Package ‘balance’

July 10, 2019

Title Visualize Balances of Compositional Data

Version 0.2.4

URL http://github.com/tpq/balance

BugReports http://github.com/tpq/balance/issues

Description Balances have become a cornerstone of compositional data analysis. However, conceptualizing balances is difficult, especially for high-dimensional data. Most often, investigators visualize balances with ``balance dendrograms''. However, this visualization tool does not scale well for large data. This package provides an alternative scheme for visualizing balances, described in [Quinn (2018) <DOI:10.12688/f1000research.15858.1>]. This package also provides a method for principal balance analysis.

License GPL-2

LazyData TRUE

VignetteBuilder knitr

RoxygenNote 6.1.0

Encoding UTF-8

Imports methods, ggplot2, grid

Depends R (>= 3.2.2)

Suggests ape, ggthemes, knitr, philr, propr, reshape2, rmarkdown, robCompositions, testthat, vegan

NeedsCompilation no

Author Thomas Quinn [aut, cre]

Maintainer Thomas Quinn <contacttomquinn@gmail.com>

Repository CRAN

Date/Publication 2019-07-10 05:40:03 UTC

R topics documented:

balance *Calculate and Visualize Balances*

Description

This function wraps balance.plot.

Usage

```
balance(...)
```

Arguments

...	Arguments to balance.plot.

balance.combine *Combine Two Sub-Plots*

Description

This function combines the "partition" sub-plot with the "distribution" sub-plot, preserving scale.

Usage

```
balance.combine(balance.partition, balance.distribution, size = "first")
```

Arguments

balance.partition	A ggplot object. The "partition" sub-plot.
balance.distribution	A ggplot object. The "distribution" sub-plot.
size	A string. Toggles whether to size final figure based on "first" (partition) or "last" (distribution) figure provided.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
res <- balance(x, sbp)
custom1 <- res[[1]] + ggplot2::theme_dark()
custom2 <- res[[2]] + ggplot2::theme_dark()
balance.combine(custom1, custom2)
```

balance.fromContrast *Transform Samples with the ilr of a Balance*

Description

Transform Samples with the ilr of a Balance

Usage

```
balance.fromContrast(x, contrast)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`contrast` A vector. One column of a serial binary partition matrix with values [-1, 0, 1] describing D components.

Value

A transformation of samples for the balance provided.

`balance.fromSBP` *Compute Balances from an SBP Matrix*

Description

Compute Balances from an SBP Matrix

Usage

```
balance.fromSBP(x, y)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`y` A serial binary partition matrix with rows as components (D) and columns as balances (D-1).

Value

A transformation of samples for each balance in the SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
balance.fromSBP(x, sbp)
```

balance.plot *Calculate and Visualize Balances*

Description

This function calculates balances based on the compositional data set and serial binary partition (SBP) matrix provided, then generates a figure from the results.

Usage

```
balance.plot(x, y, d.group, n.group, boxplot.split = TRUE,
  weigh.var = FALSE, size.text = 20, size.pt = 4)
```

Arguments

`x`	A matrix with rows as samples (N) and columns as components (D).
`y`	A serial binary partition matrix with rows as components (D) and columns as balances (D-1).
`d.group`	A vector of group labels for components. Optional. If provided, used to color component points.
`n.group`	A vector of group labels for samples. Optional. If provided, used to color sample points.
`boxplot.split`	A boolean. Toggles whether to split the boxplot by `n.group`. TRUE better resembles balance dendrogram style.
`weigh.var`	A boolean. Toggles whether to weigh line width by the proportion of explained variance. Only do this if balances come from an SBP that decomposes variance.
`size.text`	An integer. Sets legend text size.
`size.pt`	An integer. Sets point size.

Value

A list of the "partition" `ggplot` object, the "distribution" `ggplot` object, and the per-sample balances.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
balance(x, sbp)
```

bplot-class *A pba model S4 class*

Description

A pba model S4 class

Usage

```
## S4 method for signature 'bplot'
show(object)

## S4 method for signature 'bplot'
x[[i]]
```

Arguments

object, x	A bplot object.
i	An integer. Used to index the bplot object.

Methods (by generic)

- show: Method to show bplot object.
- [[: Method to subset bplot object.

Slots

balance.partition A ggplot object. The "partition" sub-plot.

balance.distribution A ggplot object. The "distribution" sub-plot.

balances The results of balance.fromSBP.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
balance(x, sbp)
```

packageCheck *Package Check*

Description

Checks whether the user has the required package installed. For back-end use only.

Usage

```
packageCheck(package)
```

Arguments

package	A character string. An R package.

pba *Principal Balance Analysis*

Description

Principal Balance Analysis

Usage

```
pba(x, how = "sbp.fromPBA", ...)

## S4 method for signature 'pba'
show(object)

## S4 method for signature 'pba'
predict(object, y)

## S4 method for signature 'pba,missing'
plot(x, y, group, pb1 = 1, pb2 = 2,
  size.text = 18)

## S4 method for signature 'pba,matrix'
plot(x, y, group, pb1 = 1, pb2 = 2,
  size.text = 18)

## S4 method for signature 'pba,data.frame'
plot(x, y, group, pb1 = 1, pb2 = 2,
  size.text = 18)
```

Arguments

`how`	A character string. The method used to construct the SBP. The default computes principal balances via `sbp.fromPBA`.
`...`	Arguments passed to `how` method.
`object, x`	A `pba` object.
`y`	A matrix on which to deploy the `pba` model.
`group`	A character vector. Group labels used to color points.
`pb1, pb2`	An integer. Sets principal balances to plot.
`size.text`	An integer. Sets legend text size.

Details

The `pba` function performs a principal balance analysis using the hierarchical clustering of components method described by Pawlowsky-Glahn et al. in "Principal balances" from the CoDaWork 2011 proceedings.

This resultant object contains the original data, the serial binary partition, the principal balances, and the fractional variances per balance. Use `predict` to deploy the `pba` model on new data.

Value

Returns a `pba` object.

Slots

`data` A matrix. The original data.

`sbp` A matrix. The SBP matrix.

`pba` A matrix. The balances.

`totvar` A numeric vector. The total variance per balance.

`subvar` A numeric vector. The fractional variance per balance.

Methods (by generic)

`show`: Method to show `pba` object.

`predict`: Method to deploy `pba` object.

`plot`: Method to plot `pba` object.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
train <- iris[1:50,1:4]
test <- iris[51:150,1:4]
model <- pba(train)
predict(model, test)
plot(model, test)
```

---

sbp.fromABA *Build SBP Matrix of "Anti-Principal Balances"*

Description

This function builds an SBP of "anti-principal balances" by clustering the difference of the log-ratio variance from the maximum log-ratio variance. Unlike principal balances, where the first balances explain the most variance, this function selects "anti-principal balances" so that the last balances explain relatively more variance.

Usage

```
sbp.fromABA(x, alpha = NA)
```

Arguments

x	A matrix with rows as samples (N) and columns as components (D).
alpha	A double. Defines a hyper-parameter used by the Box-Cox transformation to approximate log-ratio variance in the presence of zeros. Skip with NA.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp.fromABA(x)
```

sbp.fromADBA *Build SBP Matrix of "Anti-Principal Discriminant Balances"*

Description

This function builds an SBP of "discriminant balances" by clustering a matrix of the pair-wise total within-group variance, adjusted by the pair-wise total variance. The method is intended to make the smallest balances most discriminative.

Usage

```
sbp.fromADBA(x, group, ...)
```

Arguments

`x`	A matrix with rows as samples (N) and columns as components (D).
`group`	A character vector. Group or sub-group membership. Argument passed to `propr::propd`.
`...`	Other arguments passed to `propr::propd`.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
## Not run:
library(balance)
data(iris)
x <- iris[1:100,1:4]
y <- iris[1:100,5]
sbp.fromADBA(x, y)

## End(Not run)
```

`sbp.fromHclust` *Build SBP Matrix from hclust Object*

Description

This function builds an SBP matrix from an `hclust` object as produced by the `hclust` function.

Usage

```
sbp.fromHclust(hclust)
```

Arguments

`hclust`	An `hclust` object.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(cars)
h <- hclust(dist(cars))
sbp.fromHclust(h)
```

`sbp.fromPBA` *Build SBP Matrix of Principal Balances*

Description

This function builds an SBP of principal balances using the hierarchical clustering of components method described by Pawlowsky-Glahn et al. in "Principal balances" from the CoDaWork 2011 proceedings.

Usage

```
sbp.fromPBA(x, alpha = NA)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`alpha` A double. Defines a hyper-parameter used by the Box-Cox transformation to approximate log-ratio variance in the presence of zeros. Skip with NA.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp.fromPBA(x)
```

`sbp.fromPDBA` *Build SBP Matrix of "Principal Discriminant Balances"*

Description

This function builds an SBP of "discriminant balances" by clustering a matrix of the pair-wise total within-group variance, adjusted by the pair-wise total variance (inverted by subtracting this value from 1). The method is intended to make the largest balances most discriminative.

Usage

```
sbp.fromPDBA(x, group, ...)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`group` A character vector. Group or sub-group membership. Argument passed to `propr::propd`.

`...` Other arguments passed to `propr::propd`.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
## Not run:
library(balance)
data(iris)
x <- iris[1:100,1:4]
y <- iris[1:100,5]
sbp.fromPDBA(x, y)

## End(Not run)
```

`sbp.fromPropd` *Build SBP Matrix of "Anti-Principal Discriminant Balances"*

Description

A wrapper for `sbp.fromADBA`. See also `sbp.fromPDBA`.

Usage

```
sbp.fromPropd(x, group, ...)
```

Arguments

`x`	A matrix with rows as samples (N) and columns as components (D).
`group`	A character vector. Group or sub-group membership. Argument passed to `propr::propd`.
`...`	Other arguments passed to `propr::propd`.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
## Not run:
library(balance)
data(iris)
x <- iris[1:100,1:4]
y <- iris[1:100,5]
sbp.fromPropd(x, y)

## End(Not run)
```

sbp.fromRandom *Build SBP Matrix from Random Tree*

Description

This function builds an SBP from a random tree.

Usage

```
sbp.fromRandom(x)
```

Arguments

x	A matrix with rows as samples (N) and columns as components (D).

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp.fromRandom(x)
```

sbp.sort *Sort SBP Matrix*

Description

Sort SBP Matrix

Usage

```
sbp.sort(sbp)
```

Arguments

sbp	An SBP matrix.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
sbp.sort(sbp)
```

`sbp.subset` *Subset SBP Matrix*

Description

Subset SBP Matrix

Usage

```
sbp.subset(sbp, ternary = TRUE, ratios = TRUE)
```

Arguments

`sbp` An SBP matrix.

`ternary` A boolean. Toggles whether to return balances representing three components.

`ratios` A boolean. Toggles whether to return balances representing two components.

Value

An SBP matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
sbp.subset(sbp)
```

`ssBetween` *Calculate Between-Group Log-ratio Sums of Squares*

Description

This function calculates the between-group sums of squares for all variables or all log-ratios. This function only supports binary outcomes.

Usage

```
ssBetween(x, group, pairwise = FALSE, ...)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`group` A character vector. Group or sub-group membership. Argument passed to `propr::propd`.

`pairwise` A logical. Toggles whether to decompose sums of squares for each log-ratio.

`...` Other arguments passed to `propr::propd`.

Value

If `pairwise = FALSE`, this function returns a vector of the sums of squares for each variable. If `pairwise = TRUE`, this function returns a matrix of the sums of squares for each log-ratio.

Author(s)

Thom Quinn

`ssWithin` *Calculate Within-Group Log-ratio Sums of Squares*

Description

This function calculates the within-group sums of squares for all variables or all log-ratios. This function only supports binary outcomes.

Usage

```
ssWithin(x, group, pairwise = FALSE, ...)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`group` A character vector. Group or sub-group membership. Argument passed to `propr::propd`.

`pairwise` A logical. Toggles whether to decompose sums of squares for each log-ratio.

`...` Other arguments passed to `propr::propd`.

Value

If `pairwise = FALSE`, this function returns a vector of the sums of squares for each variable. If `pairwise = TRUE`, this function returns a matrix of the sums of squares for each log-ratio.

Author(s)

Thom Quinn

`vlr` *Calculate Log-ratio Variance*

Description

This function calculates the log-ratio variance for all components in a matrix.

Usage

```
vlr(x, alpha = NA)
```

Arguments

`x` A matrix with rows as samples (N) and columns as components (D).

`alpha` A double. Defines a hyper-parameter used by the Box-Cox transformation to approximate log-ratio variance in the presence of zeros. Skip with NA.

Value

A VLR matrix.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
vlr(x)
```

`wide2long` *Make Long Data from Wide Data*

Description

Make Long Data from Wide Data

Usage

```
wide2long(wide)
```

Arguments

`wide`	A data set in wide format.

Value

A data set in long format.

Author(s)

Thom Quinn

Examples

```
library(balance)
data(iris)
x <- iris[,1:4]
sbp <- sbp.fromPBA(x)
wide2long(sbp)
```

Index