Package: confSAM (via r-universe)

September 4, 2024

Type Package

Title Estimates and Bounds for the False Discovery Proportion, by
Permutation

Version 0.2

Date 2018-02-12

Author Jesse Hemerik and Jelle Goeman

Maintainer Jesse Hemerik <j.b.a.hemerik@lumc.nl>

Description For multiple testing. Computes estimates and confidence
bounds for the False Discovery Proportion (FDP), the fraction
of false positives among all rejected hypotheses. The methods
in the package use permutations of the data. Doing so, they
take into account the dependence structure in the data.

License GNU General Public License

LazyData TRUE

VignetteBuilder knitr

Suggests penalized, survival, knitr, markdown

NeedsCompilation no

Repository CRAN

Date/Publication 2018-02-19 10:39:59 UTC

Contents

| confSAM | *Permutation-based confidence bounds for the false discovery proportion* |

Description

Computes a confidence upper bound for the False Discovery Proportion (FDP). The input required is a matrix containing test statistics (or p-values) for (randomly) permuted versions of the data.

Usage

```
confSAM(p, PM, includes.id=TRUE, cutoff=0.01, reject="small", alpha=0.05,
        method="simple",  ncombs=1000)
```

Arguments

p	A vector containing the p-values for the original (unpermuted) data.
PM	A matrix (with `length(p)` columns) containing for each permutation the p-values corresponding to the permuted version of the data. If PM contains the original values p, then they should be in the first row of PM.
includes.id	Set this to `FALSE` if PM does not contain the original p-values p.
cutoff	A number or a vector of length `length(p)`. In the first case all hypotheses with test statistics exceeding `cutoff` are rejected. In the second case there is a separate cut-off for each hypothesis.
reject	If `reject="small"`, then all hypotheses with test statistics (p-values) smaller than `cutoff` are rejected. If `reject="larger"`, then all hypotheses with test statistics larger than `cutoff` are rejected. If `reject="absolute"`, then all hypotheses with test statistics with absolute value larger than `cutoff` are rejected.
alpha	1-alpha is the desired confidence level of the bounds.
method	If `method="simple"`, then a basic (fast) bound for V (the number of false positives) is computed. If `method="full"`, then a (computationally intensive) closed testing-based bound for V is computed. This is usually infeasible when the number of rejections is large. If `method="approx"`, then an approximation of the closed testing-based bound for V is computed. The resulting bound may be anti-conservative if ncombs is too small.
ncombs	Only applies when `method="approx"`. It is the number of random combinations that the approximation method checks. Larger values of ncombs give more reliable results.

Value

A vector with three values is returned. The first value is the number if rejections. The second value is a basic median unbiased estimate of the number of false positives V. This estimate coincides with the simple upper bound for alpha=0.5. The third value is a (1-alpha)-confidence upper bound for V (it depends on the argument `method` which bound this is.)

Examples

```
#This is a fast example. It is recommended to take w and ncombs larger in practice.
set.seed(423)
m <- 100   #number of hypotheses
n <- 10     #the amount of subjects is 2n (n cases, n controls).
w <- 50    #number of random permutations. Here we take w small for computational speed

X <- matrix(rnorm((2*n)*m), 2*n, m)
X[1:n,1:50] <- X[1:n,1:50]+1.5 # the first 50 hypotheses are false
#(increased mean for the first n individuals).

y <- c(numeric(n)+1,numeric(n)-1)
Y <- t(replicate(w, sample(y, size=2*n, replace=FALSE)))
Y[1,] <- y  #add identity permutation

pvalues <- matrix(nrow=w,ncol=m)
for(j in 1:w){
  for(i in 1:m){
    pvalues[j,i] <- t.test( X[Y[j,]==1,i], X[Y[j,]==-1,i] ,
                            alternative="two.sided" )$p.value
  }
}

## number of rejections:
confSAM(p=pvalues[1,], PM=pvalues, cutoff=0.05, alpha=0.1, method="simple")[1]

## basic median unbiased estimate of #false positives:
confSAM(p=pvalues[1,], PM=pvalues, cutoff=0.05, alpha=0.1, method="simple")[2]

## basic (1-alpha)-upper bound for #false positives:
confSAM(p=pvalues[1,], PM=pvalues, cutoff=0.05, alpha=0.1, method="simple")[3]

## potentially smaller (1-alpha)-upper bound for #false positives:
## (taking 'ncombs' much larger recommended)
confSAM(p=pvalues[1,], PM=pvalues, cutoff=0.05, alpha=0.1, method="approx",
        ncombs=50)[3]


## actual number of false positives:
sum(pvalues[1,51:100]<0.05)
```

Index

confSAM, 2