Package: fastrda (via r-universe)

October 20, 2026

Type Package

Title Fast Redundancy Analysis (RDA) with High-Performance 'C++' Backend

Version 0.2.0

Description Provides a high-performance implementation of redundancy analysis (RDA) in 'C++' using 'Armadillo' and 'OpenMP'. Supports standard and partial RDA, centering, scaling, overall and axis-wise permutation tests, biplot visualization, score extraction, and prediction. Designed for large ecological, genomic, and other multivariate data sets where computational speed and memory efficiency are required.

Depends R (>= 4.0.0)

Imports Rcpp (>= 1.0.0), parallel, ggplot2 (>= 3.3.0)

Suggests RhpcBLASctl, ggrepel (>= 0.9.0), RColorBrewer (>= 1.1-2), knitr, rmarkdown, vegan

LinkingTo Rcpp, RcppArmadillo (>= 0.12.0)

VignetteBuilder knitr

License GPL (>= 3)

Encoding UTF-8

SystemRequirements OpenMP

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Zeynel Cebeci [aut, cre]

Maintainer Zeynel Cebeci <cebeciz@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-08-21 14:30:02 UTC

RemoteUrl https://github.com/cran/fastrda

RemoteRef HEAD

RemoteSha 5ea508a015003dbaee40db4fb4c9897679ce487b

Contents

biplotrda 2
fastrda 5

Index **11**

`biplotrda` *Biplot and Biplot Scores for fastrda Objects*

Description

Computes vegan-compatible response and predictor biplot scores from a fitted `fastrda` object, and generates publication-ready ordination biplots using **ggplot2**.

Usage

```
biplot_scores(
  object,
  type = c("response", "predictors", "both"),
  scaling = NULL,
  X = NULL,
  ...
)

biplotrda(
  x,
  axes = 1:2,
  type = c("all", "samples", "response", "predictors",
           "samples_response", "samples_predictors"),
  scaling = 2,
  X = NULL,
  show_ids = FALSE,
  colpal = NULL,
  sample_col = NULL,
  response_col = NULL,
  pred_col = NULL,
  sample_size = 2.5,
  arrow_size = 0.8,
  label_size = 3.5,
  arrow_scale = 1.5,
  alpha = 0.8,
  max_labels = 50,
  title = "fastrda Biplot",
  subtitle = NULL,
  theme = ggplot2::theme_minimal()
)
```

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

## S3 method for class 'fastrda'
biplot(x, ...)

## S3 method for class 'fastrda'
plot(x, ...)
```

Arguments

object, x	A fitted fastrda object (or a fastrda_biplot object for print methods).
type	Character string specifying what to compute or display. For biplot_scores: "response", "predictors", or "both". For biplotrda: "all", "samples", "response", "predictors", "samples_response", or "samples_predictors".
scaling	Integer (0-3) specifying the scaling type for scores. If NULL, uses the scaling from the original analysis model.
X	Optional matrix or data frame of predictor variables if not stored in the object.
axes	Integer vector of length 2 specifying the constrained axes to plot. Default is 1:2.
show_ids	Logical; if TRUE, displays sample, response, or predictor variable labels. Default is FALSE.
colpal	Character string specifying a color palette name from **RColorBrewer**. Default is "Set1".
sample_col	Custom color for sample points (overrides palette).
response_col	Custom color for response arrows (overrides palette).
pred_col	Custom color for predictor arrows (overrides palette).
sample_size	Numeric; size of sample points. Default is 2.5.
arrow_size	Numeric; thickness of arrows. Default is 0.8.
label_size	Numeric; size of labels. Default is 3.5.
arrow_scale	Numeric; scaling multiplier factor for biplot arrows. Default is 1.5.
alpha	Numeric; transparency value for points. Default is 0.8.
max_labels	Integer; maximum number of labels to display when show_ids = TRUE. Default is 50.
title	Character string for the plot title. Default is "fastrda Biplot".
subtitle	Character string for the plot subtitle. Default is NULL.
theme	**ggplot2** theme object. Default is ggplot2::theme_minimal().
...	Additional arguments passed to lower-level plot or score functions.

Details

The function biplot_scores extracts or dynamically calculates sample-weighted canonical coefficients and response scores conforming to the standard **vegan** package definitions across various

scaling configurations (0 to 3). It seamlessly supports both matrix and list scaling structure formats in `fastrda`.

The function `biplotrda` (and its S3 generics `biplot.fastrda` and `plot.fastrda`) builds a customizable **ggplot2** graphic supporting point layers for samples, vector arrows for response and predictor variables, automatic axis proportion variance labeling, and intelligent label overlapping control via **ggrepel** (if installed).

Value

For `biplot_scores`, returns a numeric matrix (if type is response or predictors) or a list with `response` and `predictors` components assigned to class `fastrda_biplot`.

For `biplotrda`, `biplot.fastrda`, and `plot.fastrda`, returns a **ggplot2** object representing the ordination biplot.

For `print.fastrda_biplot`, invisibly returns the input object after printing summary text to the console.

See Also

`fastrda`, `ordiplot`

Examples

```
## Not run:
set.seed(42)

n <- 30

pred <- data.frame(
  Temperature = seq(10, 30, length.out = n),
  Moisture = rev(seq(20, 80, length.out = n)),
  pH = rnorm(n, 6.8, 0.4)
)

Y <- cbind(
  Var1 = 1.2 * scale(pred$Temperature) + rnorm(n, 0, 0.30),
  Var2 = 0.9 * scale(pred$Temperature) + rnorm(n, 0, 0.35),
  Var3 = 1.1 * scale(pred$Moisture) + rnorm(n, 0, 0.30),
  Var4 = -0.9 * scale(pred$Moisture) + rnorm(n, 0, 0.30),
  Var5 = 1.0 * scale(pred$pH) + rnorm(n, 0, 0.30),
  Var6 = -0.8 * scale(pred$pH) + rnorm(n, 0, 0.30)
)

fit <- fastrda(
  response = Y,
  predictors = pred,
  axes = 2,
  scaling = 2,
  keep_workspace = "none",
  verbose = FALSE
)
```

```
bp_scores <- biplot_scores(fit, type = "both")

bp_scores$predictors
bp_scores$response

biplotrda(
  fit,
  axes = 1:2,
  scaling = 2,
  type = "all",
  show_ids = TRUE,
  title = "Simulated RDA Example"
)

## End(Not run)
```

---

fastrda *Fast Redundancy Analysis and Associated Methods*

Description

A high-performance implementation of Redundancy Analysis (RDA) and partial RDA using efficient QR decomposition, complete with integrated S3 methods for prediction, score extraction, permutation tests, summary reporting, and biplot visualization.

Usage

```
fastrda(
  response,
  predictors,
  covariates = NULL,
  scale = TRUE,
  center = TRUE,
  axes = 3,
  scaling = 2,
  keep_workspace = "minimal",
  store_fitted = FALSE,
  threads = min(2, parallel::detectCores()),
  verbose = TRUE
)

## S3 method for class 'fastrda'
predict(
  object,
  newdata = NULL,
  type = c("lc", "response"),
  rank = NULL,
  scaling = NULL,
```

```
  ...
)

## S3 method for class 'fastrda'
scores(
  x,
  display = c("both", "samples", "response", "wa", "lc", "sites"),
  choices = 1:2,
  scaling = NULL,
  ...
)

anova_fastrda(
  object,
  by = c("overall", "axis"),
  permutations = 999,
  threads = min(2, parallel::detectCores())
)

## S3 method for class 'fastrda'
anova(object, ...)

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

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

## S3 method for class 'anova.fastrda'
print(x, ...)

## S3 method for class 'anova.fastrda.axis'
print(x, ...)

get_sample_scores(object, scaling = 2)

get_site_scores(object, scaling = 2)
```

Arguments

response	A numeric matrix or data frame of response variables. Rows represent samples, columns represent variables.
predictors	A numeric matrix or data frame of predictor variables. Rows must match the number of rows in response.
covariates	An optional numeric matrix or data frame of conditioning variables to be partialled out (partial RDA). Default is NULL.
scale	Logical; if TRUE, variables are scaled to unit variance before analysis. Default is TRUE.

`center` Logical; if `TRUE`, variables are centered to zero mean before analysis. Default is `TRUE`.

`axes` Integer; number of constrained axes to retain. Must be >= 1. Default is `3`.

`scaling` Integer (0-3); type of scaling for ordination scores:

0 No scaling.

1 Samples scaled by eigenvalue sqrt, response unscaled.

2 Samples unscaled, response scaled by eigenvalue sqrt (vegan default equivalent).

3 Symmetric scaling (both samples and response scaled by eigenvalue fourth root).

`keep_workspace` Logical or character specifying workspace retention:

"minimal" Store only Q and QtY (default).

"compact" Store Q, QtY and Y_res.

"full" Store all workspace elements.

"none" Store no workspace.

`store_fitted` Logical; if `TRUE`, stores fitted values in the output. Default is `FALSE`.

`threads` Integer; number of threads for parallel computation. Defaults to number of available cores.

`verbose` Logical; if `TRUE`, prints progress and summary messages. Default is `TRUE`.

`object`, `x` A fitted `fastrda` object.

`newdata` An optional matrix or data frame of new predictor variables for prediction.

`type` Character string specifying the prediction type:

"lc" Linear combination scores for new data.

"response" Reconstructed response variables.

`rank` Integer specifying the number of constrained axes to use for prediction or low-rank reconstruction.

`choices` Integer vector specifying the constrained axes to extract.

`by` Character string specifying the permutation test type:

"overall" Global permutation test for the model.

"axis" Marginal permutation test for each constrained axis.

`permutations` Integer; number of permutations to perform. Default is `999`.

`display` Character string specifying what scores to extract:

"both" Both samples and response scores.

"samples" Sample scores only.

"response" Response scores only.

"wa" Alias for sample scores (weighted averages / vegan compatibility).

"sites" Alias for sample scores.

"lc" Linear combination scores.

`...` Additional arguments passed to other methods.

Details

Redundancy Analysis (RDA) is a multivariate constrained ordination method that finds linear combinations of predictor variables that best explain variation in response variables. It extends PCA by including explanatory variables.

`fastrda` implements RDA using a reduced-rank formulation based on efficient QR decomposition. For partial RDA models, conditioning variables are handled via Z-residualization.

Key Features:

High Performance C++ core with OpenMP parallelization for large matrices.

Scaled Scores Sample and response scores are pre-computed for fast access.

Partial RDA Support for conditioning variables.

Permutation Tests Overall and axis-wise permutation tests.

vegan Compatibility S3 methods mimic `vegan` interface.

Scaling Types: The `scaling` parameter controls how sample and response scores are scaled:

scaling = 0 No scaling (raw scores).

scaling = 1 Sample scores scaled by eigenvalue sqrt, response unscaled.

scaling = 2 Sample scores unscaled, response scaled by eigenvalue sqrt (vegan default equivalent).

scaling = 3 Symmetric scaling (both scaled by eigenvalue fourth root).

Note on Adjusted R-squared: The semi-partial adjusted R^2 formulation computed in this version differs from the default implementation found in the **vegan** package.

Value

For `fastrda`, an object of class `fastrda` containing a list of model results:

sample_scores Pre-computed scaled sample scores for the fitted scaling (access via `fit$sample_scores`).

raw_sample_scores Unscaled sample scores for flexible re-scaling.

site_scores Backward compatibility alias for `sample_scores`.

raw_site_scores Backward compatibility alias for `raw_sample_scores`.

response_scores Pre-computed scaled response scores for the fitted scaling.

loadings Response canonical coefficients/loadings.

coefficients Predictor coefficients for the full-rank model.

lc_coefficients_raw Raw linear combination coefficients.

eigenvalues Eigenvalues of the constrained axes.

prop_total Proportion of total inertia explained per axis.

prop_constrained Proportion of constrained inertia explained per axis.

n_axes Number of retained constrained axes.

n_samples Number of samples (rows).

df_model Model degrees of freedom.

df_resid Residual degrees of freedom.

df_cond Conditioning (partial) degrees of freedom.

total_inertia Total inertia of the response matrix.

constrained_inertia Inertia explained by constraints.

unconstrained_inertia Residual inertia.

conditioned_inertia Inertia removed by conditioning variables (partial RDA).

R2 Coefficient of determination.

adj_R2 Adjusted R-squared.

pseudo_F Pseudo-F model statistic.

scaling_constants Pre-computed scaling multipliers for all scaling types.

workspace Workspace used for permutation tests (if retained).

call Matched function call.

pred_names Column names of the predictor matrix.

response_names Column names of the response matrix.

sample_names Row names of the response matrix.

site_names Backward compatibility alias for `sample_names`.

For S3 methods and helper functions:

predict.fastrda Returns either linear combination scores (`type = "lc"`) or reconstructed response matrix (`type = "response"`).

scores.fastrda Returns a matrix of sample scores or a list of sample/response scores.

anova_fastrda Returns an object of class `anova.fastrda` (overall test) or `anova.fastrda.axis` (axis-wise test).

get_sample_scores Returns sample scores with optional re-scaling.

get_site_scores Backward compatibility alias for `get_sample_scores`.

summary.fastrda Invisibly returns the `fastrda` object after printing high-level model summaries.

See Also

`biplotrda`, `biplot_scores`, and `rda` for the reference implementation from the **vegan** package.

Examples

```
set.seed(123)
n <- 100
p <- 200
k <- 3
Y <- matrix(rnorm(n * p), n, p)
X <- matrix(rnorm(n * k), n, k)
Z <- matrix(rnorm(n * 2), n, 2)
colnames(Y) <- paste0("Var", 1:p)
colnames(X) <- paste0("Pred", 1:k)

# Fit standard RDA
fit <- fastrda(response = Y, predictors = X, axes = 3, verbose = FALSE)
```

```
# Print model summary
summary(fit)

# Access pre-computed scores
head(fit$sample_scores)
head(fit$response_scores)

# Fit partial RDA with covariates
fit_prda <- fastrda(response = Y, predictors = X, covariates = Z, axes = 3, verbose = FALSE)

# Extract scores using S3 method
sc <- scores(fit, display = "both", choices = 1:2)

# Predict low-rank linear combination scores (rank = 1 path)
pred_lc <- predict(fit, type = "lc", rank = 1)

# Predict response matrix for new data
pred_resp <- predict(fit, newdata = X, type = "response")

# Get sample scores with different scaling (e.g., scaling = 1)
sample_sc <- get_sample_scores(fit, scaling = 1)
head(sample_sc)


# Permutation tests (isolated to respect CRAN execution time limits)
anv_overall <- anova(fit, permutations = 99)
print(anv_overall)

anv_axis <- anova_fastrda(fit, by = "axis", permutations = 99)
print(anv_axis)
```

Index

∗ **hplot**
    biplotrda, 2
∗ **multivariate**
    biplotrda, 2
    fastrda, 5
∗ **ordination**
    biplotrda, 2
    fastrda, 5
∗ **regression**
    fastrda, 5

anova.fastrda *(*fastrda*)*, 5
anova_fastrda *(*fastrda*)*, 5

biplot.fastrda *(*biplotrda*)*, 2
biplot_scores, *9*
biplot_scores *(*biplotrda*)*, 2
biplotrda, 2, *9*

fastrda, *4*, 5

get_sample_scores *(*fastrda*)*, 5
get_site_scores *(*fastrda*)*, 5

ordiplot, *4*

plot.fastrda *(*biplotrda*)*, 2
predict.fastrda *(*fastrda*)*, 5
print.anova.fastrda *(*fastrda*)*, 5
print.fastrda *(*fastrda*)*, 5
print.fastrda_biplot *(*biplotrda*)*, 2

rda, *9*

scores *(*fastrda*)*, 5
summary.fastrda *(*fastrda*)*, 5