Package: ribiosMath (via r-universe)

October 26, 2026

Type Package

Title Mathematical and Statistical Tools in 'ribios'

Version 1.2.0

Date 2026-01-24

Description Mathematical and statistical tools for computational biology in drug discovery. Functions are designed for high performance. Implements the hierarchical fuzzy multi-linkage partitioning method proposed by Huang et al. (2007) <doi:10.1186/gb-2007-8-9-r183>.

Depends R (>= 3.4.0)

Imports methods, stats, grDevices, graphics, Rcpp

Suggests testthat

LinkingTo Rcpp, RcppArmadillo

License GPL-3

URL https://github.com/bedapub/ribiosMath

BugReports https://github.com/bedapub/ribiosMath/issues

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation yes

Author Jitao David Zhang [aut, cre, ctb] (ORCID: <https://orcid.org/0000-0002-3085-0909>)

Maintainer Jitao David Zhang <jitao_david.zhang@roche.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-02-17 22:00:02 UTC

RemoteUrl https://github.com/cran/ribiosMath

RemoteRef HEAD

RemoteSha 0cb3a53d577f99f8b36a71fbdc7ff23139a656bf

Contents

colKappa 2
cosdist 3
cossim 4
davidClustering_kappa 5
davidClustering_kappa_R 7
empval 8
kappaSimp 9
rowKappa 10
simulate_from_density 11
tfidf 11

Index **13**

`colKappa` *Calculate column-wise kappa statistics of a matrix*

Description

The function returns column-wise kappa statistics of a matrix, using a linear algebra procedure implemented in C++.

Usage

```
colKappa(matrix, minOverlap = 0L)
```

Arguments

`matrix`	An adjacency matrix, containing values of either 0 or 1 (default), or values between 0 and 1 (weighted).
`minOverlap`	Integer, minimal overlap between two columns in order to be considered. Pairs with fewer overlaps will return `NA`.

Value

A matrix of size $n \times n$ if the input matrix is of size $m \times n$.

Note

A kappa statistics of value 1 indicates perfect agreement. A value of 0 indicates no agreement. Note that the value can be negative, which implies the agreement is worse than random.

See Also

`rowKappa` to calculate the statistic of rows

Other kappa functions: `kappaSimp()`, `rowKappa()`

Examples

```
testMat <- cbind(c(1,1,0,0,1,0), c(1,1,0,1,1,0))
colKappa(testMat)
```

`cosdist` *Calculate the cosine distance between two vectors (matrices)*

Description

Calculate the cosine distance between two vectors (matrices)

Usage

```
cosdist(x, y, na.rm = TRUE)
```

Arguments

`x`	An integer or numeric vector or matrix
`y`	An integer or numeric vector or matrix
`na.rm`	Logical, whether `NA` should be removed

Details

Cossine distance is defined by $1 - cossim$, where $cossim$ represents the cosine similarity.

If parameters are given as matrices, the function calculates the cossine distance between all pair of *columns* of both matrices.

Value

Numeric vector or matrix, the cossine similarity between the inputs

Note

Currently, `na.rm` is only considered when both inputs are vectors

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

https://en.wikipedia.org/wiki/Cosine_similarity

See Also

`cor`, `cossim`

Examples

```
testVal1 <- rnorm(10)
testVal2 <- rnorm(10)
testVal3 <- c(rnorm(9), NA)

cosdist(testVal1, testVal2)
cosdist(testVal1, testVal3, na.rm=TRUE)
cosdist(testVal1, testVal3, na.rm=FALSE)
## test matrix
testMat1 <- matrix(rnorm(1000), nrow=10)
testMat2 <- matrix(rnorm(1000), nrow=10)

testVecMatDist1 <- cosdist(testMat1[,1L], testMat2)
testVecMatDist <- cosdist(testMat1, testMat2)
```

`cossim` *Calculate the cosine similarity between two vectors (matrices)*

Description

Calculate the cosine similarity between two vectors (matrices)

Usage

```
cossim(x, y, na.rm = TRUE)
```

Arguments

`x`	An integer or numeric vector or matrix
`y`	An integer or numeric vector or matrix
`na.rm`	Logical, whether `NA` should be removed

Details

If given as vectors, `x` and `y` must be of the same length. If given as matrices, both must have the same number of rows. If given as a pair of matrix and vector, the length of the vector must match the row number of the matrix. Otherwise the function aborts and prints error message.

If parameters are given as matrices, the function calculates the cossine similarity between all pair of *columns* of both matrices.

If `na.rm` is set `FALSE`, any `NA` in the input vectors will cause the result to be `NA`, or `NaN` if all values turn out to be `NA`.

Value

Numeric vector or matrix, the cossine similarity between the inputs

Note

Currently, `na.rm` is only considered when both inputs are vectors

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

https://en.wikipedia.org/wiki/Cosine_similarity

See Also

cor, cosdist

Examples

```
testVal1 <- rnorm(10)
testVal2 <- rnorm(10)
testVal3 <- c(rnorm(9), NA)

cossim(testVal1, testVal2)
cossim(testVal1, testVal3, na.rm=TRUE)
cossim(testVal1, testVal3, na.rm=FALSE)

cosdist(testVal1, testVal2)
cosdist(testVal1, testVal3, na.rm=TRUE)
cosdist(testVal1, testVal3, na.rm=FALSE)
## test matrix
testMat1 <- matrix(rnorm(1000), nrow=10)
testMat2 <- matrix(rnorm(1000), nrow=10)
system.time(testMatCos <- cossim(testMat1, testMat2))

testMatVec <- cossim(testMat1, testMat2[,1L])
testVecMat <- cossim(testMat1[,1L], testMat2)
```

---

davidClustering_kappa *Cluster rows of a Kappa-statistic-matrix by the hierarhical fuzzy multi-linkage partitioning method proposed by DAVID*

Description

The function implements the Hierarhical fuzzy multi-linkage partitioning method used in the DAVID Bioinformatics tool.

Usage

```
davidClustering_kappa(
  kappaMatrix,
  kappaThr = 0.35,
  initialGroupMembership = 3L,
  multiLinkageThr = 0.5,
  mergeRule = 1L
)
```

Arguments

`kappaMatrix`	A numeric matrix of Kappa statistics, which is likely returned by `rowKappa` or `colKappa`
`kappaThr`	Numeric, the threshold of the Kappa statistic, which is used to select initial seeds. Default value: 0.35, as recommended by the authors of the original study based on their experiences.
`initialGroupMembership`	Non-negative integer, the number of minimal members in initial groups. Default value: 3.
`multiLinkageThr`	Numeric, the minimal linkage between two groups to be merged. Default value: 0.5.
`mergeRule`	Integer, how two seeds are merged. See below. Currently following merge rules are implemented: • 1 (OR RULE) length of intersect divided by length of *either* seeds no less than `multiLinkageThr`. Empirical evidence suggests that it is a bit coarse grain than the native DAVID clustering algorithm, but the performance is quite good judged by biological relevance. • 2 (AND RULE) length of intersect divided by length of *both* seeds no less than `multiLinkageThr`, which gives slightly fragmented cluster by empirical experieince • 3 (UNION RULE) length of intersect divided by length of the union no less than `multiLinkageThr`, which performs similar to the *AND RULE* above. • 4 (GMEAN RULE) Geometric mean of length of intersect divided by length of *both* seeds no less than `multiLinkageThr`, the clusters tend to be highly fragemented. • 5 (AMEAN RULE) Arithmetic mean of length of intersect divided by length of *both* seeds no less than `multiLinkageThr`, a few items tend to appear in multiple clusters.

Value

A list of integer vectors. Each element represents a cluster and contains the indices of rows belonging to that cluster. Rows can appear in multiple clusters (fuzzy clustering).

Note

The function has only been tested in a few anecdotal examples. Cautions and more systematic tests are required before it is applied to critical datasets.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

Huang et al. The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biology, 2007. doi:10.1186/gb200789-r183

Examples

```
synData <- matrix(c(rep(c(rep(1, 10), rep(0, 5)), 3),
rep(0, 4), rep(1, 7), rep(0,4),
rep(c(rep(0,5), rep(1,10)), 3),
rep(c(rep(0,3), 1), 4)[-16]), ncol=15, byrow=TRUE)
rownames(synData) <- sprintf("Gene %s", letters[1:8])
colnames(synData) <- sprintf("t%d", 1:15)
synKappaMat <- rowKappa(synData)
synKappaMat.round2 <- round(synKappaMat, 2)
davidClustering_kappa(synKappaMat.round2)
```

---

davidClustering_kappa_R

Cluster rows of a Kappa-statistic-matrix by the hierarhical fuzzy multi-linkage partitioning method proposed by DAVID

Description

The function implements the Hierarhical fuzzy multi-linkage partitioning method used in the DAVID Bioinformatics tool.

Usage

```
davidClustering_kappa_R(
  kappaMatrix,
  kappaThr = 0.35,
  initialGroupMembership = 3,
  multiLinkageThr = 0.5,
  removeRedundant = TRUE,
  debug = FALSE
)
```

Arguments

kappaMatrix	A numeric matrix of Kappa statistics, which is likely returned by rowKappa or colKappa.
kappaThr	Numeric, the threshold of the Kappa statistic, which is used to select initial seeds. Default value: 0.35, as recommended by the authors of the original study based on their experiences.
initialGroupMembership	Integer, the number of minimal members in initial groups. Default value: 3.
multiLinkageThr	Numeric, the minimal linkage between two groups to be merged. Default value: 0.5.
removeRedundant	Logical, whether redundant initial groups should be removed before clustering. Used for debugging. Setting as TRUE accelerates the iterative clustring process.
debug	Logical, whether seed information is printed for debugging purposes.

Note

The function has only been tested in a few anecdotal examples. Cautions and more systematic tests are required before it is applied to critical datasets.

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

Huang et al. The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biology, 2007. doi:10.1186/gb200789-r183

---

empval *Get empirical p-value*

Description

Calculate empirical p-values from real values and simulated values

Usage

```
empval(stat, sim)
```

Arguments

stat	A numeric vector of calculated statistic from the actual data
sim	A numeric vector (or matrix) of simulated statistics, e.g. by Monte-Carlo methods.

Details

The estimate of the P-value is obtained as $\hat{p} = (r + 1)/(n + 1)$, where `n` is the number of replicate samples that have been simulated and `r` is the number of these replicates that produce a test statistic greater than or equal to that calculated for the actual data.

Value

A vector of empirical p-values, of the same length as the input

Author(s)

Jitao David Zhang <jitao_david.zhang@roche.com>

References

- Davison AC, Hinkley DV (1997) Bootstrap methods and their applications. Cambridge University Press, Cambridge, United Kindom.
- North BV, Curtis D, Sham PC (2002) A note on the calculation of empirical p values from Monte Carlo Procedures. Am J Hum Genet. 2002 August; 71(2):439–441.

Examples

```
set.seed(1995)
testStat <- c(-100, -3, -1, 0, 1, 3, 100)
testSim <- rnorm(1000)
empval(stat=testStat, sim=testSim)
```

---

kappaSimp — *Calculate column-wise kappa statistics of a matrix, using a simple procedure by going through the matrix and counting*

Description

Calculate column-wise kappa statistics of a matrix, using a simple procedure by going through the matrix and counting

Usage

```
kappaSimp(matrix, minOverlap = 0)
```

Arguments

`matrix`	a binary matrix of either 0 or one
`minOverlap`	Numeric/integer, the minimal overlap between two columns to be considered for further calculation

Value

A matrix of size nxn if the input matrix is of size mxn (m is arbitrary)

See Also

colKappa to calculate the same statistic using a linear algebra based routine

Other kappa functions: colKappa(), rowKappa()

---

rowKappa *Calculate row-wise kappa statistics of a matrix*

Description

The function returns row-wise kappa statistics of a matrix, using a linear algebra procedure implemented in C++.

Usage

```
rowKappa(matrix, minOverlap = 0L)
```

Arguments

matrix	An adjacency matrix, containing values of either 0 or 1.
minOverlap	Integer, minimal overlap between two columns in order to be considered. Pairs with fewer overlaps will return NA.

Value

A matrix of size $m \times m$ if the input matrix is of size $m \times n$.

Note

A kappa statistics of value 1 indicates perfect agreement. A value of 0 indicates no agreement. Note that the value can be negative, which implies the agreement is worse than random.

See Also

colKappa to calculate the statistic of rows

Other kappa functions: colKappa(), kappaSimp()

Examples

```
testMat <- cbind(c(1,1,0,0,1,0), c(1,1,0,1,1,0), c(0,1,0,0,1,0), c(1,0,1,0,1,0))
rowKappa(testMat)
stopifnot(identical(rowKappa(testMat), colKappa(t(testMat))))
```

simulate_from_density *Simulate from density*

Description

Compared with bootstrapping, the results do not reveal input values, and the empirical distribution can be smoother. The function assumes that the distribution can be aproximated using a gaussian kernel.

Usage

```
simulate_from_density(vec, N = 1e+05)
```

Arguments

vec	Numeric vector
N	Integer, number of simulated instances

Value

A numeric vector of length N with values simulated from the kernel density estimate of vec.

Author(s)

Iakov Davydov

Examples

```
my_vec <- c(23, 27, 26, 24, 25)
simulate_from_density(my_vec, 10)
```

tfidf *Calculate TF-IDF using a input matrix with terms in rows and documents in columns*

Description

Calculate TF-IDF using a input matrix with terms in rows and documents in columns

Usage

```
tfidf(
  tdMat,
  tfVariant = c("raw", "binary", "frequency", "log", "doubleNorm0.5"),
  idfVariant = c("raw", "smooth", "probabilistic"),
  idfAddOne = TRUE
)
```

Arguments

tdMat	A term-document matrix, terms in rows, documents in columns, and counts as integers (or logical values) in cells
tfVariant	Variant of term frequency. See details below.
idfVariant	Variant of inverse document frequency. See details below.
idfAddOne	Logical, whether one should be added to both numerator and denominator to calculate IDF. See details below.

Details

`tfVariant` accepts following options:

raw The input matrix is used as it is.

binary The input matrix is transformed into logical values.

frequency Term frequency per document is calculated from the input matrix.

log Transformation `log(1+tfMat)`

doubleNorm0.5 Double normalisation 0.5

`idfVariant` accepts following options:

raw `log(N/Nt)`

smooth `log(1+N/Nt)`

probabilistic `log((N-nt)/nt)`

, where `N` represents the total number of documents in the corpus, and `nt` is the number of documents where the term `t` appears. If `idfAddOne` is set `TRUE`, both numbers with addition of 1 to prevent division-by-zero.

Value

A numeric matrix of the same dimensions as `tdMat`, containing the TF-IDF values.

References

The Wikipedia item on TF-IDF: https://en.wikipedia.org/wiki/Tf%E2%80%93idf.

Examples

```
tiExample <- matrix(c(1,1,1,1,1,
1,1,0,0,0,
1,0,0,0,0,
0,1,0,0,0,
0,0,0,1,0,
1,0,1,0,1,
0,0,0,0,1), ncol=5, byrow=TRUE)
colnames(tiExample) <- sprintf("D%d", 1:ncol(tiExample))
rownames(tiExample) <- sprintf("t%d", 1:nrow(tiExample))
tiRes <- tfidf(tiExample)
```

Index

∗ **kappa functions**
colKappa, 2
kappaSimp, 9
rowKappa, 10

colKappa, 2, *6*, *8*, *10*
cor, *3*, *5*
cosdist, 3, *5*
cossim, *3*, 4

davidClustering_kappa, 5
davidClustering_kappa_R, 7

empval, 8

kappaSimp, *2*, 9, *10*

rowKappa, *2*, *6*, *8*, 10, *10*

simulate_from_density, 11

tfidf, 11