divo

Diversity and Overlap Analysis Package
User's Guide

Maciej Pietrzak, Michal Seweryn
and Grzegorz Rempala

January 28, 2015

Contents

`divo Package Information`

Version 0.1

Type Package

Authors Maciej Pietrzak, Michal Seweryn, Grzegorz Rempala

Maintainer Maciej Pietrzak `<pietrzak.20@osu.edu>`

Depends R (>= 3.1.0), RcppCNPy, cluster, Python (>=2.76), Numpy

Description Package implements empirical analysis of diversity and similarity (overlap) in biological or ecological systems.

License GPL (>=2)

divo Overview *divo Diversity and Overlap Analysis Package*

Description

Package implements various algorithms for empirical analysis of diversity and similarity (overlap) in biological or ecological systems. The implemented indices of diversity and overlap are based both on the information-theoretic and geometric considerations. The indices have the capacity to naturally up-weight or down-weight rare and abundant population species counts, by applying the Good-Turing sample coverage correction. The functional version of a diversity index, the so-called diversity profile, is also implemented along with the diversity and overlap indices inversions known as the effective numbers of species (ENS). **For examples and detailed information on specific functions, see their manual pages:**

`cvg`	Coverage
`dp`	Diversity Profile
`dp.ht`	Diversity Profile with the Horvitz-Thompson Correction
`ens`	Effective Number of Species
`ens.ht`	Effective Number of Species with the Horvitz-Thompson Correction
`i.in`	Information Index (I-index) for 2-Way Table
`i.inp`	Information Index (I-index) for 2 Way, 2 Column Table
`ji`	Jaccard Index
`li`	Sorensen Index
`mh`	Morisita-Horn Index
`pg`	Power-Geometric Index
`pg.ht`	Power-Geometric Index with the Horvitz-Thompson Correction
`rd`	Renyi's Divergence
`srd`	Symmetrized Renyi's Divergence

divo System Requirements

Description

divo package requires R >=3.1.0; RcppCNPy and cluster packages; Python >= 2.76 with NumPy.

Installation of Python and NumPy.

Python is already installed in computers with an OSX or a Linux OS. To install Python in Windows visit `https://www.python.org/downloads/`; download the latest version of python and follow the installation instruction. For more information visit `https://www.python.org/doc/`

To install NumPy go to `http://sourceforge.net/projects/numpy/files/NumPy`, download the latest version of software appropriate for your operating system and follow installation manual. For more information visit `http://www.numpy.org`.

In Linux both Python and Numpy may be installed using system package manger e.g. ”synaptic” in Debian distribution.

Setting path on Windows.

1. Open System Properties and click Advanced system settings > Advanced tab;
2. Click on Environment Variables, under System Variables, find PATH, and click on it;
3. In the Edit window, modify PATH by adding the location of Python folder.

If you do not have the item PATH, you may select to add a new variable and add PATH as the name and the location of the Python folder.

cvg Coverage

Description

Calculates the sample coverage estimate using the Good-Turing formula. The sample coverage is an estimate of the probability of pulling a new species in the next draw, given a set of past observations For more details on CVG see Good I.J. (1953).

Usage

```
cvg(x)
```

Arguments

x	a vector containing input population

Examples

```
data(TCR.Data)
result <- cvg(x[,1])
```

dp Diversity Profile

Description

Calculates diversity profile (DP) (Rempala and Seweryn 2013 or Tothmeresz 1995) using the Renyi entropy (Renyi 1961) as a diversity measure. The function calculates the Renyi entropy values for a given range of the Renyi index (the index should be greater than 0). When the index is less then one, the rare counts are up-weighted and when it is greater than one, the rare counts are down-weighted. Since the Renyi entropy is a non-increasing function of the index, the profile plot should be always non-increasing.

Usage

```
dp(x, alpha = seq(0.1, 2, 0.1), CI = 0.95, resample = 100,
single_graph = FALSE, pooled_graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` a matrix containing alpha values, default = seq(0.1, 2, 0.1)

`CVG` a vector containing alpha values multiplied by coverage; default = FALSE

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`single_graph` default = FALSE, plot of the Diversity Profile for each population; single_graph = 'fileName' user-defined output file name

`pooled_graph` default = FALSE, plot of the Diversity Profile for all populations; pooled_graph = 'fileName' user-defined output file name

`csv_output` save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- dp(x, resample = 50)
```

dp.ht Diversity Profile with the Horvitz-Thompson Adjustment

Description

Calculates diversity profile with the Horvitz-Thompson adjustment (DP-HT), as defined in Rempala and Seweryn (2013) using the Renyi entropy (Renyi 1961) as a diversity measure. The function calculates the Renyi entropy values for a given range of the Renyi index (the index should be greater than 0). When the index is less then one, the rare counts are up-weighted and when it is greater than one, the rare counts are down-weighted. Since the Renyi entropy is a non-increasing function of the index, the profile plot should be always non-increasing. For more information, see Rempala and Seweryn (2013).

Usage

```
dp.ht(x, alpha = seq(0.1, 2, 0.1), CI = 0.95, resample = 100,
single_graph = FALSE, pooled_graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` a vector containing alpha values, default = seq(0.1, 2, 0.1)

`CVG` a vector containing alpha values multiplied by coverage; default = FALSE

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`single_graph` default = FALSE, plot of the Diversity Profile for each population; single_graph = 'fileName' user-defined output file name

`pooled_graph` default = FALSE, plot of the Diversity Profile for all populations; pooled_graph = 'fileName' user-defined output file name

`csv_output` save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- dp.ht(x, resample = 50)
```

ens Effective Number of Species

Description

Calculates diversity profile (DP) using the effective number of species (ENS) based on inverting the Renyi entropy. For any monotone diversity index (see, Rempala and Seweryn 2013) the ENS is defined as the size of a uniform population with the same index value as the current population. The ENS may be considered as a measure of population diversity expressed in the units of species counts. The ENS profile is calculated against the Renyi entropy index, which allows for a direct comparison with the diversity profile (as in `dp`). The option of performing the Horvitz-Thompson correction is available in the function `ens.ht`. For more details on ENS, see Rempala and Seweryn (2013) or Jost (2006).

Usage

```
ens(x, alpha = seq(0.1, 2, 0.1), CI = 0.95, resample = 100,
single_graph = FALSE, pooled_graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x`	a matrix containing input populations
`alpha`	a vector containing alpha values, default = seq(0.1, 2, 0.1)
`CVG`	a vector containing alpha values multiplied by coverage; default = FALSE
`CI`	Confidence Interval default = 0.95, range (0, 1)
`resample`	set number of repetitions, default = 100
`single_graph`	default = FALSE, plot of the Diversity Profile for each population; single_graph = 'fileName' user-defined output file name
`pooled_graph`	default = FALSE, plot of the Diversity Profile for all populations; pooled_graph = 'fileName' user-defined output file name
`csv_output`	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
`PlugIn`	standard plug-in estimator, default = FALSE
`size`	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)
`saveBootstrap`	Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- ens(x, resample = 50)
```

`ens.ht Effective Number of Species with the Horvitz-Thompson Correction`

Description

Calculates diversity profile (DP) using the effective number of species (ENS) based on inverting the Renyi entropy with the Horvitz-Thompson correction. For any monotone diversity index (see, e.g., Rempala and Seweryn 2013) the ENS is defined as the size of a uniform population with the same index value as the current population. The ENS may be considered as a measure of population diversity expressed in the units of species counts. The ENS profile is calculated against the Renyi entropy index, which allows for a direct comparison with the diversity profile (as in `dp`). The ENS without the Horvitz-Thompson correction is available as function `ens`. For more details on ENS see Rempala and Seweryn (2013) or Jost (2006).

Usage

```
ens.ht(x, alpha = seq(0.1, 2, 0.1), CI = 0.95, resample = 100,
single_graph = FALSE, pooled_graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` a vector containing alpha values, default = seq(0.1, 2, 0.1)

`CVG` a vector containing alpha values multiplied by coverage; default = FALSE

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`single_graph` default = FALSE, plot of the Diversity Profile for each population; single_graph = 'fileName' user-defined output file name

`pooled_graph` default = FALSE, plot of the Diversity Profile for all populations; pooled_graph = 'fileName' user-defined output file name

`csv_output` save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- ens.ht(x, resample = 50)
```

`i.in Information Index (I index) for 2-Way Table`

Description

The I-index is a measure of overlap in two way tables based on the generalized mutual information statistic. The I-index measures dependence among columns of two-way tables, taking values between 0 and 1. It returns a value of zero when the table columns form an orthogonal system and a value of one when the table columns rank is one. The value of the parameter alpha is related to the structure of dependence, as described in Rempala and Seweryn (2013).

Usage

```
i.in(x, alpha = 1, CI = 0.95, resample = 100, PlugIn = FALSE, size = 1,
CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` I index of order *alpha*, must be between 0 and 1, default = 0.5

`CVG` I index of order *alpha* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- i.in(x, resample = 50)
```

`i.inp Information Index (I index) for 2 Way, 2 Column Table`

Description

The I-index is a measure of overlap in two way tables based on the generalized mutual information statistic. This function implements a special case of table with two columns only. In general, the I-index measures dependence in any two-way tables, taking values between 0 and 1. It returns a value of zero when the table columns form an orthogonal system and a value of one when the table columns rank is one. The value of the parameter alpha is related to the structure of dependence, as described in Rempala and Seweryn (2013).

Usage

```
i.inp(x, alpha = 1, CI = 0.95, resample = 100, graph = FALSE,
csv_output = FALSE, PlugIn = FALSE, size = 1, CVG = FALSE,
saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` I index of order *alpha* < 1 puts more weight on the rare species and the I Index of order *alpha* > 1 puts more weight on the abundant ones, default = 1

`CVG` I index of order *alpha* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE

`CI`	Confidence Interval default = 0.95, range (0, 1)
`resample`	set number of repetitions, default = 100
`graph`	default = FALSE, plot the results of hierarchical clustering of pairwise analysis of I Index; `graph = 'fileName'` user-defined output file name
`csv_output`	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
`PlugIn`	standard plug-in estimator, default = FALSE
`size`	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)
`saveBootstrap`	Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- i.inp(x, resample = 50)
```

ji Jaccard Index

Description

The Jaccard similarity (overlap) index measures the size of the intersection of two populations relative to size of their union. It varies between zero (no overlap) and one (perfect overlap). The Jaccard Index is closely related the Sorensen (implemented in function `li`) and the Dice indices which are widely used in both the ecological and immunological literature (see, Rempala and Seweryn 2013).

Usage

```
ji(x, CI = 0.95, resample = 100, graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, saveBootstrap = FALSE)
```

Arguments

`x`	a matrix containing input populations
`CI`	Confidence Interval default = 0.95, range (0, 1)
`resample`	set number of repetitions, default = 100
`graph`	default = FALSE, plot the results of hierarchical clustering of pairwise analysis of Jaccard Index; `graph = 'fileName'` user-defined output file name
`csv_output`	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
`PlugIn`	standard plug-in estimator, default = FALSE
`size`	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

saveBootstrap Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- ji(x, resample = 50)
```

li Sorensen Index

Description

The Sorensen similarity (overlap) index measures the overlap between two populations by taking the ratio of the number of species shared between the two populations, relative to the number of species in both populations. The index varies between zero (no overlap) and one (perfect overlap). It is closely related to the Jaccard index of similarity (implemented in function `ji`).

Usage

```
li(x, CI = 0.95, resample = 100, graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, saveBootstrap = FALSE)
```

Arguments

x a matrix containing input populations

CI Confidence Interval default = 0.95, range (0, 1)

resample set number of repetitions, default = 100

graph default = FALSE, plot the results of hierarchical clustering of pairwise analysis of Sorensen Index; `graph = 'fileName'` user-defined output file name

csv_output save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name

PlugIn standard plug-in estimator, default = FALSE

size resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

saveBootstrap Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- li(x, resample = 50)
```

mh Morisita-Horn Index

Description

The Morisita-Horn index is a popular angular overlap measure used both in ecological and immunological literature. It quantifies overlap as cosine of an angle between two standardized population vectors. It ranges between zero (no overlap) and one (perfect overlap). MH tends to be over-sensitive to abundant species. For details see Rempala and Seweryn (2013) or Magurran (2005).

Usage

```
mh(x, CI = 0.95, resample = 100, graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`graph` default = FALSE, plot the results of hierarchical clustering of pairwise analysis of Morisita-Horn Index; `graph = 'fileName'` user-defined output file name

`csv_output` save the result of the analysis as .CSV file, default = FALSE; csv_output = ’fileName’ user-defined output file name

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = ’FolderName’ - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- mh(x, resample = 50)
```

`pg Power-Geometric Index`

Description

The Power Geometric (PG) index is a geometric angular overlap measure parameterized by a two-dimensional vector (alpha, beta). The PG index is a generalization of the Morisita-Horn index as well as the Bhattacharyya's coefficient. It allows for increasing or decreasing the relative contribution of the rare species to the overall overlap and may be therefore used to account for the species undersampling. It quantifies overlap as cosine of an angle between two exponentially normalized population vectors. For further details and definition, see Rempala and Seweryn (2013).

Usage

```
pg(x, alpha = 1, beta=alpha, CI = 0.95, resample = 100, graph = FALSE,
csv_output = FALSE, PlugIn = FALSE, size = 1, CVG = FALSE,
saveBootstrap = FALSE)
```

Arguments

`x`	a matrix containing input populations
`alpha`	PG of order *alpha* $<$ 1 puts more weight on the rare species and the I Index of order *alpha* $>$ 1 puts more weight on the abundant ones for first population, default = 1
`beta`	PG of order *beta* $<$ 1 puts more weight on the rare species and the I Index of order *beta* $>$ 1 puts more weight on the abundant ones for second population, default = alpha
`CVG`	PG of order *alpha* or *beta* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE
`CI`	Confidence Interval default = 0.95, range (0, 1)
`resample`	number of repetitions, default = 100
`graph`	default = FALSE, plot the results of hierarchical clustering of pairwise analysis of Power-Geometric Index; `graph = 'fileName'` user-defined output file name
`csv_output`	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
`PlugIn`	standard plug-in estimator, default = FALSE
`size`	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)
`saveBootstrap`	Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- pg(x, resample = 50)
```

pg.ht Power-Geometric Index with the Horvitz-Thompson Correction

Description

The Horvitz-Thompson corrected version of the Power Geometric (PG) index (see help for pg). The PG index is a generalization of the Morisita-Horn index as well as the Bhattacharyya's coefficient. It quantifies overlap as cosine of an angle between two exponentially normalized population vectors. For further details and definitions, see Rempala and Seweryn (2013).

Usage

```
pg.ht(x, alpha = 1, beta=alpha, CI = 0.95, resample = 100, graph = FALSE,
csv_output = FALSE, PlugIn = FALSE, size = 1, CVG = FALSE,
saveBootstrap = FALSE)
```

Arguments

`x` a matrix containing input populations

`alpha` PG of order *alpha* <1 puts more weight on the rare species and the I Index of order *alpha* >1 puts more weight on the abundant ones for first population, default = 1

`beta` PG of order *beta* <1 puts more weight on the rare species and the I Index of order *beta* >1 puts more weight on the abundant ones for second population, default = alpha

`CVG` PG of order *alpha* or *beta* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE

`CI` Confidence Interval default = 0.95, range (0, 1)

`resample` set number of repetitions, default = 100

`graph` default = FALSE, plot the results of hierarchical clustering of pairwise analysis of Power-Geometric Index, `graph = 'fileName'` user-defined output file name

`csv_output` save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name

`PlugIn` standard plug-in estimator, default = FALSE

`size` resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)

`saveBootstrap` Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- pg.ht(x, resample = 50)
```

`rd` Renyi's Divergence

Description

The Renyi divergence (RD) is a measure of similarity between two discrete probability distributions. The Renyi divergence is non-negative, not symmetric, and is not defined when there is no common support between two distributions RD is parameterized by a single non-negative parameter which may be used to adjust the relative contributions of small and large probabilities to its overall value. RD is a generalization of the Kullback-Leibler divergence. For details, see Rempala and Seweryn (2013).

Usage

```
rd(x, alpha = 0.5, CI = 0.95, resample = 100, graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

`x`	a matrix containing input populations
`alpha`	Renyi's Divergence index of order *alpha* $<$ 1 puts more weight on the rare species and the I Index of order *alpha* $>$ 1 puts more weight on the abundant ones, default = 1
`CVG`	Renyi's Divergence index of order *alpha* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE
`CI`	Confidence Interval default = 0.95, range (0, 1)
`resample`	number of repetitions, default = 100
`graph`	default = FALSE, plots the results of hierarchical clustering of pairwise analysis of Renyi's Divergence; `graph = 'fileName'` user-defined output file name
`csv_output`	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
`PlugIn`	standard plug-in estimator, default = FALSE
`size`	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)
`saveBootstrap`	Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- rd(x, resample = 50, alpha=0.5)
```

srd Symmetrized Renyi's Divergence

Description

The symmetrized Renyi divergence (RD) is a measure of similarity between two discrete probability distributions which is non negative and symmetric. For details, see the description of function rd or Rempala and Seweryn (2013).

Usage

```
srd(x, alpha = 0.5, CI = 0.95, resample = 100, graph = FALSE, csv_output = FALSE,
PlugIn = FALSE, size = 1, CVG = FALSE, saveBootstrap = FALSE)
```

Arguments

x	a matrix containing input populations
alpha	Renyi's Divergence index of order *alpha* must be between 0 and 1, default = 0.5
CVG	Renyi's Divergence index of order *alpha* = coverage. If CVG = TRUE argument *alpha* is ignored; default = FALSE
CI	Confidence Interval default = 0.95, range (0, 1)
resample	number of repetitions, default = 100
graph	default = FALSE, plots the results of hierarchical clustering of pairwise analysis of Renyi's Divergence; `graph = 'fileName'` user-defined output file name
csv_output	save the result of the analysis as .CSV file, default = FALSE; csv_output = 'fileName' user-defined output file name
PlugIn	standard plug-in estimator, default = FALSE
size	resampled fraction of the population, default = 1 (actual size of populations). The value should not be smaller than 10% of population (size = 0.1)
saveBootstrap	Saves bootstrap result to a file. Use `saveBootstrap = TRUE` to save bootstrap results to a `Bootstrap` folder in current directory; saveBootstrap = 'FolderName' - saves bootstrap results to user-named folder

Examples

```
data(TCR.Data)
result <- srd(x, resample = 50, alpha=0.5)
```

TCR.Data *TCR.Data Repertoires of Naive and Regulatory T-cell Populations*

Description

T-cell receptor repertoires sequenced using Ion Torrent technology. Dataset contains receptors found in four different organs, each with two functional populations (naive and regulatory (Treg)). Cells are isolated from colon (Col), peripheral lymph nodes (PLN), mesenteric lymph nodes (MLN) and thymus (Thym). TCR populations data are stored in a matrix (object named `x`). Each column of x contains sequenced counts of specific TCR variants in given organ population `x`.

Examples

```
data(TCR.Data)
head(x)
```

References

1. Cebula A., Seweryn M., Rempala G.A., Pabla S.S., McIndoe R.A., Denning T.L., Bry L., Kraj P., Kisielow P., Ignatowicz L. (2013) Thymus-derived regulatory T cells contribute to tolerance to commensal microbiota. Nature 497:258-62
2. Good I.J. The population frequencies of species and the estimation of population parameters. (1953) Biometrika 40:237-64
3. Jost L. (2006) Entropy and diversity. Oikos 113:363-75 Magurran A.E. (2005) Biological diversity. Curr Biol 15:R116-8
4. Rempala G.A., Seweryn M. (2013) Methods for diversity and overlap analysis in T-cell receptor populations. J Math Biol 67:1339-68
5. Renyi P. (1961) On measures of information and entropy. In: Proceedings of the 4th Berkeley symposium on mathematics, statistics and probability, pp 547-61
6. Tothmeresz B. (1995) Comparison of different methods for diversity ordering. J Veget Sci 6:283-90