

Package ‘LISTO’

March 6, 2026

Type Package

Title Performing Comprehensive Overlap Assessments

Version 0.6.5

Description The implementation of a statistical framework for performing overlap assessments on lists comprising sets of strings (such as lists of gene sets) described in Stoica (2023) <<https://ora.ox.ac.uk/objects/uuid:b0847284-a02f-47ee-88e3-a3c4e0cdb8b1>>. It can assess overlaps of pair of sets of strings selected from the same universe or from different universes, and overlaps of triplets of sets of strings selected from the same universe. Designed for single-cell RNA-sequencing data analysis applications, but suitable for other purposes as well.

License MIT + file LICENSE

Imports methods, parallel, primes, satisfactory, stats

Encoding UTF-8

RoxygenNote 7.3.3

Suggests qs2, scRNAseq, scuttle, Seurat, testthat (>= 3.0.0), withr

URL <https://github.com/andrei-stoica26/LISTO>

BugReports <https://github.com/andrei-stoica26/LISTO/issues>

Config/testthat/edition 3

NeedsCompilation no

Author Andrei-Florian Stoica [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5253-0826>>)

Maintainer Andrei-Florian Stoica <andreistoica@foxmail.com>

Repository CRAN

Date/Publication 2026-03-06 18:10:12 UTC

Contents

buildSeuratMarkerList	2
factorialPrimePowers	3
mtCorrectDF	3

mtCorrectV	4
probCounts2MN	5
probCounts3N	6
pvalCounts2MN	6
pvalCounts3N	7
pvalObjects	8
pvalSets2MN	9
pvalSets2N	10
pvalSets3N	10
runLISTO	11
vChoose	12
vSum	13

Index	14
--------------	-----------

buildSeuratMarkerList *Build a Seurat marker list ready to be used by LISTO*

Description

This function builds a Seurat marker list ready to be used by LISTO. Requires Seurat (not automatically installed with LISTO).

Usage

```
buildSeuratMarkerList(seuratObj, col, logFCThr = 1, minPct = 0.2, ...)
```

Arguments

seuratObj	A Seurat object.
col	Seurat metadata column used for grouping.
logFCThr	Fold change threshold for testing.
minPct	The minimum fraction of in-cluster cells in which tested genes need to be expressed.
...	Additional arguments passed to <code>Seurat::FindMarkers</code> .

Value

A list consisting of data frames generated with `Seurat::FindMarkers`.

Examples

```
seuratPath <- system.file('extdata', 'seuratObj.qs2', package='LISTO')
seuratObj <- qs2::qs_read(seuratPath)
a <- buildSeuratMarkerList(seuratObj, 'Cell_Cycle', logFCThr=0.1)
```

factorialPrimePowers *Generate the prime factor decomposition of n factorial.*

Description

This function generates the prime factor decomposition of n factorial.

Usage

```
factorialPrimePowers(n)
```

Arguments

n A positive integer.

Value

A vector in which positions represent prime numbers (that is, the first position corresponds to 2, the second position corresponds to 3, the third position corresponds to 5, etc.) and values represent their exponents in the factorial decomposition.

Examples

```
factorialPrimePowers(8)
```

mtCorrectDF *Perform multiple testing correction on a data frame*

Description

This function orders a data frame based on a column of p-values, performs multiple testing correction on the column, and filters the data-frame based on the adjusted p-values.

Usage

```
mtCorrectDF(  
  df,  
  mtMethod = c("BY", "holm", "hochberg", "hommel", "bonferroni", "BH", "fdr", "none"),  
  colStr = "pval",  
  newColStr = "pvalAdj",  
  doOrder = TRUE,  
  doFilter = TRUE,  
  pvalThr = 0.05,  
  ...  
)
```

Arguments

df	A data frame with a p-values columnn.
mtMethod	Multiple testing correction method. Choices are 'BY' (default) 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'fdr' and 'none'.
colStr	Name of the column of p-values.
newColStr	Name of the column of adjusted p-values that will be created.
doOrder	Whether to increasingly order the data frame based on the adjusted p-values.
doFilter	Whether to filter the data frame based on the adjusted p-values.
pvalThr	p-value threshold used for filtering. Ignored if doFilter is FALSE.
...	Additional arguments passed to the multiple testing correction method.

Value

A data frame in which the p-value column has been corrected for multiple testing.

Examples

```
df <- data.frame(elem = c('A', 'B', 'C', 'D', 'E'),
  pval = c(0.032, 0.001, 0.0045, 0.051, 0.048))
mtCorrectDF(df)
```

mtCorrectV

Perform multiple testing correction on a vector of p-values

Description

This function performs multiple testing correction on a vector of p-values.

Usage

```
mtCorrectV(
  pvals,
  mtMethod = c("BY", "holm", "hochberg", "hommel", "bonferroni", "BH", "fdr", "none"),
  mtStat = c("identity", "median", "mean", "max", "min"),
  nComp = length(pvals)
)
```

Arguments

pvals	A numeric vector.
mtMethod	Multiple testing correction method. Choices are 'BY' (default) 'holm', 'hochberg', 'hommel', 'bonferroni', 'BH', 'fdr' and 'none'.
mtStat	A statistics to be optionally computed. Choices are 'identity' (no statistics will be computed and the adjusted p-values will be returned as such), 'median', 'mean', 'max' and 'min'.
nComp	Number of comparisons. In most situations, this parameter should not be changed.

Value

If `mtStat` is 'identity' (as default), a numeric vector of p-values corrected for multiple testing. Otherwise, a statistic based on these corrected p-values defined by `mtStat`.

Examples

```
pvals <- c(0.032, 0.001, 0.0045, 0.051, 0.048)
mtCorrectV(pvals)
```

probCounts2MN	<i>Compute the probability that two subsets of sets M and N intersect in k points</i>
---------------	---------------------------------------------------------------------------------------

Description

This function computes the probability that two subsets of sets M and N intersect in k points. Intersection sizes (M with N, A with N and B with M) must be provided.

Usage

```
probCounts2MN(intMN, intAN, intBM, k)
```

Arguments

<code>intMN</code>	Number of elements in the intersection of sets M and N.
<code>intAN</code>	Number of elements in the intersection of sets A (subset of M) and N.
<code>intBM</code>	Number of elements in the intersection of sets B (subset of N) and M.
<code>k</code>	Number of elements in the intersection of sets A and B.

Value

A numeric value in [0, 1] representing the probability that two subsets of sets M and N intersect in k points.

Examples

```
probCounts2MN(8, 6, 4, 2)
```

probCounts3N	<i>Compute the probability that three subsets of given sizes intersect in k points</i>
--------------	----------------------------------------------------------------------------------------

Description

This function computes the probability that three subsets of given sizes intersect in k points.

Usage

```
probCounts3N(a, b, c, n, k)
```

Arguments

a	Size of the first subset.
b	Size of the second subset.
c	Size of the third subset.
n	Size of the set.
k	Size of the intersection.

Value

A numeric value in $[0, 1]$ representing the probability that three subsets of given sizes intersect in k points.

Examples

```
probCounts3N(8, 6, 10, 20, 3)
```

pvalCounts2MN	<i>Compute the probability that two subsets of sets M and N intersect in at least k points</i>
---------------	------------------------------------------------------------------------------------------------

Description

This function computes the probability that two subsets A and B of sets M and N intersect in at least k points.

Usage

```
pvalCounts2MN(intMN, intAN, intBM, k)
```

Arguments

intMN	Number of elements in the intersection of sets M and N.
intAN	Number of elements in the intersection of sets A (subset of M) and N.
intBM	Number of elements in the intersection of sets B (subset of N) and M.
k	Number of elements in the intersection of sets A and B.

Value

A numeric value in [0, 1] representing the probability that two subsets of sets M and N intersect in at least k points.

Examples

```
pvalCounts2MN (300, 23, 24, 6)
```

pvalCounts3N	<i>Compute the probability that three subsets of a set intersect in at least k points</i>
--------------	-------------------------------------------------------------------------------------------

Description

This function computes the probability that three subsets of a set intersect in at least k points.

Usage

```
pvalCounts3N(lenA, lenB, lenC, n, k)
```

Arguments

lenA	Size of the first subset.
lenB	Size of the second subset.
lenC	Size of the third subset.
n	Size of the set comprising the subsets.
k	Size of the intersection.

Value

A numeric value in [0, 1] representing the probability that three subsets of a set intersect in at least k points.

Examples

```
pvalCounts3N (300, 200, 250, 400, 180)
```

pvalObjects

Assess the overlap of two or three objects

Description

This function assesses the overlap of two or three objects (character vectors, or data frames having a numeric column).

Usage

```
pvalObjects(
  obj1,
  obj2,
  obj3 = NULL,
  universe1,
  universe2 = NULL,
  numCol = NULL,
  isHighTop = TRUE,
  maxCutoffs = 5000,
  mtMethod = c("BY", "holm", "hochberg", "hommel", "bonferroni", "BH", "fdr", "none"),
  nCores = 1,
  type = c("2N", "2MN", "3N")
)
```

Arguments

obj1	A data frame with a numeric column, or a character vector.
obj2	A data frame with a numeric column, or a character vector.
obj3	A data frame with a numeric column, or a character vector.
universe1	The set from which the items stored in obj1 are selected.
universe2	The set from which the items stored in obj2 are selected.
numCol	The name of the numeric column used for data frame ordering.
isHighTop	Whether higher values in the numeric column correspond to top-ranked items.
maxCutoffs	Maximum number of cutoffs. If the input data frames contain more cutoffs than this value, only maxCutoffs linearly spaced cutoffs will be selected from the generated cutoff list.
mtMethod	Multiple testing correction method.
nCores	Number of cores. If performing an overlap assessment between sets belonging to the same universe, it is recommended not to use parallelization (that is, leave this parameter as 1).
type	Type of overlap assessment. Choose between: two sets belonging to the same universe ('2N'), two sets belonging to different universes ('2MN'), three sets belonging to the same universe ('3MN').

Value

A numeric value in [0, 1] representing the p-value of the overlap of the two objects.

Examples

```
pvalObjects(LETTERS[seq(2, 7)], LETTERS[seq(3, 19)], universe1=LETTERS)
```

pvalSets2MN

Compute the p-value of intersection of two subsets of sets M and N

Description

This function computes the p-value of intersection of two subsets of sets M and N.

Usage

```
pvalSets2MN(a, b, m, n)
```

Arguments

a	A character vector.
b	A character vector.
m	Set from which a is selected.
n	Set from which b is selected.

Details

A thin wrapper around pvalCounts2MN.

Value

A numeric value in [0, 1] representing the p-value of intersection of two subsets of sets M and N.

Examples

```
pvalSets2MN(LETTERS[seq(4, 10)],
LETTERS[seq(7, 15)],
LETTERS[seq(19)],
LETTERS[seq(6, 26)])
```

pvalSets2N *Calculate the p-value of intersection for two sets*

Description

This function calculates the p-value of intersection for two sets.

Usage

```
pvalSets2N(a, b, n)
```

Arguments

a	A character vector.
b	A character vector.
n	Set from which a and b are selected.

Details

A thin wrapper around `stats::phyper`.

Value

A numeric value in $[0, 1]$ representing the p-value of intersection for two sets.

Examples

```
pvalSets2N(LETTERS[seq(4, 10)], LETTERS[seq(7, 15)], LETTERS)
```

pvalSets3N *Compute the p-value of intersection of three subsets*

Description

This function computes the p-value of intersection of three subsets.

Usage

```
pvalSets3N(a, b, c, n)
```

Arguments

a	A character vector.
b	A character vector.
c	A character vector.
n	Set from which a, b and c are selected.

Details

A thin wrapper around pvalCounts3N.

Value

A numeric value in [0, 1] representing the p-value of intersection of three subsets.

Examples

```
pvalSets3N(LETTERS[seq(4, 10)],  
LETTERS[seq(7, 15)],  
LETTERS[seq(19)],  
LETTERS)
```

runLISTO

Assess the overlap of two or three lists of objects.

Description

This function assesses the overlap of two or three lists of objects (character vectors, or data frames having at least one numeric column).

Usage

```
runLISTO(  
  list1,  
  list2,  
  list3 = NULL,  
  universe1,  
  universe2 = NULL,  
  numCol = NULL,  
  isHighTop = TRUE,  
  maxCutoffs = 5000,  
  mtMethod = c("BY", "holm", "hochberg", "hommel", "bonferroni", "BH", "fdr", "none"),  
  filterResults = FALSE,  
  nCores = 1,  
  verbose = TRUE,  
  ...  
)
```

Arguments

list1	A list containing character vectors, or data frames having a numeric column.
list2	A list containing character vectors, or data frames having a numeric column.
list3	A list containing character vectors, or data frames having a numeric column.

universe1	Character vector; the set from which the items corresponding to the elements in list1 are selected.
universe2	Character vector; the set from which the items corresponding to the elements in list2 are selected.
numCol	The name of the numeric column used for data frame ordering.
isHighTop	Whether higher values in the numeric column correspond to top-ranked items.
maxCutoffs	Maximum number of cutoffs. If the input data frames contain more cutoffs than this value, only maxCutoffs linearly spaced cutoffs will be selected from the generated cutoff list.
mtMethod	Multiple testing correction method.
filterResults	Logical; whether to filter the results based on the adjusted p-values.
nCores	Number of cores. If performing an overlap assessment between sets belonging to the same universe, it is recommended not to use parallelization (that is, leave this parameter as 1).
verbose	Logical; whether the output should be verbose.
...	Additional arguments passed to mtCorrectDF.

Value

A data frame listing the p-value and adjusted p-value for each overlap. Combinations of overlaps are represented through the first two (or three if list3 is not NULL) columns, while the penultimate column records the overlap p-values and the last column records the adjusted overlap p-values.

Examples

```
donorPath <- system.file('extdata', 'donorMarkers.qs2', package='LISTO')
donorMarkers <- qs2::qs_read(donorPath)[seq(3)]
labelPath <- system.file('extdata', 'labelMarkers.qs2', package='LISTO')
labelMarkers <- qs2::qs_read(labelPath)[seq(3)]
universe1Path <- system.file('extdata', 'universe1.qs2', package='LISTO')
universe1 <- qs2::qs_read(universe1Path)
res <- runLISTO(donorMarkers, labelMarkers, universe1=universe1,
numCol='avg_log2FC')
```

vChoose

Compute the prime factor decomposition of the binomial coefficient

Description

This function computes the prime factor decomposition of the binomial coefficient.

Usage

```
vChoose(n, k)
```

Arguments

n	Total number of elements.
k	Number of selected elements.

Value

A vector in which positions represent prime numbers (that is, the first position corresponds to 2, the second position corresponds to 3, the third position corresponds to 5, etc.) and values represent their exponents in the factorial decomposition.

Examples

```
vChoose(8, 4)
```

vSum	<i>Add numeric vectors of different lengths</i>
------	-------------------------------------------------

Description

This function adds numeric vectors of different lengths by filling shorter vectors with zeroes.

Usage

```
vSum(...)
```

Arguments

... Numeric vectors.

Value

A numeric vector.

Examples

```
vSum(c(1, 4), c(2, 8, 6), c(1, 7), c(10, 4, 6, 7))
```

Index

buildSeuratMarkerList, 2

factorialPrimePowers, 3

mtCorrectDF, 3

mtCorrectV, 4

probCounts2MN, 5

probCounts3N, 6

pvalCounts2MN, 6

pvalCounts3N, 7

pvalObjects, 8

pvalSets2MN, 9

pvalSets2N, 10

pvalSets3N, 10

runLISTO, 11

vChoose, 12

vSum, 13