_ecoglmm

Export ecoGLMM results to an output directory

Description

Export ecoGLMM results to an output directory

Usage

```
export_ecoglmm(
  object,
  path,
  diagnostics = NULL,
```

```

    figures = TRUE,
    diagnostic_pdf = TRUE
  )

```

Arguments

object	An 'ecoglmm_fit' object.
path	Output directory.
diagnostics	Optional result returned by [diagnose_models()].
figures	Save effect and selection figures.
diagnostic_pdf	Save a multi-page PDF containing the DHARMA plot and a diagnostic summary for each selected response model. The PDF is created only when diagnostics is supplied.

Value

Invisibly returns generated file paths.

Examples

```

set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglmm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
output <- file.path(tempdir(), "ecoGLMM-example")
files <- export_ecoglmm(fit, path = output, figures = FALSE)
basename(files)

```

extract_results	<i>Extract coefficients and model-level statistics</i>
-----------------	--

Description

Extract coefficients and model-level statistics

Usage

```
extract_results(models, selections, conf_level = 0.95)
```

Arguments

models	Named list of selected models.
selections	Named list of model-selection tables.
conf_level	Confidence level for Wald intervals.

Value

Tidy coefficient data frame.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
extract_results(fit$best_models, fit$selection)
```

plot_coefficients	<i>Plot coefficients and confidence intervals</i>
-------------------	---

Description

Plot coefficients and confidence intervals

Usage

```
plot_coefficients(object, include_intercept = FALSE)
```

Arguments

object	An 'ecoglm_fit' object.
include_intercept	Include intercept terms.

Value

A faceted ggplot object.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
plot_coefficients(fit)
```

plot_effect

Plot the marginal effect of a selected model

Description

Plot the marginal effect of a selected model

Usage

```
plot_effect(object, response, show_observed = TRUE)
```

Arguments

object	An 'ecoglm_fit' object.
response	Response name.
show_observed	Add observed values.

Value

A ggplot object.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
plot_effect(fit, "index", show_observed = FALSE)
```

plot_selection	<i>Plot model-selection results</i>
----------------	-------------------------------------

Description

Plot model-selection results

Usage

```
plot_selection(object, response, metric = c("delta", "weight"))
```

Arguments

object	An 'ecoglm_fit' object.
response	Response name.
metric	Either 'delta' or 'weight'.

Value

A ggplot object.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
plot_selection(fit, "index")
```

`prepare_acoustic_indices`*Prepare common acoustic-index transformations*

Description

Prepare common acoustic-index transformations

Usage

```
prepare_acoustic_indices(  
  data,  
  ndsi = "NDSI",  
  beta_responses = c("AEI", "ACT"),  
  ndsi_output = "NDSI_beta",  
  epsilon = 0.001  
)
```

Arguments

<code>data</code>	A data frame.
<code>ndsi</code>	Name of the NDSI column, or 'NULL'.
<code>beta_responses</code>	Names of responses already bounded by zero and one.
<code>ndsi_output</code>	Name assigned to transformed NDSI.
<code>epsilon</code>	Boundary adjustment used by <code>[adjust_beta()]</code> .

Value

A modified data frame.

Examples

```
indices <- data.frame(  
  NDSI = c(-1, 0, 1),  
  AEI = c(0, 0.5, 1),  
  ACT = c(0.2, 0.6, 0.9)  
)  
prepare_acoustic_indices(indices)
```

prepare_ecodata	<i>Validate and prepare ecological modelling data</i>
-----------------	---

Description

Validate and prepare ecological modelling data

Usage

```
prepare_ecodata(
  data,
  config,
  predictors,
  period,
  group,
  period_levels = NULL,
  standardize = TRUE,
  complete_cases = TRUE
)
```

Arguments

<code>data</code>	A data frame containing responses, predictors and grouping data.
<code>config</code>	Response configuration with ‘response’ and ‘family’ columns.
<code>predictors</code>	Environmental predictor names.
<code>period</code>	Temporal-factor column.
<code>group</code>	Random-intercept grouping column.
<code>period_levels</code>	Optional ordered levels for the temporal factor.
<code>standardize</code>	Logical; standardize numeric predictors using z-scores.
<code>complete_cases</code>	Logical; use one shared complete-case dataset for all candidate models. This should normally remain ‘TRUE’ for valid AICc comparisons.

Value

Prepared data with preparation metadata stored as attributes.

Examples

```
dat <- data.frame(
  response = c(1.1, 1.4, 1.2, 1.8, 1.5, 2),
  forest = c(10, 20, 30, 40, 50, 60),
  period = rep(c("early", "late"), 3),
  site = rep(c("a", "b", "c"), each = 2)
)
config <- data.frame(response = "response", family = "gaussian")
prepared <- prepare_ecodata(
```

```

    dat, config, predictors = "forest",
    period = "period", group = "site"
)
head(prepared)

```

run_ecoglmm

Run an ecological GLMM candidate-model pipeline

Description

Run an ecological GLMM candidate-model pipeline

Usage

```

run_ecoglmm(
  data,
  config,
  predictors,
  period,
  group,
  period_levels = NULL,
  standardize = TRUE,
  complete_cases = TRUE,
  include_additive = TRUE,
  include_interactions = TRUE,
  competitive_delta = 2,
  control = NULL
)

```

Arguments

data	Prepared or raw data frame.
config	Data frame with 'response', 'family', and optional 'label'.
predictors	Environmental predictor names.
period	Temporal-factor column name.
group	Random-intercept grouping column name.
period_levels	Optional order of temporal levels.
standardize	Standardize predictors with z-scores.
complete_cases	Use one common complete-case dataset.
include_additive	Include predictor + period models.
include_interactions	Include predictor * period models.
competitive_delta	Delta AICc threshold.
control	Optional [glmmTMB::glmmTMBControl()] object.

Value

An object of class 'ecoglmm_fit'.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglmm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = FALSE
)
fit
summary(fit)
```

test_interactions	<i>Likelihood-ratio tests for additive versus interaction models</i>
-------------------	--

Description

Likelihood-ratio tests for additive versus interaction models

Usage

```
test_interactions(models, predictors)
```

Arguments

models	Named candidate-model list.
predictors	Predictor names.

Value

Data frame of likelihood-ratio tests.

Examples

```
set.seed(1)
example_data <- expand.grid(
  site = paste0("s", 1:6),
  period = c("early", "late"),
  replicate = 1:3,
  stringsAsFactors = FALSE
)
example_data$forest <- runif(nrow(example_data))
site_effect <- setNames(rnorm(6, sd = 0.3), paste0("s", 1:6))
example_data$index <- 1 + 0.8 * example_data$forest +
  0.4 * (example_data$period == "late") +
  site_effect[example_data$site] + rnorm(nrow(example_data), sd = 0.15)
config <- data.frame(response = "index", family = "gaussian")
fit <- run_ecoglm(
  data = example_data,
  config = config,
  predictors = "forest",
  period = "period",
  group = "site",
  include_interactions = TRUE
)
test_interactions(fit$models$index, predictors = "forest")
```

Index

`adjust_beta`, [2](#)
`compare_models`, [3](#)
`diagnose_models`, [3](#)
`ecoGLMM` (ecoGLMM-package), [2](#)
`ecoGLMM-package`, [2](#)
`export_ecoglmm`, [4](#)
`extract_results`, [6](#)

`plot_coefficients`, [7](#)
`plot_effect`, [8](#)
`plot_selection`, [9](#)
`prepare_acoustic_indices`, [10](#)
`prepare_ecodata`, [11](#)

`run_ecoglmm`, [12](#)
`test_interactions`, [13](#)