Package ‘coda.plot’

March 6, 2026

Type Package

Title Plots for Compositional Data

Version 0.2.1

Date 2026-03-06

Description Provides a collection of easy-to-use functions for creating visualizations of compositional data using 'ggplot2'. Includes support for common plotting techniques in compositional data analysis.

Depends R (>= 3.5), coda.base (>= 1.0.0)

License GPL

Encoding UTF-8

Imports ggplot2 (>= 3.5.0), ggrepel

RoxygenNote 7.3.2

NeedsCompilation no

Author Marc Comas-Cufí [aut, cre] (ORCID: <https://orcid.org/0000-0001-9759-0622>)

Maintainer Marc Comas-Cufí <mcomas@imae.udg.edu>

Repository CRAN

Date/Publication 2026-03-06 08:20:08 UTC

Contents

balance_dendrogram 2
clr_biplot 2
coda.plot 4
geometric_mean_barplot 4
ternary_diagram 5

Index **8**

balance_dendrogram *Compositional Balance Dendrogram*

Description

Plots a balance dendrogram based on a compositional data set and a corresponding balance matrix. This visualization helps interpret the structure of balances in compositional data analysis.

Usage

```
balance_dendrogram(X, B, group = NULL)
```

Arguments

X	A numeric matrix or data frame representing the compositional data. Rows are observations and columns are components (must be strictly positive).
B	A numeric matrix representing the balance basis (e.g., an isometric log-ratio (ilr) balance matrix).
group	Optional. If provided, show grouped box summaries under each node.

Value

A `ggplot2` object representing the balance dendrogram.

Examples

```
# Simulated compositional data and balances
X = matrix(runif(50, 1, 10), ncol = 5)
colnames(X) = LETTERS[1:5]
B = coda.base::pb_basis(X, method = 'exact')
balance_dendrogram(X, B)
```

clr_biplot *Compositional CLR Biplot*

Description

Generates a centered log-ratio (CLR) biplot for compositional data.

Usage

```
clr_biplot(
  X,
  group = NULL,
  biplot_type = "covariance",
  alpha = NULL,
  shape_group = NULL,
  return_data = FALSE,
  repel = TRUE,
  repel_force = 1,
  repel_max_overlaps = Inf
)
```

Arguments

`X`	A matrix or data frame containing compositional data (strictly positive).
`group`	Optional factor/character used to color the observations.
`biplot_type`	Character string specifying the type of biplot. Either `"covariance"` (default) or `"form"`.
`alpha`	Optional numeric in [0,1]. If provided, overrides `biplot_type`. • `alpha = 0`: covariance biplot • `alpha = 1`: form biplot
`shape_group`	Optional factor/character used to assign shapes to observations.
`return_data`	Logical. If TRUE, returns a list with data frames for observations, variables, and the ggplot object.
`repel`	Logical. If TRUE (default), use ggrepel for variable labels when available.
`repel_force`	Numeric. Repulsion force passed to `ggrepel::geom_text_repel()`.
`repel_max_overlaps`	Numeric. Maximum overlaps allowed (ggrepel).

Value

A `ggplot2` object. If `return_data = TRUE`, a list with elements `obs`, `vars`, and `plot`.

Examples

```
# Basic example (no groups)
set.seed(1)
X <- matrix(runif(120, 0.1, 10), ncol = 6)
colnames(X) <- paste0("p", 1:6)
clr_biplot(X)

# Grouped example (color)
grp <- factor(sample(c("A", "B"), nrow(X), replace = TRUE))
clr_biplot(X, group = grp)

# Color + shape
```

```
shp <- factor(sample(c("S1", "S2", "S3"), nrow(X), replace = TRUE))
clr_biplot(X, group = grp, shape_group = shp)

# Form biplot (alpha = 1) with repelled variable labels (requires ggrepel)
clr_biplot(X, group = grp, biplot_type = "form", repel = TRUE)

# Covariance biplot (alpha = 0) and custom repel settings
clr_biplot(X, group = grp, alpha = 0, repel = TRUE, repel_force = 1.5, repel_max_overlaps = 30)
```

`coda.plot` *coda.plot*

Description

Easy to run function to built visualisations by means of 'ggplot2' objects.

Author(s)

Marc Comas-Cufí

`geometric_mean_barplot` *Geometric Mean Barplot for Compositional Data*

Description

Generates a barplot based on the geometric mean of compositional parts. Optionally, it can compare groups, display the parts on the x-axis, overlay boxplots, or use centered log-ratio (clr) transformation.

Usage

```
geometric_mean_barplot(
  X,
  group,
  x_show_parts = TRUE,
  include_boxplot = FALSE,
  clr_scale = FALSE
)
```

Arguments

`X`	A numeric matrix or data frame representing compositional data. Each row is an observation and each column is a part (must be strictly positive).
`group`	A factor or character vector indicating group membership for each observation. Must have length `nrow(X)`.
`x_show_parts`	Logical. If `TRUE`, the x-axis displays parts instead of group labels. Default is `TRUE`.
`include_boxplot`	Logical. If `TRUE`, a boxplot is overlaid on top of the barplot. Default is `FALSE`.
`clr_scale`	Logical. If `TRUE`, the data are transformed to clr coordinates before computing means. Default is `FALSE`.

Details

For each part, the function computes (within each group) the mean of either `log(X)` (default) or `clr(X)` (`clr_scale = TRUE`), and subtracts the overall mean across all observations. Therefore, bars represent deviations from the overall (global) mean on the chosen scale. Overlaying a boxplot can help visualize within-group variability.

Value

A `ggplot2` object representing the geometric mean barplot.

Examples

```
set.seed(1)
X <- matrix(runif(30, 1, 10), ncol = 3)
colnames(X) <- c("A", "B", "C")
group <- rep(c("G1", "G2"), each = 5)
geometric_mean_barplot(X, group, include_boxplot = TRUE)
geometric_mean_barplot(X, group, clr_scale = TRUE)
```

`ternary_diagram` *Ternary diagram for compositional data (D = 3)*

Description

Create a ternary diagram from compositional data with exactly three parts. Optionally center and/or scale the data in log-ratio coordinates, color points by group, and overlay the first two principal component directions computed in *ilr* coordinates.

Usage

```
ternary_diagram(
  X,
  group = NULL,
  center = FALSE,
  scale = FALSE,
  show_pc = FALSE
)
```

Arguments

X	A numeric matrix or data frame with exactly three columns (the parts of the composition). Values should be positive. Column names (if present) are used as corner labels.
group	Optional. A factor or character vector of length `nrow(X)` used to color points by group.
center	Logical. If `TRUE`, center the log-ratio coordinates before plotting. Default is `FALSE`.
scale	Logical. If `TRUE`, scale log-ratio coordinates to unit variance before plotting. Default is `FALSE`.
show_pc	Logical. If `TRUE`, overlay the first two principal component directions computed on log-ratio coordinates (recommended: *ilr*). Default is `FALSE`.

Details

Input requirements. `X` must have exactly three columns. For log-ratio methods, parts should be strictly positive. If `X` contains zeros or non-positive values, log-ratio transforms may fail; consider applying a zero-replacement strategy before calling this function.

Centering/scaling. When `center` and/or `scale` are enabled, the function transforms `X` to log-ratio coordinates (via `coda.base::coordinates()`), applies centering/scaling in Euclidean space, and maps back to the simplex using `coda.base::composition()`.

Principal components. If `show_pc = TRUE`, the function computes principal components in log-ratio space (recommended: *ilr* basis for $D = 3$) and draws the first two PC directions back in the simplex as curves on the ternary diagram. When `group` is provided, PCs are computed and drawn separately for each group.

Value

A `ggplot2` object.

See Also

`composition`, `coordinates`, `ilr_basis`

Examples

```
# Example with grouping
X <- milk_cows[, 5:7]
group <- milk_cows$group
ternary_diagram(X, group = group)

# With centering/scaling and principal components
ternary_diagram(X, group = group, center = TRUE, scale = TRUE, show_pc = TRUE)

# Without grouping
ternary_diagram(X, show_pc = TRUE)
```

Index

balance_dendrogram, 2

clr_biplot, 2
coda.plot, 4
coda.plot-package *(*coda.plot*)*, 4
composition, *6*
coordinates, *6*

geometric_mean_barplot, 4

ilr_basis, *6*

ternary_diagram, 5