Package: coda.plot (via r-universe)

May 12, 2026

Type Package

Title Plots for Compositional Data

Version 0.2.2

Date 2026-04-12

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 https://bsolidaria.r-universe.dev

Date/Publication 2026-04-12 07:47:07 UTC

RemoteUrl https://github.com/cran/coda.plot

RemoteRef HEAD

RemoteSha cd659b1e78573fc31fa685848fbfb46977364afc

Contents

`add_ternary_grid` *Add a ternary grid layer*

Description

Add a ternary grid layer

Usage

```
add_ternary_grid(p, ticks = seq(0.1, 0.9, 0.1), n = 300, eps = 1e-06, ...)
```

Arguments

`p`	A `ggplot2` object created by `ternary_plot()`.
`ticks`	Numeric vector of grid levels.
`n`	Number of sampled points per grid line.
`eps`	Small positive offset used before log-ratio transformation.
`...`	Further arguments passed to `ggplot2::geom_path()`.

Value

A `ggplot2` object.

`add_ternary_path` *Add a compositional path to a ternary plot*

Description

Add a compositional path to a ternary plot

Usage

```
add_ternary_path(p, X, group = NULL, transform = TRUE, ...)
```

Arguments

p	A ggplot2 object created by ternary_plot().
X	A numeric matrix or data frame with exactly three columns.
group	Optional grouping variable of length nrow(X) for multiple paths.
transform	Logical. If TRUE, apply the frame transformation.
...	Further arguments passed to ggplot2::geom_path().

Value

A ggplot2 object.

---

add_ternary_pc *Add principal component paths to a ternary plot*

Description

Add principal component paths to a ternary plot

Usage

```
add_ternary_pc(
  p,
  X,
  group = NULL,
  pcs = 1:2,
  basis = coda.base::ilr_basis(3),
  n = 600,
  eps = 0.001,
  ...
)
```

Arguments

p	A ggplot2 object created by ternary_plot().
X	A numeric matrix or data frame with exactly three columns.
group	Optional grouping variable of length nrow(X). If supplied, PCs are computed separately by group.
pcs	Integer vector indicating which principal components to draw.
basis	An ilr basis. Default is coda.base::ilr_basis(3).
n	Number of sampled points per PC path.
eps	Small positive threshold used to keep the path inside the simplex.
...	Further arguments passed to ggplot2::geom_path().

Value

A ggplot2 object.

add_ternary_points *Add compositional points to a ternary plot*

Description

Add compositional points to a ternary plot

Usage

```
add_ternary_points(p, X, group = NULL, transform = TRUE, ...)
```

Arguments

p	A ggplot2 object created by ternary_plot().
X	A numeric matrix or data frame with exactly three columns.
group	Optional grouping variable of length nrow(X).
transform	Logical. If TRUE, apply the frame transformation.
...	Further arguments passed to ggplot2::geom_point().

Value

A ggplot2 object.

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_coords *Transform compositional data into ternary plotting coordinates*

Description

Convert a compositional dataset into ternary plotting coordinates under a given `ternary_frame`.

Usage

```
ternary_coords(frame, X, transform = TRUE, group = NULL)
```

Arguments

`frame`	A `ternary_frame` object.
`X`	A numeric matrix or data frame with exactly three columns.
`transform`	Logical. If `TRUE`, apply the frame transformation before converting to ternary coordinates. Default is `TRUE`.
`group`	Optional grouping variable of length `nrow(X)`.

Value

A data frame with compositional columns `c1`, `c2`, `c3`, ternary coordinates `.x`, `.y`, and optionally `group`.

---

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 or numeric. If FALSE, no scaling is applied. If TRUE, log-ratio coordinates are scaled by their empirical standard deviations. If a single positive numeric value is supplied, centered log-ratio coordinates are multiplied by that value, so values larger than 1 increase visual spread and values between 0 and 1 shrink it.
show_pc	Logical. If TRUE, overlay the first two principal component directions computed on log-ratio coordinates (recommended: *ilr*). Default is FALSE.

Details

This function is kept as a convenient wrapper around the modular ternary API: ternary_frame(), ternary_plot(), add_ternary_points(), and add_ternary_pc().

Value

A ggplot2 object.

See Also

ternary_frame, ternary_plot, add_ternary_points, add_ternary_pc

Examples

```
X <- milk_cows[, 5:7]
group <- milk_cows$group

ternary_diagram(X, group = group)
ternary_diagram(X, group = group, center = TRUE, scale = TRUE)
ternary_diagram(X, group = group, center = TRUE, scale = 1.5)
ternary_diagram(X, show_pc = TRUE)
```

ternary_frame *Ternary frame for compositional data (D = 3)*

Description

Build a ternary plotting frame for compositional data with exactly three parts. The returned object stores the geometric and log-ratio transformation machinery needed to add multiple data layers consistently to the same ternary diagram.

Usage

```
ternary_frame(X, center = FALSE, scale = FALSE, labels = NULL)
```

Arguments

X	A numeric matrix or data frame with exactly three columns. This data defines the reference frame used for centering/scaling in log-ratio space.
center	Logical. If TRUE, center log-ratio coordinates using the mean coordinates of X. Default is FALSE.
scale	Logical or numeric. If FALSE, no scaling is applied. If TRUE, log-ratio coordinates are scaled by their empirical standard deviations. If a single positive numeric value is supplied, centered log-ratio coordinates are multiplied by that value.
labels	Optional character vector of length 3 used as corner labels. If NULL, colnames(X) are used when available. Otherwise, temporary labels c1, c2, c3 are used.

Value

An object of class "ternary_frame".

ternary_plot *Create a base ternary plot*

Description

Create the base ggplot object associated with a ternary_frame.

Usage

```
ternary_plot(
  frame = NULL,
  show_grid = TRUE,
  show_outline = TRUE,
  show_labels = TRUE,
  grid_ticks = seq(0.1, 0.9, 0.1)
)
```

Arguments

frame	Optional ternary_frame object. If NULL, a default ternary frame is created with no centering and no scaling.
show_grid	Logical. If TRUE, draw the ternary grid.
show_outline	Logical. If TRUE, draw the ternary triangle outline.
show_labels	Logical. If TRUE, draw corner labels.
grid_ticks	Numeric vector of grid levels.

Value

A ggplot2 object with the ternary_frame attached as attribute.

Index