Package ‘mHG’

July 22, 2025

Type Package

Title Minimum-Hypergeometric Test

Version 1.1

Date 2017-08-18

Author Kobi Perl

Maintainer Kobi Perl <kobipe3@gmail.com>

Description Runs a minimum-hypergeometric (mHG) test as described in: Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa.

License GPL-2

Depends methods

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2017-08-18 07:57:10 UTC

Contents

mHG-package *Minimum-Hypergeometric Test*

Description

Sometimes when running a hypergeometric test to check for enrichment for a feature in a group versus the background, the separation between the group and the background is done arbitrarily by setting a threshold on some other property. When the correct threshold is unknown, different thresholds can be tried, and the minimal p-value of the hypergeometric tests can be retreived. If the elements can be sorted according to the property, it is possible to perform the hypergeometric tests on groups of increasing size. The minimum over all the tests is the minimum hypergeometric statistic, or mHG. The mHG is not a p-value by itself, as multiple tests were performed, without correcting for this. The package provides means to calculate the statistic (mHG.statistic.calc), to fix the p-value (mHG.pval.calc) or to perform the entire test at once (mHG.test). This is an R implementation of the algorithm described in:
Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf

Details

Package:	mHG
Type:	Package
Version:	1.0
Date:	2015-05-18
License:	GPL-2
Depends:	methods

The package provides means to calculate the statistic (`mHG.statistic.calc`), to fix the p-value (`mHG.pval.calc`) or to perform the entire test at once (`mHG.test`).

Author(s)

Kobi Perl <kobipe3@gmail.com>

References

Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf

See Also

`mHG.statistic.calc` `mHG.pval.calc` `mHG.test`

Examples

```
N <- 50
B <- 15
lambdas <- numeric(50)
lambdas[sample(N, B)] <- 1
t <- mHG.test(lambdas)
t <- mHG.test(lambdas, n_max = 20)
```

`mHG.pval.calc` *Calculate the p-value of a minimum-hypergeometric (mHG) statistic.*

Description

Calculates the p-value associated with the (minimum-hypergeometric) mHG statistic.

Usage

```
mHG.pval.calc(p, N, B, n_max = N)
```

Arguments

`p` the mHG statistic. It is marked as p as it represents an "uncorrected" p-value.

`N` total number of white and black balls (according to the hypergeometric problem definition).

`B` number of black balls.

`n_max` the algorithm will calculate the p-value under the assumption that only the first n_{max} partitions were taken into account in determining the mHG statistic.

Details

$O(NB)$ running time, $O(NB)$ space.

Value

the p-value of the test.

Author(s)

Kobi Perl

References

Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf (pages 11-12, 19-20)

Examples

```
N <- 50
B <- 15
lambdas <- numeric(50)
lambdas[sample(N, B)] <- 1
p <- mHG.statistic.calc(lambdas)@mHG
p.corrected <- mHG.pval.calc(p, N, B) # Could have used mHG.test directly
```

mHG.statistic.calc *Calculate the minimum-hypergeometric (mHG) statistic.*

Description

Calculates the minimum-hypergeometric (mHG) statistic.
mHG definition: $mHG(\lambda) = min_{1<=n<N} HGT(b_n(\lambda))$
Where HGT is the hypergeometric tail: $HGT(b; N, B, n) = Probability(X >= b)$,
and $b_n = \sum_{i=1}^{n} \lambda_i$.

Usage

```
mHG.statistic.calc(lambdas, n_max = length(lambdas))
```

Arguments

lambdas	$\{0, 1\}^N$, sorted from top to bottom.
n_max	the algorithm will only consider the first n_{max} partitions.

Details

$O(n_{max} + B^2 * log(B))$ running time, $O(B)$ space.

Value

Instance of the class mHG.statistic.info (stores the statistics, and for which n and b_n it was obtained). If several n give the same mHG, the smallest one is chosen.

Author(s)

Kobi Perl

References

Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf (pages 10-11, 18-19)

See Also

mHG.statistic.info-class

Examples

```
N <- 50
B <- 15
lambdas <- numeric(50)
lambdas[sample(N, B)] <- 1
mHG.statistic.info <- mHG.statistic.calc(lambdas)@mHG
```

---

mHG.statistic.info-class

Class "mHG.statistic.info"

Description

Summarizes data about the minimum-hypergeometric (mHG) statistic of a {0,1}^N vector.

Objects from the Class

Objects can be created by calls of the form `new("mHG.statistic.info", ...)`.

Slots

`mHG`**:** The actual statistic.

`n`**:** The index in which the minimum was obtained.

`b`**:** $\sum_{i=1}^{n} \lambda_i$.

Methods

No methods defined with class "mHG.statistic.info" in the signature.

Author(s)

Kobi Perl

References

Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf (page 10)

See Also

`mHG.statistic.calc`

Examples

```
showClass("mHG.statistic.info")
```

mHG.test *Perform Minimum-Hypergeometric test.*

Description

Performs a minimum-hypergeometric (mHG) test. The null-hypothesis is that provided list was randomly and equiprobable selected from all lists containing N entries, B of which are 1s. The alternative hypothesis is that the 1s tend to appear at the top of the list.

Usage

```
mHG.test(lambdas, n_max = length(lambdas))
```

Arguments

`lambdas`	$\{0,1\}^N$, sorted from top to bottom.
`n_max`	the algorithm will only consider the first n_{max} partitions.

Details

$O(NB)$ running time, $O(NB)$ space.

Value

A list with class "htest" containing the following components:

`statistic`	The mHG statistic.
`p.value`	The p-value for the test.
`parameters`	• N - total number of white and black balls. • B - number of black balls. • n_max - Max partition considered by the algorithm.
`n`	The index for which the mHG was obtained (smallest one if several n give the same mHG).
`b`	$\sum_{i=1}^{n} \lambda_i$.

Author(s)

Kobi Perl

References

Eden, E. (2007). Discovering Motifs in Ranked Lists of DNA Sequences. Haifa. Retrieved from http://bioinfo.cs.technion.ac.il/people/zohar/thesis/eran.pdf (pages 10-12, 18-20)

Examples

```
N <- 50
B <- 15
lambdas <- numeric(50)
lambdas[sample(N, B)] <- 1
t <- mHG.test(lambdas)
t <- mHG.test(lambdas, n_max = 20)
```

Index