Package ‘EmpiricalBrownsMethod’

April 8, 2024

Title Uses Brown's method to combine p-values from dependent tests
Version 1.30.0
Author William Poole
Maintainer David Gibbs <dgibbs@systemsbiology.org>
Description Combining P-values from multiple statistical tests is common in bioinformatics. However, this procedure is non-trivial for dependent P-values. This package implements an empirical adaptation of Brown’s Method (an extension of Fisher’s Method) for combining dependent P-values which is appropriate for highly correlated data sets found in high-throughput biological experiments.
Depends R (>= 3.2.0)
Suggests BiocStyle, testthat, knitr, rmarkdown
License MIT + file LICENSE
VignetteBuilder knitr
URL https://github.com/IlyaLab/CombiningDependentPvaluesUsingEBM.git
LazyData true
Encoding UTF-8
biocViews StatisticalMethod, GeneExpression, Pathways
git_url https://git.bioconductor.org/packages/EmpiricalBrownsMethod
git_branch RELEASE_3_18
git_last_commit 7d9465f
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-04-08

\textbf{R topics documented:}

- ebmTestData .............................................. 2
- empiricalBrownsMethod .............................. 2
- kostsMethod ............................................. 3

Index 5
Data used in tests and examples.

ebmTestData

Description
This data is used in the unit tests and usage examples. There are four items:
allPvals, dat, pathways, and randData. allPvals is a data.frame of p-values for the spearman correlation between CHD4 and each of the 45 genes.
dat is the gene expression data corresponding to genes in allPvals.
pathways is a data.frame listing gene membership for 3 biochemical pathways.
randData is a gaussian generated data set, emphasizing dependence among variables. Independent Var [line 1] are 25 samples from a unit normal distribution. Dependent Var 1-10 [line 2-11] are each 25 samples drawn from a 10 dimensional normal distribution centered at the origin with off diagonal terms a=0.25. The P values from a pearson correlation between the independent var and each dependent var are combined.

Usage
data(ebmTestData)

Format
Rdata object

Value
data objects in the environment

Source
GEO and generated.

empiricalBrownsMethod

Description
Combining P-values from multiple statistical tests is common in bioinformatics. However, this procedure is non-trivial for dependent P-values. This package provides an empirical adaptation of Brown’s Method (an extension of Fisher’s Method) for combining dependent P-values which is appropriate for highly correlated data sets, like those found in high-throughput biological experiments.

Usage
empiricalBrownsMethod(data_matrix, p_values, extra_info)
**kostsMethod**

The Kost Method For Combining P-values

**Description**

Combining P-values from multiple statistical tests is common in bioinformatics. However, this procedure is non-trivial for dependent P-values. This package provides an implementation of Kost’s Method for combining dependent P-values which is appropriate for highly correlated data sets, like those found in high-throughput biological experiments.

**Usage**

\[
\text{kostsMethod}(\text{data\_matrix}, \text{p\_values}, \text{extra\_info})
\]

**Arguments**

- `data_matrix` An m x n numeric matrix with m variables in rows and n samples in columns.
- `p_values` A numeric vector of p-values with length m.
- `extra_info` boolean, TRUE additionally returns the p-value from Fisher’s method, the scale factor c, and the new degrees of freedom from Brown’s Method

**Value**

The output is a list containing:

- `P_Brown` p-value for Brown’s method
- `P_Fisher` p-value for Fisher’s method
- `Scale_Factor` the scale factor c
- `DF` the degrees of freedom used in Brown’s method

**Examples**

```r
## restore the saved values to the current environment
data(ebmTestData)
glypGenes <- pathways$gene[pathways$pathway == "GLYPICAN 3 NETWORK"]
glypPvals <- allPvals$pvalue.with.CHD4[match(glypGenes, allPvals$gene)];
glypDat <- dat[match(glypGenes, dat$V1), 2:ncol(dat)];
empiricalBrownsMethod(data_matrix=glypDat, p_values=glypPvals, extra_info=TRUE);
```
Value

The output is a list containing list(P_test=p_brown, P_Fisher=p_fisher, Scale_Factor_C=c, DF=df)

- **P_test**: p-value for Kost's method
- **P_Fisher**: p-value for Fisher's method
- **Scale_Factor**: the scale factor c
- **DF**: the degrees of freedom

Examples

```r
## restore the saved values to the current environment
data(ebmTestData)
glypGenes <- pathways$gene[pathways$pathway == "GLYPICAN 3 NETWORK"]
glypPvals <- allPvals$pvalue.with.CHD4[match(glypGenes, allPvals$gene)]
glypDat <- as.matrix(dat[match(glypGenes, dat$V1), 2:ncol(dat)])
kostsMethod(data_matrix=glypDat, p_values=glypPvals, extra_info=TRUE);
```
Index

* datasets
  * ebmTestData, 2
* multivariate
  * empiricalBrownsMethod, 2
  * kostsMethod, 3

allPvals(ebmTestData), 2
dat(ebmTestData), 2
ebmTestData, 2
dat(ebmTestData), 2
ebmTestData, 2
dat(ebmTestData), 2

empiricalBrownsMethod, 2
kostsMethod, 3
pathways(ebmTestData), 2
randData(ebmTestData), 2