Package ‘CombinePValue’

February 19, 2015

Type Package
Title Combine a Vector of Correlated p-values
Version 1.0
Date 2014-11-02
Author Hongying Dai
Maintainer Hongying Dai <hdai@cmh.edu>
Description We offer two statistical tests to combine p-values: selfcontained.test vs competitive.test. The goal is to test whether a vector of pvalues are jointly significant when we combine them together.
License GPL-3
NeedsCompilation no
Repository CRAN
Date/Publication 2014-11-03 00:32:55

R topics documented:

CombinePValue-package .......................................................... 1
competitive.test ................................................................. 3
selfcontained.test .............................................................. 4

Index

CombinePValue-package  CombinePValue: combine a vector of correlated p-values

Description

We offer two statistical tests to combine p-values: selfcontained.test vs competitive.test. The goal is to test whether a vector of pvalues are jointly significant when we combine them together.

Advantage 1: The Lancaster procedure is a generalized Fisher’s method with weight functions to p-values.

Advantage 2: Correlations among p-values are modeled using Satterwite approximation (i.e. matching first two moments of chi-squares statistic.)
Details

Package: CombinePValue
Type: Package
Version: 1.0
Date: 2014-10-29
License: GPL-3

Parameters:
Pvalue: this is a vector. For instance, pvalue=c(0.01,0.04,0.06,0.15,0.17).
Weight: This is weight for p-values. Weight should be positive and weight is a vector with the same length as pvalue. For instance, weight=c(2,4,5,2,3).
If you do not have weight for p-values, simply set weight = NA. Default sets weight=2 for each p-value. This equals to Fisher’s method with correlation. Traditional Fisher’s method is based on independent assumptions. In this code, all tests (including Fisher’s method), take correlations into account.
In selfcontained.test, we are interested to test whether H0: pvalue is uniform vs. Ha: pvalue is not uniform.
p_permu: this is p-value matrix. One can use bootstrap method to randomly select samples or use permutation method to randomly shuffle phenotype (outcome variable) 200 times. Feel free to change 200 to other numbers. Bootstrapping of samples or permutation of phenotype allows us to generate samples of pvalue. Thus we can incorporate correlation among these pvalues in p_permu matrix. Every time after you perform bootstrap or permutation, you need to rerun your original analysis to generate a new vector of pvalue. Put pvalue vector in the column of p_permu. For instance, if you bootstrap samples or shuffle phenotype 200 times and pvalue is a vector of length 5, then p_permu is a matrix with 5 rows x 200 columns.
In competitive.test, we are interested to test whether pvalue is more significant than randomly selected p-values. p_random: this is p-value matrix by randomly selecting 100000 vectors of pvalues. Put randomly selected pvalue vectors in the columns of p_random.

Author(s)
Hongying Dai Maintainer: Hongying Dai <hdai@cmh.edu>

References

See Also
d.web.umkc.edu/hdai
selfcontained.test
competitive.test
Examples

# p_permu is generated by bootstrap or permutation to estimate correlations among p values.

a<-matrix(c(0.02,0.06,0.07,0.01,0.02,0.09,0.01,0.01,0.10,0.12,0.14,0.07,0.09,0.10,0.15),nrow=3)
selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=NA,p_permu=a)

#The traditional Fisher's test assuming p-values are independent.

selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=NA,p_permu=NA)

#Generalized Fisher method with a weight function. If p_permu=NA, then p-values are independent.

selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=c(7,2,1.5),p_permu=NA)

#Compare Pvalue vs randomly selected p values in columns of p_random.

b<-matrix(c(0.12,0.06,0.07,0.12,0.02,0.09,0.01,0.10,0.32,0.14,0.07,0.001,0.009,0.013),nrow=3)
competitive.test(pvalue=c(0.01,0.04,0.06),Weight=c(2,5,1),p_random=b)

Description

In this test, we are interested to test whether pvalue is more significant than randomly selected p-values.

Details

Package: CombinePValue
Type: Package
Version: 1.0
Date: 2014-10-29
License: GPL-3

Parameters:

Pvalue: this is a vector. For instance, pvalue=c(0.01,0.04,0.06,0.15,0.17).

Weight: This is weight for p-values. Weight should be positive and weight is a vector with the same length as pvalue. For instance, weight=c(2,4,5,2,3).

If you do not have weight for p-values, simply set weight = NA. Default sets weight=2 for each p-value. This equals to Fisher’s method with correlation. Traditional Fisher’s method is based on independent assumptions. In this code, all tests (including Fisher’s method), take correlations into account.
p_random: this is p-value matrix by randomly selecting 100000 vectors of pvalues. Put randomly selected pvalue vectors in the columns of p_random.

Author(s)

Hongying Dai

Maintainer: Hongying Dai <hdai@cmh.edu>

References


See Also

d.web.umkc.edu/hdai
CombinePValue
selfcontained.test

Examples

#Compare Pvalue vs randomly selected p values in columns of p_random.

b<-matrix(c(0.12,0.06,0.07,0.71,0.02,0.09,0.41,0.01,0.10,0.32,0.14,0.37,0.001,0.009,0.013),nrow=3)
competitive.test(Pvalue=c(0.01,0.04,0.06),Weight=c(2,5,1),p_random=b)

Parameters:
pvalue: this is a vector. For instance, pvalue=c(0.01,0.04,0.06,0.15,0.17).
weight: This is weight for p-values. Weight should be positive and weight is a vector with the same
length as pvalue. For instance, weight=c(2,4,5,2,3).

If you do not have weight for p-values, simply set weight = NA. Default sets weight=2 for each p-value. This equals to Fisher’s method with correlation. Traditional Fisher’s method is based on independent assumptions. In this code, all tests (including Fisher’s method), take correlations into account.

p_permu: this is p-value matrix. One can use bootstrap method to randomly select samples or use permutation method to randomly shuffle phenotype (outcome variable) 200 times. Feel free to change 200 to other numbers. Bootstrapping of samples or permutation of phenotype allows us to generate samples of pvalue. Thus we can incorporate correlation among these pvalues in p_permu matrix. Every time after you perform bootstrap or permutation, you need to rerun your original analysis to generate a new vector of pvalue. Put pvalue vector in the column of p_permu. For instance, if you bootstrap samples or shuffle phenotype 200 times and pvalue is a vector of length 5, then p_permu is a matrix with 5 rows x 200 columns.

Author(s)
Hongying Dai
Maintainer: Hongying Dai <hdai@cmh.edu>

References

See Also
d.web.umkc.edu/hdai
CombinePValue
competitive.test

Examples
# p_permu is generated by bootstrap or permutation to estimate correlations among p values.

ac<-matrix(c(0.02,0.06,0.07,0.01,0.02,0.09,0.01,0.01,0.10,0.12,0.14,0.07,0.09,0.10,0.15),nrow=3)
selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=NA,p_permu=ac)

#The traditional Fisher's test assuming p-values are independent.
selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=NA,p_permu=NA)

#Generalized Fisher method with a weight function. If p_permu=NA, then p-values are independent.
selfcontained.test(pvalue=c(0.01,0.04,0.06),weight=c(7,2,1.5),p_permu=NA)
Index

*Topic package
  CombinePValue-package, 1
  competitive.test, 3
  selfcontained.test, 4

CombinePValue (CombinePValue-package), 1
CombinePValue-package, 1
competitive.test, 3
selfcontained.test, 4