Package ‘brsim’

December 12, 2023

Title Brainerd-Robinson Similarity Coefficient Matrix

Version 0.3

Description
Provides the facility to calculate the Brainerd-Robinson similarity coefficient for the rows of an input table, and to calculate the significance of each coefficient based on a permutation approach; a heatmap is produced to visually represent the similarity matrix. Optionally, hierarchical agglomerative clustering can be performed and the silhouette method is used to identify an optimal number of clusters; the results of the clustering can be optionally used to sort the heatmap.

Depends R (>= 4.0.0)

Imports cluster (>= 2.1.4), corrplot (>= 0.92), grDevices (>= 4.0.0),
RcmdrMisc (>= 2.7.0), stats (>= 4.0.0)

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Gianmarco Alberti [aut, cre]

Maintainer Gianmarco Alberti <`gianmarcoalberti@gmail.com`>

Repository CRAN

Date/Publication 2023-12-12 07:30:02 UTC

R topics documented:

brsim 2

Index **7**

brsim *Brainerd-Robinson similarity coefficient matrix calculation, with permutation-based p-values and optional clustering*

Description

This function calculates the Brainerd-Robinson (BR) similarity coefficient for each pair of row of the input table (Robinson-Brainerd 1951, 1952). It also performs a permutation test to assess the significance of each BR coefficient (DeBoer-Kintigh-Rostoker 1996), and allows to carry out a hierarchical agglomerative clustering. An optimal cluster solution can be established using the silhouette method (see details provided below). The function produces a correlation matrix in tabular form, which is also visually plotted as an heatmap. In the heatmap (which is built using the `corrplot` package), the size and the color of the squares are proportional to the Brainerd-Robinson coefficients. Optionally, the heatmap can be reordered on the basis of the hierachical clustering, with clusters enclosed by red rectangles.
Visit this LINK to access the package's vignette.

Usage

```
brsim(
  df,
  num.perm = 1000,
  clust = FALSE,
  part = NULL,
  aggl.meth = "ward.D2",
  sort.map = FALSE,
  number.cex = 0.7,
  cex.dndr.lab = 0.7,
  cex.sil.lab = 0.7,
  cex.dot.plt.lab = 0.7,
  oneplot = TRUE
)
```

Arguments

`df`	A table (dataframe format) where each row represents an assemblage and each column represents an item.
`num.perm`	A numeric value indicating the number of permutations to perform in each test (default is 1000).
`clust`	TRUE (default) or FALSE if the user does or does not want a agglomerative hierarchical clustering to be performed.
`part`	Desired number of clusters; if NULL (default), an optimal partition is calculated (see Details).

`aggl.meth` Agglomeration method ("ward.D2" by default) to be used; the selected method is internally used for the reordering of the heatmap if `order.map` is set to `TRUE`; for other methods see `hclust`.

`sort.map` TRUE or FALSE (default) if the user does or does not want the rendered heatmap to be ordered on the basis of the selected hierachical clustering.

`number.cex` Numeric. Set the size of the labels used for the coefficients displayed in the rendered heatmap.

`cex.dndr.lab` Numeric. Set the size of the labels used in the dendrogram.

`cex.sil.lab` Numeric. Set the size of the labels used in the silhouette plot.

`cex.dot.plt.lab` Numeric. Set the size of the labels used in the Cleveland's dotplots representing by-cluster proportions.

`oneplot` TRUE (default) or FALSE if the user wants or does not want the plots to be visualized in a single window.

Details

Permutation-based p-values

The rationale behind the p-value calculation is as follows: for each pair of assemblages in the input data, the function first calculates the observed Brainerd-Robinson (BR) coefficient. This is a measure of the similarity between the two assemblages.

The function then performs a certain number of permutations (default is 1000). In each permutation, it generates two new assemblages (each featuring a sample size corresponding to the size of each assemblage being compared) by randomly sampling from the global pool (the combined data of all assemblages), and calculates the BR coefficient for this new pair of assemblages (see DeBoer-Kintigh-Rostoker 1996). This creates a distribution of BR coefficients that we would expect to see by chance.

The p-value is then calculated as the proportion of the permuted BR coefficients that are less than or equal to the observed BR coefficient. A small p-value (typically < 0.05) suggests that the observed similarity between the two assemblages is statistically significant; it is unlikely to have occurred just by chance.

In simple terms, the p-value calculation is trying to answer the question: if there were no real similarity between these two assemblages, what is the probability that I would observe a similarity as extreme as (or more extreme than) the one I actually observed, just by chance?

The p-values are returned in two matrices: in the first, the p-values are reported as they are, whereas in the second they are classified as <0.05, <0.01, <0.001, or not significant.

Hierarchical agglomerative clustering

By setting the parameter `clust` to `TRUE`, the units (rows) for which the BR coefficients have been calculated will be clustered. Note that the clustering is based on a dissimilarity matrix which is internally calculated as the maximum value of the BR coefficient (200) minus the observed BR coefficient.

This allows a simpler reading of the dendrogram which is produced by the function, where the less dissimilar (i.e., more similar) units will be placed at lower levels, while more dissimilar (i.e., less similar) units will be placed at higher levels within the dendrogram. The latter depicts the hierarchical clustering based (by default) on the Ward's agglomeration method; rectangles identify the selected cluster partition.

Optionally, by setting the `sort.map` to `TRUE`, the heatmap can be reordered on the basis of the hierarchical clustering, with clusters indicated by red rectangles. The number of clusters indicated depends on what requested by the user (see the next section). Note that, internally, the reordering is based on the same agglomeration method selected by the user via the `aggl.method` parameter, which is set to `ward.D2` by default.

Number of clusters and silhouette method

Besides the dendrogram, a silhouette plot is produced, which allows to measure how 'good' is the selected cluster solution. If the parameter `part` is left empty (default), an optimal cluster solution is obtained.

The optimal partition is selected via an iterative procedure which identifies at which cluster solution the highest average silhouette width is achieved. The cluster solution ranges from a minimum of 2 to a maximum which is equal to the number of units (i.e., the rows of the input dataset) minus 1. The number of units essentially represents the maximum number of clusters that could potentially be formed if each row were its own cluster. However, since a cluster solution requires at least two groups, the maximum number of meaningful clusters is one less than the number of rows.

If a user-defined partition is needed, the user can input the desired number of clusters using the parameter `part`.

In either case, an additional plot is returned besides the cluster dendrogram and the silhouette plot; it displays a scatterplot in which the cluster solution (x-axis) is plotted against the average silhouette width (y-axis). A black dot represents the partition selected either by the iterative procedure or by the user.

Note that in the silhouette plot, the labels on the left-hand side of the chart show the units' names and the cluster number to which each unit is closer.

The silhouette plot is obtained from the `silhouette()` function out from the `cluster` package. For a detailed description of the silhouette plot, its rationale, and its interpretation, see Rousseeuw 1987.

Descriptive by-cluster dotplots

The function also provides a Cleveland's dotplots that represent by-cluster proportions. The clustered units are grouped according to their cluster membership, the frequencies are summed, and then expressed as percentages. The latter are represented by the dotplots, along with the average percentage. The latter provides a frame of reference to understand which percentage is below, above, or close to the average. The raw data on which the plots are based are stored in the list returned by the function (see below).

Value

The function returns a list storing the following components

- `BR.similarity.matrix`: similarity matrix reporting the BR coefficients.
- `P-value.matrix`: matrix reporting the permuted p-values.
- `classified.P-values.matrix`: matrix reporting the permuted p-value classified as <0.05, <0.01, <0.001, or not significant.
- `BR.distance_matrix`: distance matrix on which the hierarchical clustering is performed (returned if clustering is selected).
- `avr.silh.width.by.n.of.clusters`: average silhouette width by number of clusters (returned if clustering is selected).
- `partition.silh.data`: silhouette data for the selected partition (returned if clustering is selected).
- `data.with.cluster.membership`: copy of the input data table with an additional column storing the cluster membership for each row (returned if clustering is selected).
- `by.cluster.proportion`: table reporting the proportion of column categories across each cluster; rows sum to 100 percent (returned if clustering is selected).

References

Robinson, W. S. (1951). A Method for Chronologically Ordering Archaeological Deposits. In American Antiquity (Vol. 16, Issue 4, pp. 293–301). Cambridge University Press.

Robinson, W. S., & Brainerd, G. W. (1952). Robinson's Coefficient of Agreement – A Rejoinder. In American Antiquity (Vol. 18, Issue 1, pp. 60–61). Cambridge University Press.

Rousseeuw P J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. In Journal of Computational and Applied Mathematics 20, 53-65.

DeBoer, W. R., Kintigh, K., & Rostoker, A. G. (1996). Ceramic Seriation and Site Reoccupation in Lowland South America. In Latin American Antiquity (Vol. 7, Issue 3, pp. 263–278). Cambridge University Press.

See Also

`corrplot` , `silhouette`

Examples

```
# build a toy dataset (subset of the 'Nelson' dataset out of the 'archdata' package )

mytable <- structure(list(Biscuit = c(10, 17, 2, 10, 2, 1),
Type_I = c(2,2, 10, 40, 118, 107),
Type_II_Red = c(24, 64, 68, 91, 45, 3),
Type_II_Yellow = c(23, 90, 18, 20, 1, 0),
Type_II_Gray = c(34,76, 48, 15, 5, 0)),
row.names = c("1", "2", "3", "7", "8", "9"),
class = "data.frame")

# run the function and store the results in the 'result' object
```

```
result <- brsim(mytable, clust=TRUE)

# same as above, but with an user-defined cluster partition

result <- brsim(mytable, clust=TRUE, part=3)

# same as above, but rendering with a reordered heatmap

result <- brsim(mytable, clust=TRUE, part=3, sort.map=TRUE)
```

Index

∗ **similarity**
    brsim, 2

brsim, 2

corrplot, *5*

hclust, *3*

silhouette, *5*