Package: ccrepe (via r-universe)

November 13, 2026

Type Package

Title ccrepe_and_nc.score

Version 1.48.0

Imports infotheo (>= 1.1)

Date 2024-02-06

Author Emma Schwager <emh146@mail.harvard.edu>,Craig Bielski<craig.bielski@gmail.com>, George Weingart<george.weingart@gmail.com>

Maintainer Emma Schwager <emma.schwager@gmail.com>,Craig Bielski<craig.bielski@gmail.com>, George Weingart<george.weingart@gmail.com>

Description The CCREPE (Compositionality Corrected by REnormalizaion and PErmutation) package is designed to assess the significance of general similarity measures in compositional datasets. In microbial abundance data, for example, the total abundances of all microbes sum to one; CCREPE is designed to take this constraint into account when assigning p-values to similarity measures between the microbes. The package has two functions: ccrepe: Calculates similarity measures, p-values and q-values for relative abundances of bugs in one or two body sites using bootstrap and permutation matrices of the data. nc.score: Calculates species-level co-variation and co-exclusion patterns based on an extension of the checkerboard score to ordinal data.

License MIT + file LICENSE

VignetteBuilder knitr

Suggests knitr, BiocStyle, BiocGenerics, testthat, RUnit

biocViews ImmunoOncology, Statistics, Metagenomics, Bioinformatics, Software

PackageStatus Deprecated

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:38:29 UTC

RemoteUrl https://github.com/bioc/ccrepe

RemoteRef RELEASE_3_23

RemoteSha c620a51b173a9557e81e330481dfa96b4ee91992

Contents

ccrepe-package 2
ccrepe 3
ccrepeSampleTestFunction 6
nc.score 7

Index **9**

`ccrepe-package` *A package for analysis of sparse compositional data. Allows calculation of the similarity measure nc-score and calculation of compositionality-corrected p-values for arbitrary similarity scores (including user-defined) applied to compositional data.*

Description

ccrepe was developed for use with microbial relative abundance data data, which is both sparse and compositional in nature.

Details

Package:	ccrepe
Type:	Package
Version:	1.0
Date:	2013-04-18
License:	MIT

Author(s)

Emma Schwager <emma.schwager@gmail.com>, Craig Bielski<craig.bielski@gmail.com>
Maintainer: Emma Schwager <emma.schwager@gmail.com>,
Craig Bielski<craig.bielski@gmail.com>,
George Weingart<george.weingart@gmail.com>

ccrepe *Calculates compositionality-corrected p-values and q-values for compositional data using an arbitrary distance metric.*

Description

ccrepe calculates compositionality-corrected p-values and q-values for compositional data by generating first a null distribution of the distance metric generated by permutation and renormalization of the data, and then by generating an alternative distribution of the distance metric by bootstrap resampling of the data. For greater detail, see References

The two distributions are compared using a pooled-variance Z-test to give a compositionality-corrected p-value. The p-values can be calculated for all appropriate (passing certain quality-control measures) pairwise comparisons, or for a subset of user-specified ones.

Q-values are additionally calculated using the Benjamin-Hochberg-Yekutieli procedure (see References)

Usage

```
ccrepe(
x = NA,
y = NA,
sim.score = cor,
sim.score.args = list(),
min.subj = 20,
iterations = 1000,
subset.cols.x = NULL,
subset.cols.y = NULL,
errthresh  = 1e-04,
verbose = FALSE,
iterations.gap = 100,
distributions = NA,
compare.within.x = TRUE,
concurrent.output = NA,
make.output.table = FALSE)
```

Arguments

x	First dataframe or matrix containing the relative abundances in cavity1 : columns are bugs, rows are samples. (Rows should therefore sum to a constant.) The subjectIDs, if present, are assumed to be the row names and NOT the first column of data.
y	Second dataframe or matrix (optional) containing the relative abundances in cavity2: columns are bugs, rows are samples. The subjectIDs, if present, are assumed to be the row names. If both x and y are

specified, they will be merged by row names. If no row names are specified for either or both datasets, the default is to use the row numbers as subject IDs.

`sim.score` A function defining a similarity measure, such as cor or nc.score. This similarity measure can be a pre-defined R function or user-defined. If the latter, certain properties should be satisfied as detailed below (also see examples). The default similarity measure is Spearman correlation.
A user-defined similarity measure should:
1.Be able to take either two inputs which are vectors or one input which is either a matrix or a dataframe
2.In the case of two inputs, return a single number
3.In the case of one input, return a matrix in which the (i,j)th entry is the similarity score for column i and column j in the original matrix
4.Resulting matrix (in the case of one input) must be symmetric
5.The inputs must be named x and y

`sim.score.args` A list of arguments for the measurement function. For example: In the case of cor, the following would be acceptable: sim.score.args = list(method='spearman',use='complete.obs').

`min.subj` Minimum number of samples that must be non-missing in a bug/feature/column in order to apply the similarity measure to that bug/feature/column. This is to ensure that there are sufficient subjects to perform a bootstrap (default: 20).

`iterations` The number of iterations of bootstrap and permutation (default: 1000).

`subset.cols.x` A vector of column indices from x to indicate which features to compare

`subset.cols.y` A vector of column indices from y to indicate which features to compare

`errthresh` If feature has number of zeros greater than errthresh^(1/n) , that feature is excluded

`verbose` Logical: an indicator whether the user requested verbose output, which prints periodic progress of the algorithm through the dataset(s), as well as including more detailed output. (default:FALSE)

`iterations.gap` If output is verbose - number of iterations after issue a status message (Default=100 - displayed only if verbose=TRUE).

`distributions` Output Distribution file (default:NA).

`compare.within.x`
A boolean value indicating whether to do comparisons given by taking all subsets of size 2 from subset.cols.x or to do comparisons given by taking all possible combinations of subset.cols.x or subset.cols.y. If TRUE but subset.cols.y=NA, returns all comparisons involving any features in subset.cols.x. This argument is only used when y=NA.

`concurrent.output`
Optional output file to which each comparison will be written as it is calculated.

`make.output.table`
A boolean value indicating whether to include table-formatted output.

Value

Returns a list containing the calculation results and the parameters used.
Default parameters shown:

`min.subj`	Description above
`errThresh`	Description same as errthresh above
`sim.score`	A matrix of the simliarity scores for all the requested comparisons. The (i,j)th element of sim.score correponds to the similarity score of column i (or the ith column of subset.cols.1) and column j (or the jth column of subset.cols.1) in one dataset, or to the similarity score of column i (or the ith column of subset.cols.1) in dataset x and column j (or the jth column of subset.cols.2)in dataset y in the case of two datasets.
`p.values`	A matrix of the p-values for all the requested comparisons. The (i,j)th element of p.values corresponds to the p-value of the (i,j)th element of sim.score.
`q.values`	A matrix of the Benjamini-Hochberg-Yekutieli FDR corrected p-values. The (i,j)th element of q.values corresponds to the q-value fo the (i,j)th element of sim.score.
`z.stat`	A matrix of the z-statistics for all the requested comparisons. The (i,j)th element corresponds to the z-statistic which gave rise to the (i,j)th p-value.
`output.table`	(Only if make.output.table=TRUE) A table where each row is one comparision. Each row contains the features being compared with their similarity scores, z-statistics, p-values and q-values

Additional parameters if verbose=TRUE:

`iterations`	Description Above
`subset.cols.x`	Description Above
`subset.cols.y`	Description Above
`iterations.gap`	Description Above
`sim.score.parameters`	Description Above
`compare.within.x`	Description Above
`make.output.table`	Description Above

Author(s)

Emma Schwager <emma.schwager@gmail.com>

References

Emma Schwager and Colleagues. Detecting statistically significant associtations between sparse and high dimensional compositioanl data. In Progress.

Benjamini and Yekutieli (2001). "The control of the false discovery rate in multiple testing under dependency." The Annals of Statistics. Vol. 19, No. 4. pp. 1165-1188.

Examples

```
data <- matrix(rlnorm(40,meanlog=0,sdlog=1),nrow=10)
data.rowsum <- apply(data,1,sum)
data.norm <- data/data.rowsum
testdata <- data.norm
dimnames(testdata) <- list(paste("Sample",seq(1,10)),paste("Feature",seq(1,4)))
ccrepe.results  <-ccrepe  (x=testdata, iterations=20, min.subj=10)
ccrepe.results.nc.score <- ccrepe(x=testdata,iterations=20,min.subj=10,sim.score=nc.score)
ccrepe.results
ccrepe.results.nc.score
```

---

ccrepeSampleTestFunction

ccrepeSampleTestFunction - Simple example of a test measurent function to be used with ccrepe

---

Description

This simple example of a test measurent function to be used with ccrepe
used in the same fashion that cor would be used
Some properties of the function:
1. Be able to take either two inputs which are vectors or one input which
is either a matrix or a data frame
3.In the case of one input, return a matrix in which the (i,j)th entry
is the similarity score for column i and column j in the original matrix
4.Resulting matrix must be symmetric
5.The inputs must be named x and y

Usage

```
ccrepeSampleTestFunction(x, y = NA)
```

Arguments

x	x is a vector or a matrix
y	y is a vector. if y selected then x must be a vector too

Value

If x and y are vectors it returns a number: 0.5 If x is a matrix it returns a matrix of all 0.5

Author(s)

Emma Schwager <emma.schwager@gmail.com>

nc.score *nc.score*

Description

nc.score calculates species-level co-variation and co-exclusion patterns based on an extension of the checkerboard score to ordinal data.

It is an extension to Diamond's checkerboard score (See references below) to ordinal data and implements a framework for robust detection of species-level association patterns in metagenomic data.

Usage

```
nc.score(x,
 y = NULL,
 use = "everything",
 nbins = NULL,
 bin.cutoffs = NULL
)
```

Arguments

x	A numeric vector, data frame, or matrix. The first entity to be processed. Columns are bugs, rows are samples.
y	NULL(default) or a umeric vector, data frame, or matrix with compatible dimensions to x. Columns are features, rows are samples.
use	An optional character string givinga method for computing covariances in the presence of missing values. This must be (an abbreviaion of) on of the strings "everything", "all.obs", "complete.obs","na.or.complete", or "pairwise.complete.obs".
nbins	A non-negative integer of the number of bins to generate (cutoffs will be generated by the discretize function from the infotheo package).
bin.cutoffs	A list of values demarcating the bin cutoffs. The binning is performed using the findInterval function.

Value

Matrix or vector of normalized scores.

Author(s)

Craig Bielski<craig.bielski@gmail.com>

References

Emma Schwager and Colleagues. Detecting statistically significant associtations between sparse and high dimensional compositioanl data. In Progress.

Examples

```
data <- matrix(rlnorm(40,meanlog=0,sdlog=1),nrow=10)
data.rowsum <- apply(data,1,sum)
data.norm <- data/data.rowsum
testdata <- data.norm
dimnames(testdata) <- list(paste("Sample",seq(1,10)),paste("Feature",seq(1,4)))

nc.score.results <- nc.score( x=testdata )
nc.score.results.bins <- nc.score( x=testdata  )
nc.score.results.bin.cutoffs <- nc.score( x=testdata  )
nc.score.results
nc.score.results.bins
nc.score.results.bin.cutoffs
```

Index

ccrepe, 3
ccrepe-package, 2
ccrepeSampleTestFunction, 6

nc.score, 7