Package 'CorDiff'

September 17, 2018

Type Package

Imports mcc

Title Set-Based Differential Covariance Testing for Genomics

Version 1.0

Date 2018-09-05

Author Yi-Hui Zhou

Maintainer Yi-Hui Zhou <yihui_zhou@ncsu.edu>

Description We describe four different summary statistics, to ensure power and flexibility under various settings, including a new connectivity statistic that is sensitive to changes in overall covariance magnitude.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2018-09-17 13:10:10 UTC

R topics documented:

CorDiff-package *Set-based differential covariance testing for genomics*

Description

We describe four different summary statistics, to ensure power and flexibility under various settings. This is a uniform framework to test association of covariance matrices with an experimental variable, whether discrete or continuous. (1) A sumation statistic S which is to detect global changes in covariances that are concordantly associated with the experimental variable y; (2) A quadratic form statistic Q which is sensitive to changes that are not directionally concordant; (3) A connectivity statistic C which reflects the tendency for the aggregate magnitude of feature-feature correlations to be associated with y; (4) A maximum statistic M.

Author(s)

Yi-Hui Zhou

Maintainer: Yi-Hui Zhou <yihui_zhou@ncsu.edu>

References

Set-based differential covariance testing for genomics, Yi-Hui Zhou, under review

Examples

```
library(mcc)

n1=5
n2=5
y=c(rep(1/n1,n1),rep(-1/n2,n2))
data(x)
w=(colSums(x))^2
output=getbetap.A(getAmoment(rbind(y,y),w,z=NULL),A=NULL,fix.obs=TRUE)
S.p=output$twosidedp[1]

Qresult=Qresid(y,x,numperms=1e6,thresh=10)
Q.p=Qresult$myp

newx=(t(x)%*%x)^2
v=colSums(newx)
output2=getbetap.A(getAmoment(rbind(y,y),v,z=NULL),A=NULL,fix.obs=TRUE)
C.p=output2$twosidedp[1]
M.p=getMpfast(y,x,num.perms=1e4)$pval
```

fastresid *Residulize the effect of y away from x*

Description

This function is to prepare for the next Q calculation. Basically, Q does not like phenotype y to add complication. Therefore we use this function to get rid of the impact of y.

Usage

```
fastresid(X, y)
```

Arguments

X	The data matrix, each column is for each sample and each row is for different feature.
y	Experimental condition/phenotypes, it can be discrete or continuous

Value

Xresid	The new x after residulizing y

Author(s)

Yi-Hui Zhou

References

Set-based differential covariance testing for genomics

getMpfast *Calculate the statistic M*

Description

This function provides the permutation algorithm to calculate the maximum statistic M

Usage

```
getMpfast(y, x, num.perms = 1000)
```

Arguments

y	Experimental condition/phenotypes, it can be discrete or continuous
x	The data matrix, each column is for each sample and each row is for different feature.
num.perms	You can specifiy the number of permutation in the calculation. The default is 1000.

Value

Mobs	M statistic
pval	p value under permutation

Author(s)

Yi-Hui Zhou

References

Set-based differential covariance testing for genomics

Qresid *Calculate statistic Q.*

Description

For the purposes of computing type I error and power, we only need care about p-values that are smallish. If the pvalue is large, we do not care if it's 0.8 or 0.9. When we hit ratio=10, then our current pvalue is 10 standard deviations larger than zero, which is a safe crierion to stop and say we have enough permutations. Therefore we saved a ton of time.

Usage

```
Qresid(y, X, numperms = 10000, thresh = 10)
```

Arguments

y	Experimental condition/phenotypes, it can be discrete or continuous
X	The data matrix, each column is for each sample and each row is for different feature.
numperms	The number of permuations.
thresh	The threshold we set up to stop the permutation. The default value is 10 which comes from a 10 standard deviation criterion.

Value

myp	P value
i	The ith permutaion we stopped the algorithm.

Author(s)

Yi-Hui Zhou

References

Set-based differential covariance testing for genomics

---

x *The toy data example for input data matrix x in the example*

Description

This is the p by n toy data we used to illustrate the package.

Usage

```
data("x")
```

Details

The toy data is a p by n matrix, where n is the sample size

Author(s)

Yi-Hui Zhou

References

Set-based differential covariance testing for genomics

Examples

```
data(x)
dim(x)
```

Index