# Package 'MCPerm'

February 19, 2015

**Type** Package

**Title** A Monte Carlo permutation method for multiple test correlation

**Version** 1.1.4

**Date** 2013-06-17

**Author** Lanying Zhang and Yongshuai Jiang

**Maintainer** Yongshuai Jiang <jiangyongshuai@gmail.com>

**Description** A Monte Carlo permutation method for multiple test correlation.

**License** GPL-2

**Suggests** stats

**Depends** metafor, graphics

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2013-06-17 16:58:49

## R topics documented:

---

`MCPerm-package` *A Monte Carlo permutation method for multiple test correction.*

---

## Description

Permutation tests exist for any test statistic, regardless of whether or not its distribution is known. Thus the permutation test is widely considered the gold standard for accurate multiple testing correction.

For example,for case/control association study for SNPs, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the genotype frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

Obviously, for multiple test correction in case/control association study for millions of SNPs, the traditional method—permutation test is very computationally impractical. Thus propose an accurate, rapid and efficient method for multiple testing correction in genome-wide association studies—MCPerm.

Method—MCPerm generates the genotype frequency for rearranged case and control data by twice generating random numbers for the hypergeometric distribution, based on the genotype statistic of original data, taking the place of the step 2) and step 3) of the traditional method. And the genotype frequency distribution generating by MCPerm is almost the same with permutation test, this simplified method greatly improves the efficiency of the permutation test and is faster. MCPerm method can be the perfect alternative to permutation test.

## Details

| | |
|---|---|
| Package: | MCPerm |
| Type: | Package |
| Version: | 1.1.4 |
| Date: | 2013-06-12 |
| License: | GPL-2 |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does mutiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

Julian P.T.Higgins, Simon G.Thompson(Statistics in Medicine,2002): Quantifying heterogeneity in a meta-analysis.

## See Also

`Armitage`, `Armitage.TradPerm`, `Armitage.MCPerm`, `OR`, `OR.TradPerm`, `OR.MCPerm`, `permuteGenotype`, `permuteGenotypeCount`, `genotypeStat`, `chisq.TradPerm`, `chisq.MCPerm`, `fisher.TradPerm`, `fisher.MCPerm`, `rhyper`, `chisq.test`, `fisher.test`, `meta`, `meta.TradPerm`, `meta.MCPerm`, `VS.Genotype.Hist`, `VS.Allele.Hist`, `VS.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.boxplot`, `PermMeta.Hist`, `pearson_scatter`, `Q.TradPerm`, `Q.MCPerm`, `I2.TradPerm`, `I2.MCPerm`

## Examples

```
## example1-----genotypeStat-----------
```

```
##  import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
## Statistical allele and genotype frequency of the specified snp for case-control data.
# result=genotypeStat(data2,data1,fromCol=2,naString="?_?",sep="_")
# genotypeCount=result$genotypeCount
# alleleCount=result$alleleCount

## example2-----permuteGenotype-----------
## a matrix with 1 row
# dataLine=matrix(c("rs12","1","2","3","4","5"),nrow=1)
## permutate the elements of the matrix
# newData=permuteGenotype(dataLine=dataLine,fromCol=2)

## example3-----permuteGenotypeCount-----------
# newMatrix=permuteGenotypeCount(case_11=24,case_12=34,case_22=45,
#   control_11=23,control_12=45,control_22=34,n=5)

## example4-----OR-----------
## OR(odd ratio) for the risk-allele
# ORvalue=OR(case_allele1=20,case_allele2=30,control_allele1=10,control_allele2=60)

## example5-----OR.TradPerm--------
## import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# result1=OR.TradPerm(genotypeLine=data2,affectionLine=data1,fromCol=2,naString="?_?",
  # sep="_",repeatNum=5)
# risk_allele=result1$risk_allele
# p=result1$pValue
# obsOR=result$OR

## example6-----OR.MCPerm----------
# OR.MCPerm(case_allele1=34,case_allele2=23,control_allele1=27,control_allele2=45,repeatNum=5)

## example7-----Armitage-------
# Armitage(case_11=23,case_12=45,case_22=12,control_11=27,control_12=12,control_22=45)

## example8-----Armitage.TradPerm-----
## import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# Armitage.TradPerm(genotypeLine=data2,affectionLine=data1,
   # fromCol=2,naString="?_?",sep="_",repeatNum=1000)
```

```
## example9----Armitage.MCPerm--------
# Armitage.MCPerm(case_11=23,case_12=45,case_22=12,
   # control_11=27,control_12=12,control_22=45,repeatNum=1000)

## example10---chisq.TradPerm---------
## import example data
data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# chisq.TradPerm(genotypeLine=data2,affectionLine=data1,
   # fromCol=2,naString="?_?",sep="_",repeatNum=1000)

## example11---chisq.MCPerm--------
# case_11=23
# case_12=45
# case_22=12
# control_11=27
# control_12=12
# control_22=45
# chisq.MCPerm(23,45,12,27,12,45,repeatNum=5)

## example12---fisher.TradPerm------
# import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# fisher.TradPerm(genotypeLine=data2,affectionLine=data1,
   # fromCol=2,naString="?_?",sep="_",repeatNum=5)

## example13---fisher.MCPerm-------
#fisher.MCPerm(23,45,12,27,12,45,repeatNum=5)
```

---

`Armitage` *Armitage's trend test for the 2x3 genotype table*

---

## Description

Armitage's trend test for the 2x3 genotype table.

## Usage

```
Armitage(case_11, case_12, case_22, control_11, control_12, control_22)
```

## Arguments

| | |
|---|---|
| case_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| case_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| case_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| control_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| control_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| control_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |

## Details

The Cochran-Armitage test for trend, is used in categorical data analysis when the aim is to assess for the presence of an association between a variable with two categories and a variable with k categories. It modifies the Pearson chi-squared test to incorporate a suspected ordering in the effects of the k categories of the second variable. The trend test is ofen used as a genotype-based test for case-control genetic association studies.

## Value

| | |
|---|---|
| statistic | numeric, the statistic of armitage test for trend. |
| pValue | numeric, the p value of armitage test for trend. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Armitage, P(1955): Tests for Linear Trends in Proportions and Frequencies.

statgen.org(2007): A derivation for Armitage's trend test for the 2x3 genotype table.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
# case_11=4
# case_12=1
# case_22=1
# control_11=3
# control_12=0
# control_22=0
# Armitage(case_11,case_12,case_22,control_11,control_12,control_22)
```

---

`Armitage.MCPerm` *A Monte Carlo permutation method for Armitage's trend test in case/control association study*

---

## Description

A Monte Carlo permutation method for Armitage's trend test in case/control association study.

## Usage

```
Armitage.MCPerm(case_11, case_12, case_22, control_11, control_12, control_22,
    repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| `case_11` | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| `case_12` | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| `case_22` | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| `control_11` | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| `control_12` | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| `control_22` | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples |
| `repeatNum` | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

The Cochran-Armitage test for trend, is used in categorical data analysis when the aim is to assess for the presence of an association between a variable with two categories and a variable with k categories. It modifies the Pearson chi-squared test to incorporate a suspected ordering in the effects of the k categories of the second variable. The trend test is ofen used as a genotype-based test for case/control genetic association studies.

"Armitage.MCPerm" simulates permutation method to correct p value, by twice generating random numbers for the hypergeometric distribution based on the genotype statistic of original data. Seealso chisq.MCPerm.

## Value

| | |
|---|---|
| pValue | the p value for the test. |
| obsStatistic | the statistic of Armitage's trend test for the true data. |
| obsP | the p value for Armitage's trend test of the true data. |
| permStatistic | a vector with 'repeatNum' elements, the statistic of Armitage's trend test for the simulation data. |
| permP | a vector with 'repeatNum' elements, the p value for Armitage's trend test for the simulation data. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

statgen.org(2007): A derivation for Armitage's trend test for the 2x3 genotype table.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
## Armitage.MCPerm(case_11=4,case_12=1,case_22=1,control_11=3,
  # control_12=5,control_22=7,repeatNum=10000)
```

---

# Armitage.TradPerm *A permutation method for Armitage's trend test in case/control association study*

---

## Description

A permutation method for Armitage's trend test in case/control association study.

## Usage

```
Armitage.TradPerm(genotypeLine, affectionLine, fromCol, naString, sep, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| genotypeLine | a matrix with one row containing information of specified snp: basic information(e.g. SNP ID number,chromo,position) and genotype of observed individuals. See below for details. |
| affectionLine | a matrix having the same dimension with parameter 'genotypeLine' contain the affection status (case or control) of each individual and other information. The affection status must be in same columns with the genotypes in parameter 'genotypeLine'. See below for details. |
| fromCol | a positive integer, the start column of genotype data in parameter 'genotypeLine'. |
| naString | a character string for NA values of genotype. |
| sep | character separator used to divide genotype between alleles "Allele1<sep>Allele2". |
| repeatNum | an integer(default 1000) specifying the number of replicates for permutation test. |

## Details

The Cochran-Armitage test for trend, is used in categorical data analysis when the aim is to assess for the presence of an association between a variable with two categories and a variable with k categories. It modifies the Pearson chi-squared test to incorporate a suspected ordering in the effects of the k categories of the second variable. The trend test is ofen used as a genotype-based test for case/control genetic association studies.

'Armitage.TradPerm' uses permutation test for multiple Armitage's trend test correction. Seealso chisq.TradPerm.

The basic information of sepcified snp for 'genotypeLine' and the other information of individuals for 'affectionLine' must be located in the matrix of previous columns;and also can not be included,thus fromCol=1.

The genotypes of the specified snp, the stored alleles is considered to be ordered, i.e. "C/T" is unequivalent to "T/C".

The affection status can be character string or numeric, but the symbol for control must be in advantageous alphabetical order (e.g. control=0,case=1).

## Value

| | |
|---|---|
| pValue | the p value for the test. |
| obsStatistic | the statistic of Armitage's trend test for the true data. |
| obsP | the p value for Armitage's trend test of the true data. |
| permStatistic | a vector with 'repeatNum' elements, the statistic of Armitage's trend test for the permutation data. |
| permP | a vector with 'repeatNum' elements, the p value for Armitage's trend test for the permutation data. |

## Note

When input the parameter 'naString' and 'sep', please make sure correct. Otherwise the result will be wrong.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Armitage, P(1955): Tests for Linear Trends in Proportions and Frequencies.

statgen.org(2007): A derivation for Armitage's trend test for the 2x3 genotype table.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
## import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# Armitage.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=10000)

## matrix
# genotypeLine=matrix(c("rs12","AA","TT","TA","AA","TT","AA","AA"),nrow=1)
# affectionLine=matrix(c("Affection",1,1,1,0,0,0,0),nrow=1)
# Armitage.TradPerm(genotypeLine,affectionLine,fromCol=2,naString="NN",sep="",repeatNum=5)

## connect file
# datafile=file("F:/data.txt","r")
## get the affection status for samples, read the first line from "data.txt"
# dataLine1=readLins(datafile,n=1)
# dataLine1=t(unlist(strsplit(dataLine1,sep="")))
## get the genotype line for samples, read the second line form "data.txt"
# dataLine2=readLines(datafile,n=1)
# dataLine2=t(unlist(strsplit(dataLine2,sep="")))
# Armitage.TradPerm(dataLine2,dataLine1,fromCol=2,naString="NN",sep="",repeatNum=5)
```

---

# chisq.MCPerm *A Monte Carlo permutation method for multiple chisq.test correction in case/control association study*

---

## Description

A Monte Carlo permutation method for multiple chisq.test correction in case/control association study.

## Usage

```
chisq.MCPerm(case_11, case_12, case_22,
    control_11, control_12, control_22, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| case_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| case_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| case_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| control_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| control_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| control_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |
| repeatNum | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

Permutation tests exist for any test statistic, regardless of whether or not its distribution is known. Thus the permutation test is widely considered the gold standard for accurate multiple testing correction.

For case/control association study for snps, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the genotype frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

Obviously, for multiple test correction in case/control association study for millions of snp, the traditional method—permutation test is very computationally impractical. Thus propose an accurate, rapid and efficient method for multiple testing correction in genome-wide association studies—MCPerm.

Method—MCPerm generates the genotype frequency for rearranged case and control data by twice generating random numbers for the hypergeometric distribution, based on the genotype statistic of original data, taking the place of the step 2) and step 3) of the traditional method. And the genotype frequency distribution generating by MCPerm is almost the same with permutation test, this simplified method greatly improves the efficiency of the permutation test and is faster. MCPerm method can be the perfect alternative to permutation test.

## Value

| | |
|---|---|
| pValue | the p value for the test. |
| obsStatistic | the value of the chi-squared test statistic for the true data. |
| obsP | the p value of the chisq.test for the true data. |
| permStatistic | a matrix with one row and 'repeatNum' columns, the value of the chi-squared test statistic for simulation data. |
| permP | a matrix with one row and 'repeatNum' columns, the p value of the chisq.test for simulation data. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
# case_11=34
# case_12=0
# case_22=16
# control_11=14
# control_12=0
# control_22=13
# chisq.MCPerm(case_11,case_12,case_22,control_11,control_12,control_22,repeatNum=10000)
```

---

# chisq.TradPerm *A permutation test for multiple chisq.test correction in case/control association study*

---

## Description

A permutation test for multiple chisq.test correction in case/control association study.

## Usage

```
chisq.TradPerm(genotypeLine, affectionLine, fromCol, naString, sep, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| genotypeLine | a matrix with one row containing information of specified snp: basic information(e.g. SNP ID number,chromo,position) and genotype of observed individuals. See below for details. |
| affectionLine | a matrix having the same dimension with parameter 'genotypeLine' contain the affection status (case or control) of each individual and other information. The affection status must be in same columns with the genotypes in parameter 'genotypeLine'. See below for details. |
| fromCol | a positive integer, the start column of genotype data in parameter 'genotypeLine'. |
| naString | a character string for NA values of genotype. |
| sep | character separator used to divide genotype between alleles "Allele1<sep>Allele2". |
| repeatNum | an integer(default 1000) specifying the number of replicates for permutation test. |

## Details

For case/control association study for snps, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the genotype frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

The basic information of sepcified snp for 'genotypeLine' and the other information of individuals for 'affectionLine' must be located in the matrix of previous columns; and also can not be included,thus fromCol=1.

The genotypes of the specified snp, the stored alleles is considered to be ordered, i.e. "C/T" is unequivalent to "T/C".

The affection status can be character string or numeric, but the symbol for control must be in advantageous alphabetical order (e.g. control=0,case=1).

## Value

| | |
|---|---|
| pValue | the p value for the test. |
| obsStatistic | the value of the chi-squared test statistic for the true data. |
| obsP | the p value of the chisq.test for the true data. |
| permStatistic | a matrix with one row and 'repeatNum' columns, the value of the chi-squared test statistic for permutation data. |
| permP | a matrix with one row and 'repeatNum' columns, the p value of the chisq.test for permutation data. |

## Note

When input the parameter 'naString' and 'sep', please make sure correct. Otherwise the result will be wrong.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
## import example data
#data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
# get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# chisq.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=5)

## matrix
# genotypeLine=matrix(c("rs12","AA","TT","TA","AA","TT","AA","AA"),nrow=1)
# affectionLine=matrix(c("Affection",1,1,1,0,0,0,0),nrow=1)
# chisq.TradPerm(genotypeLine,affectionLine,fromCol=2,naString="NN",sep="",repeatNum=5)

## connect file
# datafile=file("F:/data.txt","r")
## get the affection status for samples, read the first line from "data.txt"
# dataLine1=readLins(datafile,n=1)
# dataLine1=t(unlist(strsplit(dataLine1,sep="")))
## get the genotype line for samples, read the second line form "data.txt"
# dataLine2=readLines(datafile,n=1)
# dataLine2=t(unlist(strsplit(dataLine2,sep="")))
# chisq.TradPerm(dataLine2,dataLine1,fromCol=2,naString="NN",sep="",repeatNum=1000)
```

`fisher.MCPerm` *A Monte Carlo permutation method for multiple fisher.test correction in case/control association study*

## Description

A Monte Carlo permutation method for multiple fisher.test correction in case/control association study.

## Usage

```
fisher.MCPerm(case_11, case_12, case_22,
    control_11, control_12, control_22, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| `case_11` | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| `case_12` | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| `case_22` | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| `control_11` | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| `control_12` | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| `control_22` | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |
| `repeatNum` | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

Permutation tests exist for any test statistic, regardless of whether or not its distribution is known. Thus the permutation test is widely considered the gold standard for accurate multiple testing correction.

For case/control association study for snps, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the genotype frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

Obviously, for multiple test correction in case/control association study for millions of snp, the traditional method—permutation test is very computationally impractical. Thus propose an accurate, rapid and efficient method for multiple testing correction in genome-wide association studies—MCPerm.

Method—MCPerm generates the genotype frequency for rearranged case and control data by twice generating random numbers for the hypergeometric distribution, based on the genotype statistic of original data, taking the place of the step 2) and step 3) of the traditional method. And the genotype frequency distribution gener-ating by MCPerm is almost the same with permutation test, this simplified method greatly improves the efficiency of the permutation test and is faster. MCPerm method can be the perfect alternative to permutation test.

## Value

| | |
|---|---|
| `pValue` | the p value for the test. |
| `obsP` | the p value of the fisher.test for the true data. |
| `permP` | a matrix with one row and 'repeatNum' columns, the p value of the fisher.test for simulation data. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

`OR`, `OR.TradPerm`, `OR.MCPerm`, `Armitage`, `Armitage.TradPerm`, `Armitage.MCPerm`, `chisq.test`, `chisq.TradPerm`, `chisq.MCPerm`, `fisher.test`, `fisher.TradPerm`, `meta`, `meta.TradPerm`, `meta.MCPerm`, `permuteGenotype`, `rhyper`, `permuteGenotypeCount`, `genotypeStat`

## Examples

```
# case_11=34
# case_12=0
# case_22=16
# control_11=14
# control_12=0
# control_22=13
# result=fisher.MCPerm(case_11,case_12,case_22,control_11,control_12,control_22,repeatNum=1000)
# p=result$pValue
```

`fisher.TradPerm` *A permutation test for multiple fisher.test correction in case/control association study*

## Description

A permutation test for multiple fisher.test correction in case/control association study.

## Usage

```
fisher.TradPerm(genotypeLine, affectionLine, fromCol, naString, sep, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| `genotypeLine` | a matrix with one row containing information of specified snp(rs): basic information(e.g. SNP ID number, chromo,position) and genotype of observed individuals. See below for details. |
| `affectionLine` | a matrix having the same dimension with parameter 'genotypeLine' contain the affection status(case or control) of each individual and other information. The affection status must be in same columns with the genotypes in parameter 'genotypeLine'. See below for details. |
| `fromCol` | a positive integer, the start column of genotype data in parameter 'genotypeLine'. |
| `naString` | a character string for NA values of genotype. |
| `sep` | character separator used to divide genotype between alleles "Allele1<sep>Allele2". |
| `repeatNum` | an integer(default 1000) specifying the number of replicates for permutation test. |

## Details

For case/control association study for snps, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the genotype frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

The basic information of sepcified snp for 'genotypeLine' and the other information of individuals for 'affectionLine' must be located in the matrix of previous columns; and also can not be included,thus fromCol=1.

The genotypes of the specified snp, the stored alleles is considered to be ordered, i.e. "C/T" is unequivalent to "T/C".

The affection status can be character string or numeric, but the symbol for control must be in advantageous alphabetical order (e.g. control=0,case=1).

## Value

| | |
|---|---|
| pValue | the p value for the test. |
| obsP | the p value of the fisher.test for the true data. |
| permP | a matrix with one row and 'repeatNum' columns, the p value of the fisher.test for permutation data. |

## Note

When input the parameter 'naString' and 'sep', please make sure correct. Otherwise the result will be wrong.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

OR, OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
## import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# fisher.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=5)

## matrix
# genotypeLine=matrix(c("rs12","AA","TT","TA","AA","TT","AA","AA"),nrow=1)
# affectionLine=matrix(c("Affection",1,1,1,0,0,0,0),nrow=1)
# fisher.TradPerm(genotypeLine,affectionLine,fromCol=2,naString="NN",sep="",repeatNum=5)

## connect file
# datafile=file("F:/data.txt","r")
## get the affection status for samples, read the first line from "data.txt"
# dataLine1=readLins(datafile,n=1)
# dataLine1=t(unlist(strsplit(dataLine1,sep="")))
## get the genotype line for samples, read the second line form "data.txt"
# dataLine2=readLines(datafile,n=1)
# dataLine2=t(unlist(strsplit(dataLine2,sep="")))
# fisher.TradPerm(dataLine2,dataLine1,fromCol=2,naString="NN",sep="",repeatNum=1000)
```

genotypeData *Genotype Data from GWA16*

## Description

Genotype Data from GWA16.

## Usage

```
data(genotypeData)
```

## Format

A matrix with 10 snps with 2062 samples. The format is: first row$: affection state for 2062 samples; other rows$: genotype data for each snp, e.g. A_A A_G G_G ?_? ...

## Source

GAW16 Data from the North American Rheumatoid Arthritis Consortium (NARAC)

## References

MacCluer JW, Cupples LA and Almasy L (eds) Genetic Analysis Workshop 16: Approaches to Analysis of Genome-Wide Data. Genetic Epidemiology 33 (Suppl 1), S1-S110 (2009).

## Examples

```
## import example data(data.frame)
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]

## Statistical allele and genotype frequency of the specified snp(rs#) for case-control data.
# result2=genotypeStat(data2,data1,fromCol=2,naString="?_?",sep="_")
## allele frequency for case and control samples
# alleleResult=result2$alleleCount
## genotype frequency for case and control samples
# genotypeReslut=result2$genotypeCount

## use permutation test to do multiple test correction in case/control association study
## return the correlated p_value and other information, see 'chisq.TradPerm'.
# result1=chisq.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=1000)
# p1=result1$pValue
# result2=fisher.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=1000)
# p2=result2$pValue
# result3=Armitage.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=1000)
# p3=result3$pValue
```

```
# result4=OR.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=1000)
# risk_allele=result4$risk_allele
# p4=result4$pValue
```

---

genotypeStat *Statistical Allele and Genotype Frequency of the specified snp*

---

## Description

Statistical Allele and Genotype Frequency of the specified snp.

## Usage

```
genotypeStat(genotypeLine, affectionLine, fromCol, naString, sep)
```

## Arguments

| | |
|---|---|
| genotypeLine | a matrix with one row containing information of specified snp: basic information(e.g. SNP ID number, chromo,position) and genotype of observed individuals. See below for details. |
| affectionLine | a matrix having the same dimension with parameter 'genotypeLine' contain the affection status(case or control) of each individual and other information. The affection status must be in same columns with the genotypes in parameter 'genotypeLine'. See below for details. |
| fromCol | a positive integer, the start column of genotype data in parameter 'genotypeLine'. |
| naString | a character string for NA value of genotype. |
| sep | character separator used to divide genotype between alleles "Allele1<sep>Allele2". |

## Details

The basic information of sepcified snp for 'genotypeLine' and the other information of individuals for 'affectionLine' must be located in the matrix of previous columns; and also can not be included,thus fromCol=1.

The genotypes of the specified snp, the stored alleles is considered to be ordered, i.e. "C/T" is unequivalent to "T/C".

The affection status can be character string or numeric, but the symbol for control must be in advantageous alphabetical order (e.g. control=0,case=1).

## Value

| | |
|---|---|
| alleleCount | A vector with four elements: 'case_allele1','case_allele2','control_allele1', 'control_allele2'. |
| genotypeCount | A vector with eight elements: 'case_11','case_12','case_22','case_NA','control_11', 'control_12','control_22','control_NA'. |

## Note

When input the parameter 'naString' and 'sep', please make sure correct. Otherwise the result will be wrong.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount

## Examples

```
## import example data(data.frame)
# data(genotypeData)
## get the first line: affection state for samples
#data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]

## Statistical allele and genotype frequency of the specified snp(rs#) for case-control data.
# result2=genotypeStat(data2,data1,fromCol=2,naString="?_?",sep="_")
## allele frequency for case and control samples
# alleleResult=result2$alleleCount
## genotype frequency for case and control samples
# genotypeReslut=result2$genotypeCount

## matrix
# genotypeLine=matrix(c("rs12","AA","AT","NN","AT","AT","AT"),nrow=1)
# affectionLine=matrix(c("Affection",0,0,1,1,0,0),nrow=1)
# fromCol=2
# naString="NN"
# sep=""
# genotypeStat(genotypeLine, affectionLine, fromCol, naString, sep)

## connect file
# datafile=file("F:/data.txt","r")
## get the affection status of samples: the first line from "data.txt"
# dataLine1=readLines(datafile,n=1)
# dataLine1=t(unlist(strsplit(dataLine1,sep="")))
## get the genotypes of samples: the second line from "data.txt"
# dataLine2=readLines(datafile,n=1)
# dataLine2=t(unlist(strsplit(dataLine2,sep="")))
```

```
# genotypeStat(dataLine2,dataLine1,fromCol=2,naString="NN",sep="")
```

---

`HW.test` *Hardy-weinberg equilibrium test*

---

## Description

Hardy-weinberg equilibrium test for control samples.

## Usage

```
HW.test(count_11, count_12, count_22)
```

## Arguments

| | |
|---|---|
| `count_11` | a non-negative integer or vector, the frequency of genotype "allele1/allele1" in control samples. |
| `count_12` | a non-negative integer or vector, the frequency of genotype "allele1/allele2" in control samples. |
| `count_22` | a non-negative integer or vector, the frequency of genotype "allele2/allele2" in control samples. |

## Details

Hardy-weinberg equilibrium test states that allele and genotype frequencies in a population will remain constant from generation to generation in the absence of other evolutionary influences.

## Value

| | |
|---|---|
| `X2` | statistic(s) for the Hardy-weinberg equilibrium test. |
| `p.value` | p.value(s) for the Hardy-weinberg equilibrium test. |

## Note

Hardy-weinberg equilibrium test for control samples.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Emigh, T.H. (1980 Biometrics 36(4): 627-642): A comparison of tests for Hardy-weinberg equilibrium.

## See Also

OR, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
# case_11=c(12,23,34,33)
# case_12=c(4,34,53,4)
# case_22=c(7,5,23,9)
# HW.test(case_11,case_12,case_22)
```

---

# I2.MCPerm *Calculate p.value for Heterogeneity statistics I2 in meta analysis*

---

## Description

Calculate p.value for Heterogeneity statistics I2 in meta analysis.

## Usage

```
I2.MCPerm(case_11, case_12, case_22, control_11, control_12, control_22,
    model = "allele", method = "MH", repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| case_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| case_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| case_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| control_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| control_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| control_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |
| model | a character string indicating the type of model("allele"(default),"dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. |
| method | a character string indicating the method('Inverse','MH'(default) or 'Peto') to calculate Q value. See details. |
| repeatNum | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

I2 is calculated by formula I2=max(Q-d.f./Q, 0), considering I2=1-24 moderate heterogeneity; I2=50-74

MCPerm details see `chisq.MCPerm`.

## Value

| | |
|---|---|
| `risk_allele` | the symbol of risk allele.See details. |
| `I2` | the I2 statistics for observed meta data. |
| `corrected_I2p` | the p value for I2, the percentage of more than I2 value. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Julian P.T.Higgins, Simon G.Thompson(Statistics in Medicine,2002): Quantifying heterogeneity in a meta-analysis.

Julian P.T.Higgins, Simon G.Thompson, Jonathan J Deeks(BMJ,2003):Measuring inconsistency in meta-analyses.

## See Also

`meta.MCPerm`, `meta.TradPerm`, `Q.TradPerm`, `I2.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.Hist`, `VS.KS`, `VS.Allele.Hist`, `VS.Genotype.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.Hist`

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=I2.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result
```

`I2.TradPerm` *Calculate p.value for Heterogeneity statistics I2 in meta analysis*

## Description

Calculate p.value for Heterogeneity statistics I2 in meta analysis

## Usage

```
I2.TradPerm(genotypeData, affectionData, split, sep, naString,
    model = "allele", method = "MH", repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| `genotypeData` | a matrix with one column and multiple rows, each row contains genotype data for case and control samples of certain study. Note the field separtor of each line must be same, and same with parameter 'affectionData'. |
| `affectionData` | a matrix with one column and multiple rows, each row contains the affection stats of case and control samples of certain study which must correspond to 'genotypeData'. Note the field separtor of each line must be same,and same with parameter 'genotypeData'. |
| `split` | the field separator character, which separates elements on each line of the parameter 'genotypeData' and 'affectionData'. 'Split' and 'sep' cannot be same. |
| `sep` | character separator used to divide genotype between alleles "Allele1<sep>Allele2" in parameter 'genotypeData'. 'Split' and 'sep' cannot be same. |
| `naString` | a character string for NA values of genotype data in parameter 'genotypeData'. |
| `model` | a character string indicating the type of model("allele"(default),"dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. |
| `method` | a character string indicating the method('Inverse','MH'(default) or 'Peto') to calculate Q value. See details. |
| `repeatNum` | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

I2 is calculated by formula I2=max(Q-d.f./Q, 0), considering I2=1-24 moderate heterogeneity; I2=50-74

TradPerm details see `chisq.TradPerm`.

## Value

| | |
|---|---|
| risk_allele | the symbol of risk allele.See details. |
| I2 | the I2 statistics for true meta data. |
| corrected_I2p | the p value for I2, the percentage of more than I2 value. |

## Note

'Split':the field separtor of each line for parameter 'genotypeData' and 'affectionData' must be same. 'Split' and 'sep' cannot be same.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Julian P.T.Higgins, Simon G.Thompson(Statistics in Medicine,2002): Quantifying heterogeneity in a meta-analysis.

Julian P.T.Higgins, Simon G.Thompson, Jonathan J Deeks(BMJ,2003):Measuring inconsistency in meta-analyses.

## See Also

meta.MCPerm, meta.TradPerm, Q.TradPerm, I2.MCPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Allele.Hist, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.LnOR.CDC, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result=I2.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result
```

meta *Meta analysis*

## Description

Meta analysis.

## Usage

```
meta(model, fixed_method, random_method, Qp_alpha, case_11, case_12, case_22,
    control_11, control_12, control_22, label = NULL, dataset = NULL)
```

## Arguments

| | |
|---|---|
| model | a character string indicating the type of model("allele","dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. |
| fixed_method | a character string indicating the method('Inverse','MH' or 'Peto') to fit fixed-effects model. |
| random_method | a character string indicating the method to fit random-effects model: "HE","DL", "SJ","ML","REML" or "EB". |
| Qp_alpha | the threshold value(0-1) to refuse the null hypothesis that all studies were evaluating the same effect. Not rejecting the above hypothesis usually leads a meta-analysis to adopt a fixed-effects model. The fixed-effects model assumes that the estimated effect sizes only differ by the sampling error. In contrast, if a significant Q-statistic P<Qp_alpha indicates heterogeneity across studies, a random-effects model should be adopted. |
| case_11 | non-negative integer vector, the frequency of genotype <allele1/allele1> in case samples. |
| case_12 | non-negative integer vector, the frequency of genotype <allele1/allele2> in case samples. |
| case_22 | non-negative integer vector, the frequency of genotype <allele2/allele2> in case samples. |
| control_11 | non-negative integer vector, the frequency of genotype <allele1/allele1> in control samples. |
| control_12 | non-negative integer vector, the frequency of genotype <allele1/allele2> in control samples. |
| control_22 | non-negative integer vector, the frequency of genotype <allele2/allele2> in control samples. |
| label | character string vector, labels for the studies to show in the forest plot. When label=NULL(default), the label for the ith study will be "study i". Also see parameter 'dataset'. |

| | |
|---|---|
| dataset | data frame containing the data of genotype frequency(e.g. 'case_11','case_12') and the information of the studies(e.g. author,year). When dataset=NULL(default), the value of parameters 'case_11', 'case_12', 'case_22', 'control_11', 'control_12', 'control_22', 'label' must be vector with same length(>1). When dataset is not NULL, the value of parameters 'case_11', 'case_12', 'case_22', 'control_11', 'control_12', 'control_22' must be a non-negative integer, that specify the column of parameter needed value in 'dataset'; and lable can be a non-negative integer or vector that will be connected by " " as the labels for studies. |

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

Meta-analysis refers to methods focused on contrasting and combining results from different studies, in the hope of identifying patterns among study results, sources of disagreement among those results, or other interesting relationships that may come to light in the context of multiple studies. In its simplest form, this is normally by identification of a common measure of effect size, of which a weighted average might be the output of a meta-analysis. The weighting might be related to sample sizes within the individual studies. More generally there are other differences between the studies that need to be allowed for, but the general aim of a meta-analysis is to more powerfully estimate the true effect size as opposed to a less precise effect size derived in a single study under a given single set of assumptions and conditions.

## Value

The return value detail see package metafor. See package:metafor. If the return result is fixed-effect model, see metafor:rma(method='FE'), metafor:rma.mh, metafor:rma.peto. If the return result is random-effect model, see metafor:rma.

| | |
|---|---|
| b | combined log odds ratio. |
| se | standard errors of the combined log odds ratio. |
| zval | test statistics of the combined log odds ratio. |
| pval | p-values for the test statistics. |
| ci.lb | lower bound of the confidence intervals for the combined log odds ratio. |
| ci.ub | upper bound of the confidence intervals for the combined log odds ratio. |
| QE | test statistic for the test of heterogeneity. |
| QEp | p-value for the test of heterogeneity. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does mutiple testing correction work?

Hedges,L.V. & Vevea,J.L.(1998): Fixed- and random- effects models in meta-analysis.

## See Also

meta.TradPerm, meta.MCPerm, permuteGenotype, permuteGenotypeCount, genotypeStat, OR.TradPerm, Armitage.TradPerm, chisq.TradPerm, fisher.TradPerm, VS.Genotype.Hist, VS.Allele.Hist, VS.Hist, PermMeta.LnOR.Hist, PermMeta.LnOR.CDC, PermMeta.LnOR.boxplot, PermMeta.boxplot, PermMeta.Hist, pearson_scatter, Q.TradPerm, I2.TradPerm

## Examples

```
# case_11=c(1,2,3,4,7)
# case_12=c(1,2,2,2,4)
# case_22=c(1,2,3,4,5)
# control_11=c(1,4,5,6,3)
# control_12=c(3,4,5,6,6)
# control_22=c(3,6,7,8,5)
# result1=meta("allele","Inverse","DL",0.05,case_11,case_12,case_22,
    # control_11,control_12,control_22)
# result2=meta("dominant","MH","DL",0.05,case_11,case_12,case_22,
    # control_11,control_12,control_22)
# result3=meta("recessive","Peto","DL",0.05,case_11,case_12,case_22,
    # control_11,control_12,control_22)
```

---

# meta.MCPerm *Meta analysis corrected by permutation test*

---

## Description

Meta analysis corrected by permutation test.

## Usage

```
meta.MCPerm(case_11, case_12, case_22, control_11, control_12, control_22,
    model = "allele", fixed_method = "MH", random_method = "DL",
 Qp_alpha = 0.01, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| case_11 | non-negative integer vector, the frequency of genotype <allele1/allele1> in case samples. |
| case_12 | non-negative integer vector, the frequency of genotype <allele1/allele2> in case samples. |
| case_22 | non-negative integer vector, the frequency of genotype <allele2/allele2> in case samples. |
| control_11 | non-negative integer vector, the frequency of genotype <allele1/allele1> in control samples. |
| control_12 | non-negative integer vector, the frequency of genotype <allele1/allele2> in control samples. |

`control_22` non-negative integer vector, the frequency of genotype <allele2/allele2> in control samples.

`model` a character string indicating the type of model("allele"(default),"dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>.

`fixed_method` a character string indicating the method('Inverse','MH'(default) or 'Peto') to fit fixed-effects model.

`random_method` a character string indicating the method to fit random-effects model: "HE", "DL"(default), "SJ", "ML", "REML", or "EB".

`Qp_alpha` the threshold value(0~1)(default 0.01) to refuse the null hypothesis that all studies were evaluating the same effect. Not rejecting the above hypothesis usually leads a meta-analysis to adopt a fixed-effects model. The fixed-effects model assumes that the estimated effect sizes only differ by the sampling error. In contrast, if a significant Q-statistic P<Qp_alpha indicates heterogeneity across studies, a random-effects model should be adopted.

`repeatNum` an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation.

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

MCPerm details see `chisq.MCPerm`.

## Value

`corrected_result`
matrix with 3 rows and 3 columns. The first column is Cochran's Q-statistics value; the second column is I2(=max(Q-df/Q,0), the degree of inconsistency across studies); the third column is the merged log odds ratio. The first row is the value for the 3 columns calculated by true data of the studies; the second row is the p value calculated by true data of the studies; the third row is the p value calculated by permutation data, namely equal to the proportion of the statistic of interest as extreme as the observed value of the same statistic. Note: the p value for I2 donot calculate throngh meta-analysis, so the value is NA.

`risk_allele` the symbol of risk allele.See details.

`true_merged_LnOR`
the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_VAR`
variation of the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_p`
p value for the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_ci.lb`
lower bound of the confidence intervals for the merged log odds ratio.

`true_merged_LnOR_ci.ub`
: upper bound of the confidence intervals for the merged log odds ratio.

`study_num`
: the number of studies in the meta analysis.

`sample`
: a vector with 'study_num' elements, the sample size of each study.

`true_LnOR`
: a vector of log odd ratio calculated by each study.

`true_VARLnOR`
: a vector, the varition of log odd ratio calculated by each study.

`perm_case_11,perm_case_12,perm_case_22,perm_control_11,perm_control_12,perm_control_22`
: a matrix with one row and 'repeatNum' columns, the freqency for genotype got by permutating data by MCPerm method. MCPerm method details see `chisq.MCPerm`.

`perm_LnOR`
: matirx with rows indicating studies and 'repeatNum' columns indicating log odds ratio for each permutation data.

`perm_VARLnOR`
: matrix with rows indicating studies and 'repeatNum' columns indicating variance for the log odds ratio for each permuation data.

`perm_Qp`
: a matrix with one row and 'repeatNum' columns, statistic Q value of heterogeneity for each permutation data.

`perm_I2`
: a matrix with one row and 'repeatNum' columns, statistic I2 value of heterogeneity for each permutation data.

`perm_merged_LnOR`
: a matrix with one row and 'repeatNum' columns, merged log odd ratio for permutation data.

`perm_merged_VARLnOR`
: a matrix with one row and 'repeatNum' columns, variation of merged log odd ratio for permuation data.

`perm_p`
: a matrix with one row and 'repeatNum' columns, p value for merged log odd ratio of permuation data.

`model,fixed_method,random_method,Qp_alpha,repeatNum`
: value for parmeter of the funciton.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does mutiple testing correction work?

Hedges,L.V. & Vevea,J.L.(1998): Fixed- and random- effects models in meta-analysis.

## See Also

`meta`, `meta.TradPerm`, `permuteGenotype`, `permuteGenotypeCount`, `genotypeStat`, `OR.TradPerm`, `Armitage.TradPerm`, `chisq.TradPerm`, `fisher.TradPerm`, `VS.Genotype.Hist`, `VS.Allele.Hist`, `VS.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.boxplot`, `PermMeta.Hist`, `pearson_scatter`, `Q.TradPerm`, `I2.TradPerm`

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result
```

---

`meta.TradPerm` *Meta analysis corrected by permutation test*

---

## Description

Meta analysis corrected by permutation test.

## Usage

```
meta.TradPerm(genotypeData, affectionData, split, sep, naString,
    model = "allele", fixed_method = "MH", random_method = "DL",
 Qp_alpha = 0.01, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| `genotypeData` | a matrix with one column and multiple rows, each row contains genotype data for case and control samples of certain study. Note the field separtor of each line must be same, and same with parameter 'affectionData'. |
| `affectionData` | a matrix with one column and multiple rows, each row contains the affection stats of case and control samples of certain study which must correspond to 'genotypeData'. Note the field separtor of each line must be same,and same with parameter 'genotypeData'. |
| `split` | the field separator character, which separates elements on each line of the parameter 'genotypeData' and 'affectionData'. 'Split' and 'sep' cannot be same. |
| `sep` | character separator used to divide genotype between alleles "Allele1<sep>Allele2" in parameter 'genotypeData'. 'Split' and 'sep' cannot be same. |
| `naString` | a character string for NA values of genotype data in parameter 'genotypeData'. |
| `model` | a character string indicating the type of model("allele"(default),"dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. |
| `fixed_method` | a character string indicating the method('Inverse','MH'(default) or 'Peto') to fit fixed-effects model. |

`random_method` a character string indicating the method to fit random-effects model: "HE", "DL"(default), "SJ", "ML", "REML", or "EB".

`Qp_alpha` the threshold value(0~1)(default 0.01) to refuse the null hypothesis that all studies were evaluating the same effect. Not rejecting the above hypothesis usually leads a meta-analysis to adopt a fixed-effects model. The fixed-effects model assumes that the estimated effect sizes only differ by the sampling error. In contrast, if a significant Q-statistic P<Qp_alpha indicates heterogeneity across studies, a random-effects model should be adopted.

`repeatNum` an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation.

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

TradPerm details see `chisq.TradPerm`.

## Value

`corrected_result`
matrix with 3 rows and 3 columns. The first column is Cochran's Q-statistics value; the second column is I2(=max(Q-df/Q,0), the degree of inconsistency across studies); the third column is the merged log odds ratio. The first row is the value for the 3 columns calculated by true data of the studies; the second row is the p value calculated by true data of the studies; the third row is the p value calculated by permutation data, namely equal to the proportion of the statistic of interest as extreme as the observed value of the same statistic. Note: the p value for I2 donot calculate throngh meta-analysis, so the value is NA.

`risk_allele` the symbol of risk allele.See details.

`true_merged_LnOR`
the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_VAR`
variation of the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_p`
p value for the merged log odd ratio by meta analysis using observation data.

`true_merged_LnOR_ci.lb`
lower bound of the confidence intervals for the merged log odds ratio.

`true_merged_LnOR_ci.ub`
upper bound of the confidence intervals for the merged log odds ratio.

`study_num` the number of studies in the meta analysis.

`sample` a vector with 'study_num' elements, the sample size of each study.

`true_LnOR` a vector of log odd ratio calculated by each study.

`true_VARLnOR` a vector, the varition of log odd ratio calculated by each study.

`perm_case_11,perm_case_12,perm_case_22,perm_control_11,perm_control_12,perm_control_22`
a matrix with one row and 'repeatNum' columns, the freqency for genotype got by permutating data by MCPerm method. MCPerm method details see `chisq.MCPerm`.

`perm_LnOR` matirx with rows indicating studies and 'repeatNum' columns indicating log odds ratio for each permutation data.

`perm_VARLnOR` matrix with rows indicating studies and 'repeatNum' columns indicating variance for the log odds ratio for each permuation data.

`perm_Qp` a matrix with one row and 'repeatNum' columns, statistic Q value of heterogeneity for each permutation data.

`perm_I2` a matrix with one row and 'repeatNum' columns, statistic I2 value of heterogeneity for each permutation data.

`perm_merged_LnOR` a matrix with one row and 'repeatNum' columns, merged log odd ratio for permutation data.

`perm_merged_VARLnOR` a matrix with one row and 'repeatNum' columns, variation of merged log odd ratio for permuation data.

`perm_p` a matrix with one row and 'repeatNum' columns, p value for merged log odd ratio of permuation data.

`model,fixed_method,random_method,Qp_alpha,repeatNum` value for parmeter of the funciton.

## Note

'Split':the field separtor of each line for parameter 'genotypeData' and 'affectionData' must be same. 'Split' and 'sep' cannot be same.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does mutiple testing correction work?

Hedges,L.V. & Vevea,J.L.(1998): Fixed- and random- effects models in meta-analysis.

## See Also

`permuteGenotype`, `permuteGenotypeCount`, `genotypeStat`, `OR.TradPerm`, `Armitage.TradPerm`, `chisq.TradPerm`, `fisher.TradPerm`, `meta`, `meta.MCPerm`, `VS.Genotype.Hist`, `VS.Allele.Hist`, `VS.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.boxplot`, `PermMeta.Hist`, `pearson_scatter`, `Q.TradPerm`, `I2.TradPerm`

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
```

```
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result
```

---

`MetaGenotypeCount` *rs3131296 genetic association studies from SZGene database*

---

## Description

rs3131296 genetic association studies from SZGene database.

## Usage

```
data(MetaGenotypeCount)
```

## Format

The format is: 22 lines: the first line is the names of columns; other lines are studies of rs3131296.

## Source

http://www.szgene.org/polydetail.asp?geneID=389&studyID=1587&ethnicDataID=3452

## References

Stefansson H, Ophoff RA etc(Nature 2009 Aug 6;460(7256):744-7):Common variants conferring risk of schizophrenia.

## Examples

```
# data(MetaGenotypeCount)
```

## MetaGenotypeData *genotype raw data for rs3131296 genetic association studies*

### Description

genotype raw data for rs3131296 genetic association studies, got by simulation based on the count of genotype from SZGene database.

### Usage

```
data(MetaGenotypeData)
```

### Format

The format is: 22 lines: the first line is the names of columns, other lines are information of studies. The 14-th and 15-th column separately are gentoype data for case and control samples.

### References

Stefansson H, Ophoff RA etc(Nature 2009 Aug 6;460(7256):744-7):Common variants conferring risk of schizophrenia.

### Examples

```
# data(MetaGenotypeData)
```

## OR *OR(odd ratio) for risk_allele*

### Description

OR(odd ratio) for risk_allele.

### Usage

```
OR(case_allele1, case_allele2, control_allele1, control_allele2)
```

### Arguments

| | |
|---|---|
| case_allele1 | a non-negative integer, the frequency of allele1 in case samples. |
| case_allele2 | a non-negative integer, the frequency of allele2 in case samples. |
| control_allele1 | a non-negative integer, the frequency of allele1 in control samples. |
| control_allele2 | a non-negative integer, the frequency of allele2 in control samples. |

## Details

The odds ratio is a measure of effect size, describing the strength of association or non-independence between two binary data values.

## Value

| | |
|---|---|
| risk_allele | risk_allele(OR>1). |
| OR | the value of OR. |
| CI | a matrix with one row, the 0.95 CI(Confidence Interval) of OR: lower limit OR, upper limit OR. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Edwards, A.W.F.(1963): The measure of association in a 2x2 table.

## See Also

OR.TradPerm, OR.MCPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
# case_allele1=23
# case_allele2=34
# control_allele1=26
# control_allele2=29
# OR(case_allele1, case_allele2, control_allele1, control_allele2)
```

---

# OR.MCPerm *A Monte Carlo permutation method for multiple OR(odd ratio) test correction in case/control association study*

---

## Description

A Monte Carlo permutation method for multiple OR(odd ratio) test correction in case/control association study.

## Usage

```
OR.MCPerm(case_allele1, case_allele2, control_allele1, control_allele2, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| case_allele1 | a non-negative integer, the frequency of allele1 in case samples. |
| case_allele2 | a non-negative integer, the frequency of allele2 in case samples. |
| control_allele1 | a non-negative integer, the frequency of allele1 in control samples. |
| control_allele2 | a non-negative integer, the frequency of allele2 in control samples. |
| repeatNum | an integer(default 1000) specifying the number of replicates used in the Monte Carlo permutation. |

## Details

The odds ratio is a measure of effect size, describing the strength of association or non-independence between two binary data values.

"OR.MCPerm" simulates permutation method to get p value for risk_allele, by generating random numbers for the hypergeometric distribution based on the genotype statistic of original data. Seealso chisq.MCPerm.

## Value

| | |
|---|---|
| risk_allele | risk_allele. |
| pValue | the p value for the test. |
| obsOR | the OR value for the true data. |
| permOR | a vector with 'repeatNum' elements, the OR value for the simulation data. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edwards, A.W.F.(1963): The measure of association in a 2x2 table.

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

OR, OR.TradPerm, Armitage, Armitage.TradPerm, Armitage.MCPerm, chisq.test, chisq.TradPerm, chisq.MCPerm, fisher.test, fisher.TradPerm, fisher.MCPerm, meta, meta.TradPerm, meta.MCPerm, permuteGenotype, rhyper, permuteGenotypeCount, genotypeStat

## Examples

```
# case_allele1=23
# case_allele2=34
# control_allele1=26
# control_allele2=29
# OR.MCPerm(23,34,26,29,repeatNum=100000)
```

OR.TradPerm *A permutation test for multiple OR(odd ratio) test correction in case/control association study*

## Description

A permutation test for multiple OR(odd ratio) test correction in case/control association study.

## Usage

```
OR.TradPerm(genotypeLine, affectionLine, fromCol, naString, sep, repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| genotypeLine | a matrix with one row containing information of specified snp: basic information(e.g. SNP ID number, chromo,position) and genotype of observed individuals. See below for details. |
| affectionLine | a matrix having the same dimension with parameter 'genotypeLine' contain the affection status(case or control) of each individual and other information. The affection status must be in same columns with the genotypes in parameter 'genotypeLine'. See below for details. |
| fromCol | a positive integer, the start column of genotype data in parameter 'genotypeLine'. |
| naString | a character string for NA values of genotype. |
| sep | character separator used to divide genotype between alleles "Allele1<sep>Allele2". |
| repeatNum | an integer(default 1000) specifying the number of replicates for permutation test. |

## Details

The odds ratio is a measure of effect size, describing the strength of association or non-independence between two binary data values.

For case/control association study for snps, the permutation test proceeds as follows:

1) Combine the observations from all the samples;

2) Shuffle them and rearrangements of the labels(case/control) on the observed data;

3) Record the allele frequency of case and control samples, respectively;

4) Calculate the statistic of interest;

5) Repeat many times(at least 1000) to obtain the distribution of the statistic;

6) Determine how often the resampled statistic of interest is as extreme as the observed value of the same statistic.

The basic information of sepcified snp for 'genotypeLine' and the other information of individuals for 'affectionLine' must be located in the matrix of previous columns; and can not be included,thus fromCol=1.

The genotypes of the specified snp, the stored alleles is considered to be ordered, i.e. "C/T" is unequivalent to "T/C".

The affection status can be character string or numeric, but the symbol for control must be in advantageous alphabetical order (e.g. control=0,case=1).

## Value

| | |
|---|---|
| `risk_allele` | risk_allele(OR>1). |
| `pValue` | the p value for the test. |
| `obsOR` | the OR value for the true data. |
| `permOR` | a vector with 'repeatNum' elements, the OR value for the permutation data. |

## Note

When input the parameter 'naString' and 'sep', please make sure correct. Otherwise the result will be wrong.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edwards, A.W.F.(1963): The measure of association in a 2x2 table.

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

`OR`, `OR.MCPerm`, `Armitage`, `Armitage.TradPerm`, `Armitage.MCPerm`, `chisq.test`, `chisq.TradPerm`, `chisq.MCPerm`, `fisher.test`, `fisher.TradPerm`, `fisher.MCPerm`, `meta`, `meta.TradPerm`, `meta.MCPerm`, `permuteGenotype`, `rhyper`, `permuteGenotypeCount`, `genotypeStat`

## Examples

```
## import example data
# data(genotypeData)
## get the first line: affection state for samples
# data1=genotypeData[1,,drop=FALSE]
## get the second line: genotype data for a sepcifed snp
# data2=genotypeData[2,,drop=FALSE]
# OR.TradPerm(data2,data1,fromCol=2,naString="?_?",sep="_",repeatNum=5)

## matrix
# genotypeLine=matrix(c("rs12","AA","TT","TA","AA","TT","AA","AA"),nrow=1)
# affectionLine=matrix(c("Affection",1,1,1,0,0,0,0),nrow=1)
# OR.TradPerm(genotypeLine,affectionLine,fromCol=2,naString="NN",sep="",repeatNum=5)

## connect file
```

```
# datafile=file("F:/data.txt","r")
## get the affection status for samples, read the first line from "data.txt"
# dataLine1=readLins(datafile,n=1)
# dataLine1=t(unlist(strsplit(dataLine1,sep="")))
## get the genotype line for samples, read the second line form "data.txt"
# dataLine2=readLines(datafile,n=1)
# dataLine2=t(unlist(strsplit(dataLine2,sep="")))
# OR.TradPerm(dataLine2,dataLine1,fromCol=2,naString="NN",sep="",repeatNum=1000)
```

---

`pearson_scatter` *scatter plot and calculate Pearson correlation coefficient for paired data*

---

## Description

scatter plot and calculate Pearson correlation coefficient for paired data.

## Usage

```
pearson_scatter(Trad_data, MC_data, scatter_col = "gray28", line_col = "black",
    title = NULL, xlab = "TradPerm P-value", ylab = "MCPerm P-value")
```

## Arguments

| | |
|---|---|
| `Trad_data` | numeric vector, e.g. the result('perm_Qp'/'perm_I2'/'perm_p') of function 'meta.TradPerm'. |
| `MC_data` | numeric vector, e.g. the result('perm_Qp'/'perm_I2'/'perm_p') of function 'meta.MCPerm'. |
| `scatter_col` | the color(default 'gray28') of the scatter points. |
| `line_col` | the color(default 'black') of the line x=y. |
| `title` | The main title (on top). |
| `xlab,ylab` | X axis label, default value is 'TradPerm P-value'. Y axis label, default value is 'MCPerm P-value'. |

## Details

Scatter plot and Pearson correlation coefficient(two.sided) for 'perm_Qp'/'perm_I2'/'perm_p' of 'meta.TradPerm' and 'meta.MCPerm' are to test the consistency between them.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.Hist`, `VS.KS`, `VS.Genotype.Hist`, `VS.Allele.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.LnOR.qqnorm`, `PermMeta.Hist`, `PermMeta.boxplot`

## Examples

```
# Trad=read.table("Trad_result.txt",sep=" ",header=FALSE)
# MC=read.table("MC_result.txt",sep=" ",header=FALSE)
# par(mfrow=c(3,1))
# pearson_scatter(as.numeric(Trad[,4]),as.numeric(MC[,4]),
    # title="Q p_value ",
    # xlab="TradPerm Qp_value",
    # ylab="MCPerm Qp_value")
# pearson_scatter(as.numeric(Trad[,6]),as.numeric(MC[,6]),
    # title="I2 p_value",
    # xlab="TradPerm I2p_value",
    # ylab="MCPerm I2p_value")
# pearson_scatter(as.numeric(Trad[,9]),as.numeric(MC[,9]),
    # title="p_value",
    # xlab="TradPerm p_value",
    # ylab="MCPerm p_value")
```

---

# PermMeta.boxplot *boxplot for the result of 'meta.MCPerm' or 'meta.TradPerm'*

---

## Description

boxplot for the result of 'meta.MCPerm' or 'meta.TradPerm'

## Usage

```
PermMeta.boxplot(PermMeta, plot = "Qp", true_data_pch = 3, pch_col = "red",
    border_col = "red", fill_col = NULL,
 main = "boxplot for heterogeneity Q p_vlaue", digits = 3)
```

## Arguments

| | |
|---|---|
| PermMeta | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| plot | a character string indicating which return value of function 'meta.TradPerm' or 'meta.MCPerm' to be plot. The value can be "Qp"(default), "I2", "merged_LnOR", "merged_LnOR_VAR" or "merged_LnOR_p". And the value must be simulation data. 'Qp', "I2", "merged_LnOR", "merged_LnOR_VAR" and "merged_LnOR_p" separetly plots the return value 'perm_Qp', 'perm_I2', 'perm_merged_LnOR', 'perm_merged_VARLnOR', 'perm_merged_p'. |
| true_data_pch | the pch(default 3) to mark the observed value in the boxplot of simulation data. |
| pch_col | the color(default 'red') of pch. |
| border_col | the color(default 'red') for the border of boxplot. |
| fill_col | the filled color(default NULL) for the body of boxplot. |
| main | the main title (on top), default value is "boxplot for heterogeneity Q p_vlaue". |
| digits | integer(default 3) indicating the number of decimal places. See details. |

## Details

boxplot for the return value('perm_Qp','perm_I2','perm_merged_LnOR','perm_merged_VARLnOR', 'perm_merged_p') of 'meta.MCPerm' or 'meta.TradPerm'. And through 'pch' and text to mark the observed value.

The symbols in the legend: 'Q_stat' is the Q statistic for meta data heterogeneity; 'Q_p' is the p value of Q value(chi square distribution,the number of studies in meta analysis minus one is degree of freedom of Q value.); 'p.corrected' is the corrected p value by permutation; 'I2_stat' is the statistic I2(calculated by formula max(Q-d.f./Q, 0)) for meta data heterogeneity; 'merged_LnOR' is the merged log odd ratio of observed data; 'merged_LnOR_VAR' is the variance of log odd ratio for observed data; 'merged_LnOR_p' is the p value of log odd ratio of observed data which obey normal distribution.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.Hist`, `VS.KS`, `VS.Genotype.Hist`, `VS.Allele.Hist`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.LnOR.qqnorm`, `PermMeta.Hist`

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
## set working directory to save the plots.
# setwd("D:\")
# pdf("PermMeta.boxplot.pdf",height=6,width=6)
# PermMeta.boxplot(result,plot="Qp",
    # true_data_pch=5,pch_col='red',border_col='black',fill_col=NULL,
    # main="boxplot for heterogeneity Q p_vlaue")
# PermMeta.boxplot(result,plot="I2",
    # true_data_pch=5,pch_col="red",border_col='black',fill_col=NULL,
    # main="boxplot plot for heterogeneity I2")
# PermMeta.boxplot(result,plot="merged_LnOR",
    # true_data_pch=5,pch_col="red",border_col='black',fill_col=NULL,
 # main="boxplot plot for merged_LnOR")
# PermMeta.boxplot(result,plot="merged_LnOR_VAR",
    # true_data_pch=5,pch_col="red",border_col='black',fill_col=NULL,
    # main="boxplot plot for merged_LnOR_VAR")
# PermMeta.boxplot(result,plot="merged_LnOR_p",
    # true_data_pch=5,pch_col="red",border_col='black',fill_col=NULL,
```

```
    # main="boxplot plot for merged_LnOR_p")
# dev.off()
```

---

`PermMeta.Hist` *histplot for the result of 'meta.MCPerm' or 'meta.TradPerm'*

---

## Description

histplot for the result of 'meta.MCPerm' or 'meta.TradPerm'.

## Usage

```
PermMeta.Hist(PermMeta, plot = "Qp", fill_col = NULL, border_col = "black",
    arrows_col = "red", main = "Hist plot for heterogeneity Q p_vlaue",
 xlab = "Q p_value", ylab = "Density", digits = 3)
```

## Arguments

| | |
|---|---|
| `PermMeta` | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| `plot` | a character string indicating which return value of function 'meta.TradPerm' or 'meta.MCPerm' to be plot. The value can be "Qp"(default), "I2", "merged_LnOR", "merged_LnOR_VAR" or "merged_LnOR_p". And the value must be simulation data. 'Qp', "I2", "merged_LnOR", "merged_LnOR_VAR" and "merged_LnOR_p" separetly plots the return value 'perm_Qp', 'perm_I2', 'perm_merged_LnOR', 'perm_merged_VARLnOR', 'perm_merged_p'. |
| `fill_col` | the filled color(default NULL) for the body of histplot. |
| `border_col` | the color(default 'black') for the border of histplot. |
| `arrows_col` | the color(default 'red') of arrows which mark the place of the observed value. |
| `main` | the main title (on top), default value is "Hist plot for heterogeneity Q p_vlaue". |
| `xlab,ylab` | X axis label, default value is 'Q p_value'. Y axis label, default value is 'Density'. |
| `digits` | integer(default 3) indicating the number of decimal places. |

## Details

Histplot for the return value('perm_Qp','perm_I2','perm_merged_LnOR','perm_merged_VARLnOR', 'perm_merged_p') of 'meta.MCPerm' or 'meta.TradPerm'. And through arrows and legend to mark the observed value.

The symbols in the legend: 'Q_stat' is the Q statistic for meta data heterogeneity; 'Q_p' is the p value of Q value(chi square distribution,the number of studies in meta analysis minus one is degree of freedom of Q value.); 'p.corrected' is the corrected p value by permutation; 'I2_stat' is the statistic I2(calculated by formula max(Q-d.f./Q, 0)) for meta data heterogeneity; 'merged_LnOR' is the merged log odd ratio of observed data; 'merged_LnOR_VAR' is the variance of log odd ratio for observed data; 'merged_LnOR_p' is the p value of log odd ratio of observed data which obey normal distribution.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

### Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

### See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Genotype.Hist, VS.Allele.Hist, PermMeta.LnOR.Hist, PermMeta.LnOR.CDC, PermMeta.LnOR.boxplot, PermMeta.LnOR.qqnorm, PermMeta.boxplot

### Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
## set working directory to save the plots.
# setwd("D:\")
# pdf("PermMeta.Hist.pdf",height=6,width=6)
# PermMeta.Hist(result,plot="Qp",fill_col=NULL,border_col='black',
    # arrows_col='red',main="Hist plot for heterogeneity Q p_vlaue",xlab="Q p_value")
# PermMeta.Hist(result,plot="I2",fill_col=NULL,border_col='black',
    # arrows_col='red',main="Hist plot for heterogeneity I2",xlab="I2")
# PermMeta.Hist(result,plot="merged_LnOR",fill_col=NULL,border_col='black',
    # arrows_col='red',main="Hist plot for merged_LnOR",xlab="merged_LnOR")
# PermMeta.Hist(result,plot="merged_LnOR_VAR",fill_col=NULL,border_col='black',
    # arrows_col='red',main="Hist plot for merged_LnOR_VAR",xlab="merged_LnOR_Variance")
# PermMeta.Hist(result,plot="merged_LnOR_p",fill_col=NULL,border_col='black',
    # arrows_col='red',main="Hist plot for merged_LnOR_p",xlab="merged_LnOR_p.value")
# dev.off()
```

---

## PermMeta.LnOR.boxplot *boxplot for the return value 'perm_LnOR' or 'perm_VARLnOR' of 'meta.MCPerm' or 'meta.TradPerm'*

---

### Description

boxplot for the return value 'perm_LnOR' or 'perm_VARLnOR' of 'meta.MCPerm' or 'meta.TradPerm'.

### Usage

```
PermMeta.LnOR.boxplot(PermMeta, plot = "LnOR", plot_study = "all", order = "no",
    main = "LnOR,no order", true_value_pch = 3, pch_col = "red",
 pos = 3, text_col = "blue", digits = 2)
```

## Arguments

| | |
|---|---|
| PermMeta | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| plot | a character string indicating which return value of function 'meta.TradPerm' or 'meta.MCPerm' to be plot. The value can be "LnOR"(default), "LnOR_VAR". "LnOR" and "LnOR_VAR" separetly plots the return value 'perm_LnOR' and 'perm_VARLnOR'. |
| plot_study | a numeric vector indicates which study(ies) in meta analysis to be plotted. Default value is 'all', which indicates all studies in meta analysis to be plotted. |
| order | a character string specifying the boxplot ascending order. The order can be 'LnOR', 'LnOR_VAR', 'VAR_LnOR', 'sample' and 'no'(default). See details. |
| main | the main title (on top), default value is 'LnOR,no order'. |
| true_value_pch | the pch(default 3) to mark the observed value in the boxplot of simulation data. |
| pch_col | the color of pch, default value is 'red'. |
| pos | a position specifier for text of the observed value around the pch.Values of 1, 2, 3(default) and 4, respectively indicate positions below, to the left of, above and to the right of the specified coordinates. |
| text_col | the color(default 'blue') for the text of the observed value. |
| digits | integer(default 2) indicating the number of decimal places. |

## Details

Boxplot for the return value('perm_LnOR','perm_VARLnOR') of 'meta.MCPerm' or 'meta.TradPerm'. And through 'pch' and text to mark the observed value. The boxplot can order by 'LnOR', 'LnOR_VAR', 'VAR_LnOR', 'sample' and 'no'.

'LnOR' indicates the boxplot order by the log odd ratio calculated by each study observed genotype count. 'LnOR_VAR' indicates the boxplot order by the variance of log odd ratio calculated by formule 1/ai+1/bi+1/ci+1/di(ai/bi/ci/di separetly are case/control two alleles count). 'VAR_LnOR' indicates the variance of simulation log odd ratios. 'sample' indicates the boxplot order by the sample size of each study. 'no' indicates that the boxplot order is same as the order of parameter 'plot_study'.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Value

| | |
|---|---|
| plot | character value of parameter 'plot'. |
| plot_num | the number of the selected studies to be plotted. |
| plot_order | character value of parmeter 'order'. |
| order_index | a numeric vector of the order of studies. |
| true_LnOR | a numeric vector of log odd ratio calculated by observed value of each study. And the order is corresponding to the 'order_index'. |
| LnOR_VAR | a numeric vector of the log odd ratio variance(1/ai+1/bi+1/ci+1/di) calculated by observed value of each study. And the order is corresponding to the 'order_index'. |

| | |
|---|---|
| VAR_LnOR | a numeric vector of the variance of the simulated log odd ratios got by permutation. And the order is corresponding to the 'order_index'. |
| sample | a numeric vector of the sample size of the plotted studies, and the order is corresponding to the 'order_index'. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Genotype.Hist, VS.Allele.Hist, PermMeta.LnOR.Hist, PermMeta.LnOR.CDC, PermMeta.Hist, PermMeta.boxplot

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
## set working directory to save the plots.
# setwd("D:\")
# pdf("PermMeta.LnOR.boxplot.pdf",height=6,width=9)
# par(mfrow=c(2,3))
# PermMeta.LnOR.boxplot(result,plot="LnOR",plot_study="all",order="no")
# PermMeta.LnOR.boxplot(result,plot="LnOR",plot_study="all",order="LnOR",
    # main="LnOR, order by LnOR")
# PermMeta.LnOR.boxplot(result,plot="LnOR",plot_study="all",order="LnOR_VAR",
    # main="LnOR, order by LnOR_VAR")
# PermMeta.LnOR.boxplot(result,plot="LnOR",plot_study="all",order="VAR_LnOR",
    # main="LnOR, order by VAR_LnOR")
# PermMeta.LnOR.boxplot(result,plot="LnOR",plot_study="all",order="sample",
    # main="LnOR, order by sample")

# PermMeta.LnOR.boxplot(result,plot="LnOR_VAR",plot_study=c(1,4,14,12),order="no",
    # main="LnOR_VAR, no order")
# PermMeta.LnOR.boxplot(result,plot="LnOR_VAR",plot_study=c(1,4,14,12),order="LnOR",
    # main="LnOR_VAR, order by LnOR")
# PermMeta.LnOR.boxplot(result,plot="LnOR_VAR",plot_study=c(1,4,14,12),order="LnOR_VAR",
    # main="LnOR_VAR, order by LnOR_VAR")
# PermMeta.LnOR.boxplot(result,plot="LnOR_VAR",plot_study=c(1,4,14,12),order="VAR_LnOR",
    # main="LnOR_VAR, order by VAR_LnOR")
# PermMeta.LnOR.boxplot(result,plot="LnOR_VAR",plot_study=c(1,4,14,12),order="sample",
    # main="LnOR_VAR, order by sample")
# dev.off()
```

PermMeta.LnOR.CDC *cumulative distribution curve for the return value 'perm_LnOR' of 'meta.MCPerm' or 'meta.TradPerm'*

## Description

cumulative distribution curve for the return value 'perm_LnOR' of 'meta.MCPerm' or 'meta.TradPerm'.

## Usage

```
PermMeta.LnOR.CDC(PermMeta, plot_study = "all", nrow = 2, ncol = 3,
    PermMeta.LnOR_pch = 4, PermMeta.LnOR_col = "black",
 LnOR_VAR_pch = 18, LnOR_VAR_col = "blue",
 VAR_LnOR_pch = 18, VAR_LnOR_col = "red",
 main = "cumulative distribution curve for LnOR", title = NULL,
 xlab = "LnOR", ylab = "cumulative probability", digits = 3)
```

## Arguments

| | |
|---|---|
| PermMeta | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| plot_study | a numeric vector indicates which study(ies) in meta analysis to be plotted. Default value is 'all', which indicates all studies in meta analysis to be plotted. |
| nrow,ncol | positive integer, divides the device up into 'nrow'(default is 2) rows and 'ncol'(default is 3) columns. |
| PermMeta.LnOR_pch,PermMeta.LnOR_col | the pch(default 4) and the color(default 'black') of pch are for the cumulative distribution curve of the return value 'perm_LnOR' of certain study. |
| LnOR_VAR_pch,LnOR_VAR_col | the pch(default 18) and the color*(default 'blue') of pch are for the cumulative distribution curve of the normal distrition with mean=0 and variance=1/ai+1/bi+1/ci+1/di. |
| VAR_LnOR_pch,VAR_LnOR_col | the pch(default 18) and the color(default 'red') of pch are for the cumulative distribution curve of the normal distrition with mean=0 and variance is the variance of simulation log odd ratios. |
| main | the main title(on top), default value is "cumulative distribution curve for LnOR". |
| title | the sub main title for each plotted study(on top). |
| xlab,ylab | X axis label, default value is 'LnOR'. Y axis label, default value is 'cumulative probability'. |
| digits | integer(default 3) indicating the number of decimal places. |

## Details

Plot three cumulative distribution cures(abbreviation:CDC): 1) CDC for simulative log odd ratios; 2) CDC for normal distribution with mean=0 and var=variance of observed log odd ratio of certain

study(1/ai+1/bi+1/ci+1/di); 3) CDC for normal distribution with mean=0 and var=variance of simulative log odd ratios. The symbol—'perm_lnOR','pnorm_LnOR_VAR','pnorm_VAR_LnOR' in the topright legend separately indicated the first, second, third cure. Through three CDC compared, observe than the thrid cure is more corresponding to the first cure when sample size is smaller.

The symbol in the bottomright legend: 'LnOR' indicates the log odd ratio of observed data for the study; 'sample' indicates the sample size of the study; 'LnOR_VAR' indicates the variance of second cure; 'VAR_LnOR' indicates the variance of third cure.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

## Value

| | |
|---|---|
| `plot_study` | the value of paramter 'plot_study'. |
| `LnOR` | the numeric vector of log odd ratio of the observed data for the plotted studies. |
| `sample` | the numeric vector of sample size of the plotted studies. |
| `LnOR_VAR` | the numeric vector of variance of the sencond cure for the plotted studies. |
| `VAR_LnOR` | the numeric vector of variance of the third cure for the plotted studies. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.CDC`, `VS.KS`, `VS.Genotype.CDC`, `VS.Allele.CDC`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.qqnorm`, `PermMeta.LnOR.boxplot`, `PermMeta.Hist`, `PermMeta.boxplot`

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
# PermMeta.LnOR.CDC(result,plot_study=c(3,5,21,7,12,9),nrow=2,ncol=3)
```

---

# PermMeta.LnOR.Hist *histplot for the return value 'perm_LnOR' or 'perm_VARLnOR' of 'meta.MCPerm' or 'meta.TradPerm'*

---

## Description

histplot for the return value 'perm_LnOR' or 'perm_VARLnOR' of 'meta.MCPerm' or 'meta.TradPerm'

## Usage

```
PermMeta.LnOR.Hist(PermMeta, plot = "LnOR", plot_study = "all", nrow = 2, ncol = 2,
    main = "Background distribution for LnOR", title = NULL,
 xlab = "LnOR", hist_border_col = "black", arrows_col = "red", digits = 3)
```

## Arguments

| | |
|---|---|
| PermMeta | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| plot | a character string indicating which return value of function 'meta.TradPerm' or 'meta.MCPerm' to be plot. The value can be "LnOR", "LnOR_VAR". "LnOR" and "LnOR_VAR" separetly plots the return value 'perm_LnOR' and 'perm_VARLnOR'. Default value is 'LnOR'. |
| plot_study | a numeric vector indicates which study(ies) in meta analysis to be plotted. Default value is 'all', which indicates all studies in meta analysis to be plotted. |
| nrow,ncol | positive integer, divides the device up into 'nrow' rows and 'ncol' columns. Default value is 2. |
| main | the main title (on top), default value is "Background distribution for LnOR". |
| title | the sub main title for each plotted study(on top). |
| xlab | X axis label, default value is 'LnOR'. |
| hist_border_col | the color for the border of histplot. Default value is 'black'. |
| arrows_col | the col of arrows which mark the place of the observed value. Default value is 'red'. |
| digits | integer indicating the number of decimal places. Default value is 3. |

## Details

Histplot for the return value('perm_LnOR','perm_VARLnOR') of 'meta.MCPerm' or 'meta.TradPerm'. And through arrows and legend to mark the observed value.

The symbol in the legend: 'LnOR' indicates the log odd ratio of observed value for the study. 'LnOR_VAR' indicates the variance of log odd ratio calculated by formule 1/ai+1/bi+1/ci+1/di (ai/bi/ci/di separetly are case/control two alleles count). 'sample' indicates the sample size of the study. 'mean' indicates the mean value of simulative log odd ratios. 'var' indicates the variance of simulative log odd ratios. 'p' is the percentage of more than the observed value.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Value

| | |
|---|---|
| plot_study | the value of paramter 'plot_study'. |
| LnOR | the numeric vector of log odd ratio of the observed data for the plotted studies. |
| sample | the numeric vector of sample size of the plotted studies. |
| LnOR_VAR | the numeric vector of variance of the log odd ratio of obaserved value for the plotted studies. |
| VAR_LnOR | the numeric vector of variance of simulative log odd ratios for the plotted studies. |

### Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

### See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Genotype.Hist, VS.Allele.Hist, PermMeta.LnOR.qqnorm, PermMeta.LnOR.CDC, PermMeta.LnOR.boxplot, PermMeta.Hist, PermMeta.boxplot

### Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
# PermMeta.LnOR.Hist(result,plot="LnOR",plot_study="all",nrow=2,ncol=2,
    # title=NULL,xlab="LnOR")
# PermMeta.LnOR.Hist(result,plot="LnOR_VAR",plot_study=c(3,5,12,21),nrow=2,ncol=2,
    # main="Background distribution for LnOR_VAR",title=NULL,xlab="LnOR_VAR")
```

---

## PermMeta.LnOR.qqnorm *qqnorm plot for the return value 'perm_LnOR' of 'meta.MCPerm' or 'meta.TradPerm'*

---

### Description

qqnorm plot for the return value 'perm_LnOR' of 'meta.MCPerm' or 'meta.TradPerm'.

### Usage

```
PermMeta.LnOR.qqnorm(PermMeta, plot_study = "all", nrow = 2, ncol = 2,
    main = "qqnorm plot for LnOR", title = NULL,
 xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
 scatter_col = "black", line_col = "red")
```

### Arguments

| | |
|---|---|
| PermMeta | the result of function 'meta.TradPerm' or 'meta.MCPerm'. |
| plot_study | a numeric vector indicates which study(ies) in meta analysis to be plotted. Default value is 'all', which indicates all studies in meta analysis to be plotted. |
| nrow,ncol | positive integer, divides the device up into 'nrow' rows and 'ncol' columns. Default value is 2. |
| main | the main title (on top), default value is "qqnorm plot for LnOR". |

| | |
|---|---|
| title | the sub main title for each plotted study(on top). |
| xlab,ylab | X axis label, default value is "Theoretical Quantiles". Y axis label, default value is "Sample Quantiles". |
| scatter_col | the color of the scatter points. Default value is 'black'. |
| line_col | the color of the line which passes through the normal distribution probs quantiles, the first and third quartiles. Default value is 'red'. |

## Details

Plotting a normal QQ plot for simulative log odd ratios is to test that simulative data is whether fit normal distribution, Snd plot a line which passes through the normal distribution probs quantiles, the first and third quartiles.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.QQ`, `VS.KS`, `VS.Genotype.QQ`, `VS.Allele.QQ`, `PermMeta.LnOR.Hist`, `PermMeta.LnOR.CDC`, `PermMeta.LnOR.boxplot`, `PermMeta.Hist`, `PermMeta.boxplot`

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete first line
# temp=MetaGenotypeCount[-1,];
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",fixed_method="MH",random_method="DL",repeatNum=1000)
# PermMeta.LnOR.qqnorm(result,plot_study=c(1,4,9,15),nrow=2,ncol=2)
```

---

`permuteGenotype` *Permute the elements of genotype data*

---

## Description

Permute the elements of genotype data.

## Usage

```
permuteGenotype(dataLine, fromCol)
```

### Arguments

| | |
|---|---|
| dataLine | A matrix with one row. |
| fromCol | A positive integer, the start column to permute. |

### Details

Permute the elements of genotype data.

### Value

The return value is a matrix with one row and the elements has been permutated.

### Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

### References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

### See Also

Armitage.TradPerm, OR.TradPerm, chisq.TradPerm, fisher.TradPerm, meta.TradPerm, permuteGenotypeCount, chisq.MCPerm

### Examples

```
# dataLine1=matrix(c("rs12","AA","AG","GG","CG","AA"),nrow=1)
# permuteGenotype(dataLine1,fromCol=2)
# dataLine2=matrix(c("rs12","AA","AG","GG","CG","AA",
   # "rs12","AA","AG","GG","CG","AA"),nrow=2,byrow=TRUE)
# permuteGenotype(dataLine2,fromCol=2)
```

---

## permuteGenotypeCount *Fill the numerics of 2*3 table when fixed the row and column totals*

---

### Description

Fill the numerics of 2*3 table when fixed the row and column totals.

### Usage

```
permuteGenotypeCount(case_11, case_12, case_22, control_11, control_12, control_22, n)
```

## Arguments

| | |
|---|---|
| case_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| case_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| case_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| control_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| control_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| control_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |
| n | an integer specifying the number to generate. |

## Details

Fill the numerics of 2*3 table when fixed the row and column totals, by twice generating random numbers for the hypergeometric distribution, based on the putting data.

## Value

| | |
|---|---|
| perm_case_11 | a vector with 'n' elements. |
| perm_case_12 | a vector with 'n' elements. |
| perm_case_22 | a vector with 'n' elements. |
| perm_control_11 | a vector with 'n' elements. |
| perm_control_12 | a vector with 'n' elements. |
| perm_control_22 | a vector with 'n' elements. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

Armitage.MCPerm, OR.MCPerm, chisq.MCPerm, fisher.MCPerm, rhyper, meta.MCPerm, permuteGenotype, chisq.TradPerm

## Examples

```
# permuteGenotypeCount(case_11=23,case_12=0,case_22=34,control_11=0,control_12=34,control_22=45,n=5)
```

---

`print.PermMeta` *Print style for function 'meta.MCPerm' or 'meta.TradPerm'*

---

## Description

Print style for function 'meta.MCPerm' or 'meta.TradPerm'.

## Usage

```
## S3 method for class 'PermMeta'
print(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | the result of function 'meta.MCPerm' or 'meta.TradPerm'. |
| `...` | other arguments. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## See Also

`meta.MCPerm`, `meta.TradPerm`

---

`Q.MCPerm` *Correct p.value for Heterogeneity statistics Q in meta analysis by MCPerm method.*

---

## Description

Correct p.value for Heterogeneity statistics Q in meta analysis by MCPerm method.

## Usage

```
Q.MCPerm(case_11, case_12, case_22, control_11, control_12, control_22,
    model = "allele", method = "MH", repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| case_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in case samples. |
| case_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in case samples. |
| case_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in case samples. |
| control_11 | a non-negative integer, the frequency of genotype "allele1/allele1" in control samples. |
| control_12 | a non-negative integer, the frequency of genotype "allele1/allele2" in control samples. |
| control_22 | a non-negative integer, the frequency of genotype "allele2/allele2" in control samples. |
| model | a character string indicating the type of model("allele","dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. Default value is 'allele'. |
| method | a character string indicating the method('Inverse','MH' or 'Peto') to calculate Q value. Default value is 'MH'. See details. |
| repeatNum | an integer specifying the number of replicates used in the Monte Carlo permutation. Default value is 1000. |

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

Q value fit chi square distribution,the number of studies in meta analysis minus one is degree of freedom of Q value.

MCPerm details see chisq.MCPerm.

## Value

| | |
|---|---|
| risk_allele | the symbol of risk allele.See details. |
| Q | the Q statistics for observed meta data. |
| Qp | the p value for Q,See details. |
| corrected_Qp | the percentage of less than Qp value in simulative Qp values. |

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Julian P.T.Higgins, Simon G.Thompson(Statistics in Medicine,2002): Quantifying heterogeneity in a meta-analysis.

Julian P.T.Higgins, Simon G.Thompson, Jonathan J Deeks(BMJ,2003):Measuring inconsistency in meta-analyses.

## See Also

meta.MCPerm, meta.TradPerm, Q.TradPerm, I2.TradPerm, I2.MCPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Allele.Hist, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=Q.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result
```

---

# Q.TradPerm *Correct p.value for Heterogeneity statistics Q in meta analysis by TradPerm method*

---

## Description

Correct p.value for Heterogeneity statistics Q in meta analysis by TradPerm method.

## Usage

```
Q.TradPerm(genotypeData, affectionData, split, sep, naString, model = "allele",
    method = "MH", repeatNum = 1000)
```

## Arguments

| | |
|---|---|
| genotypeData | a matrix with one column and multiple rows, each row contains genotype data for case and control samples of certain study. Note the field separtor of each line must be same, and same with parameter 'affectionData'. |
| affectionData | a matrix with one column and multiple rows, each row contains the affection stats of case and control samples of certain study which must correspond to 'genotypeData'. Note the field separtor of each line must be same,and same with parameter 'genotypeData'. |
| split | the field separator character, which separates elements on each line of the parameter 'genotypeData' and 'affectionData'. 'Split' and 'sep' cannot be same. |

| | |
|---|---|
| sep | character separator used to divide genotype between alleles "Allele1<sep>Allele2" in parameter 'genotypeData'. 'Split' and 'sep' cannot be same. |
| naString | a character string for NA values of genotype data in parameter 'genotypeData'. |
| model | a character string indicating the type of model("allele","dominant" or "recessive") supplied to the data. The risk allele(see details) is marked as allele1. The allele model indicates allele1 versus allele2, the dominant model indicates <allele1/allele1> + <allele1/allele2> versus <allele2/allele2>, the recessive model indicates <allele1/allele1> versus <allele1/allele2> + <allele2/allele2>. Default value is 'allele'. |
| method | a character string indicating the method('Inverse','MH' or 'Peto') to calculate Q value. Default value is 'MH'. See details. |
| repeatNum | an integer specifying the number of replicates used in the Monte Carlo permutation. Default value is 1000. |

## Details

Allele 1 and allele 2 to each study have OR values. The risk allele is the allele which the number of studies which OR>1 more than half of the number of all studies.

Q value fit chi square distribution,the number of studies in meta analysis minus one is degree of freedom of Q value.

TradPerm details see chisq.TradPerm.

## Value

| | |
|---|---|
| risk_allele | the symbol of risk allele.See details. |
| Q | the Q statistics for observed meta data. |
| Qp | the p value for Q,See details. |
| corrected_Qp | the percentage of less than Qp value in simulative Qp values. |

## Note

'Split':the field separtor of each line for parameter 'genotypeData' and 'affectionData' must be same. 'Split' and 'sep' cannot be same.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

Julian P.T.Higgins, Simon G.Thompson(Statistics in Medicine,2002): Quantifying heterogeneity in a meta-analysis.

Julian P.T.Higgins, Simon G.Thompson, Jonathan J Deeks(BMJ,2003):Measuring inconsistency in meta-analyses.

## See Also

meta.MCPerm, meta.TradPerm, Q.MCPerm, I2.TradPerm, I2.MCPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Allele.Hist, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result=Q.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result
```

---

# VS.Allele.CDC

*separately plot cumulative distribution curve for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

separately plot cumulative distribution curve for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study.

## Usage

```
VS.Allele.CDC(Trad_case_1, Trad_case_2, Trad_control_1, Trad_control_2,
    MC_case_1, MC_case_2, MC_control_1, MC_control_2,
 Trad_col = "black", MC_col = "red",
    main = "cumulative distribution curve",
 title = c("case_A", "case_a", "control_A", "control_a"),
 xlab = "count", ylab = "cumulative probability")
```

## Arguments

| | |
|---|---|
| Trad_case_1 | a numeric vector, simulative allele 1 count for case samples got by TradPerm method for certain study. |

| | |
|---|---|
| Trad_case_2 | a numeric vector, simulative allele 2 count for case samples got by TradPerm method for certain study. |
| Trad_control_1 | a numeric vector, simulative allele 1 count for control samples got by TradPerm method for certain study. |
| Trad_control_2 | a numeric vector, simulative allele 2 count for control samples got by TradPerm method for certain study. |
| MC_case_1 | a numeric vector, simulative allele 1 count for case samples got by MCPerm method for certain study. |
| MC_case_2 | a numeric vector, simulative allele 2 count for case samples got by MCPerm method for certain study. |
| MC_control_1 | a numeric vector, simulative allele 1 count for control samples got by MCPerm method for certain study. |
| MC_control_2 | a numeric vector, simulative allele 2 count for control samples got by MCPerm method for certain study. |
| Trad_col | the color of cumulative distribution cure for Trad_case_1/Trad_case_2/ Trad_control_1/Trad_control_2. Default value is 'black'. |
| MC_col | the color of cumulative distribution cure for MC_case_1/MC_case_2/ MC_control_1/MC_control_2. Default value is 'red'. |
| main | the main title(on top), default value is "cumulative distribution curve". |
| title | the sub main title for each plot(on top). Default value is a vector with elements: 'case_A', 'case_a', 'control_A' and 'control_a'. |
| xlab,ylab | X axis label, default value is 'count'. Y axis label, default value is 'cumulative probability'. |

## Details

Separately plotting cumulative distribution curve for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative allele count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.QQ`, `VS.KS`, `VS.Allele.Hist`, `VS.Allele.QQ`, `VS.Genotype.CDC`, `PermMeta.LnOR.CDC`

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]
# Trad_case_2=2*result1$perm_case_22[12,]+result1$perm_case_12[12,]
# Trad_control_1=2*result1$perm_control_11[12,]+result1$perm_control_12[12,]
# Trad_control_2=2*result1$perm_control_22[12,]+result1$perm_control_12[12,]

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]
# MC_case_2=2*result2$perm_case_22[12,]+result2$perm_case_12[12,]
# MC_control_1=2*result2$perm_control_11[12,]+result2$perm_control_12[12,]
# MC_control_2=2*result2$perm_control_22[12,]+result2$perm_control_12[12,]

# VS.Allele.CDC(Trad_case_1,Trad_case_2,Trad_control_1,Trad_control_2,
    # MC_case_1,MC_case_2,MC_control_1,MC_control_2,
    # main="cumulative distribution curve for allele model",
 # title=c("case_A","case_a","control_A","control_a"))
```

---

`VS.Allele.Hist` *separately plot histplot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

separately plot histplot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study

## Usage

```
VS.Allele.Hist(Trad_case_1, Trad_case_2, Trad_control_1, Trad_control_2,
    MC_case_1, MC_case_2, MC_control_1, MC_control_2,
 Trad_col = "grey", MC_col = "black",
 main = "distribution for allele frequency",
 title = c("case_A", "case_a", "control_A", "control_a"), xlab = "count")
```

## Arguments

| | |
|---|---|
| Trad_case_1 | a numeric vector, simulative allele 1 count for case samples got by TradPerm method for certain study. |
| Trad_case_2 | a numeric vector, simulative allele 2 count for case samples got by TradPerm method for certain study. |
| Trad_control_1 | a numeric vector, simulative allele 1 count for control samples got by TradPerm method for certain study. |
| Trad_control_2 | a numeric vector, simulative allele 2 count for control samples got by TradPerm method for certain study. |
| MC_case_1 | a numeric vector, simulative allele 1 count for case samples got by MCPerm method for certain study. |
| MC_case_2 | a numeric vector, simulative allele 2 count for case samples got by MCPerm method for certain study. |
| MC_control_1 | a numeric vector, simulative allele 1 count for control samples got by MCPerm method for certain study. |
| MC_control_2 | a numeric vector, simulative allele 2 count for control samples got by MCPerm method for certain study. |
| Trad_col | the color of cumulative distribution cure for Trad_case_1/Trad_case_2/Trad_control_1/ Trad_control_2. Default value is 'grey'. |
| MC_col | the color of cumulative distribution cure for MC_case_1/MC_case_2/MC_control_1/ MC_control_2. Default value is 'black'. |
| main | the main title(on top). Default value is "distribution for allele frequency". |
| title | the sub main title for each plot(on top). Default value is a vector with elements: 'case_A', 'case_a', 'control_A' and 'control_a'. |
| xlab | X axis label. Default value is 'count'. |

## Details

Separately plotting histplot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative allele count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.QQ, VS.KS, VS.Allele.QQ, VS.Allele.CDC, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]
# Trad_case_2=2*result1$perm_case_22[12,]+result1$perm_case_12[12,]
# Trad_control_1=2*result1$perm_control_11[12,]+result1$perm_control_12[12,]
# Trad_control_2=2*result1$perm_control_22[12,]+result1$perm_control_12[12,]

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]
# MC_case_2=2*result2$perm_case_22[12,]+result2$perm_case_12[12,]
```

```
# MC_control_1=2*result2$perm_control_11[12,]+result2$perm_control_12[12,]
# MC_control_2=2*result2$perm_control_22[12,]+result2$perm_control_12[12,]

# VS.Allele.Hist(Trad_case_1,Trad_case_2,Trad_control_1,Trad_control_2,
    # MC_case_1,MC_case_2,MC_control_1,MC_control_2,
    # main="cumulative distribution curve for allele model",
 # title=c("case_A","case_a","control_A","control_a"))
```

---

`VS.Allele.QQ` *separately plot quantile-quantile plot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

separately plot quantile-quantile plot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study.

## Usage

```
VS.Allele.QQ(Trad_case_1, Trad_case_2, Trad_control_1, Trad_control_2,
    MC_case_1, MC_case_2, MC_control_1, MC_control_2,
 scatter_col = "black", line_col = "black",
 main = "QQ plot for allele model",
 title = c("case_A", "case_a", "control_A", "control_a"),
 xlab = "Quantile of count (TradPerm)", ylab = "Quantile of count (MCPerm)")
```

## Arguments

| | |
|---|---|
| `Trad_case_1` | a numeric vector, simulative allele 1 count for case samples got by TradPerm method for certain study. |
| `Trad_case_2` | a numeric vector, simulative allele 2 count for case samples got by TradPerm method for certain study. |
| `Trad_control_1` | a numeric vector, simulative allele 1 count for control samples got by TradPerm method for certain study. |
| `Trad_control_2` | a numeric vector, simulative allele 2 count for control samples got by TradPerm method for certain study. |
| `MC_case_1` | a numeric vector, simulative allele 1 count for case samples got by MCPerm method for certain study. |
| `MC_case_2` | a numeric vector, simulative allele 2 count for case samples got by MCPerm method for certain study. |
| `MC_control_1` | a numeric vector, simulative allele 1 count for control samples got by MCPerm method for certain study. |
| `MC_control_2` | a numeric vector, simulative allele 2 count for control samples got by MCPerm method for certain study. |
| `scatter_col` | the color for scatter points of quantile-quantile plot. Default value is 'black'. |

| | |
|---|---|
| line_col | the color of line which passes through the sample distribution probs quantiles, the first and third quartiles. Default value is 'black'. |
| main | the main title(on top). Default value is "QQ plot for allele model". |
| title | the sub main title for each plot(on top). Default value is a vector with elements: 'case_A', 'case_a', 'control_A' and 'control_a'. |
| xlab,ylab | X axis label, default value is "Quantile of count (TradPerm)". Y axis label, default value is "Quantile of count (MCPerm)". |

## Details

Separately plotting quantile-quantile plot for the return value(allele count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative allele count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.QQ, VS.KS, VS.Allele.Hist, VS.Allele.CDC, VS.Genotype.QQ, PermMeta.LnOR.qqnorm

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
```

```
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]
# Trad_case_2=2*result1$perm_case_22[12,]+result1$perm_case_12[12,]
# Trad_control_1=2*result1$perm_control_11[12,]+result1$perm_control_12[12,]
# Trad_control_2=2*result1$perm_control_22[12,]+result1$perm_control_12[12,]

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]
# MC_case_2=2*result2$perm_case_22[12,]+result2$perm_case_12[12,]
# MC_control_1=2*result2$perm_control_11[12,]+result2$perm_control_12[12,]
# MC_control_2=2*result2$perm_control_22[12,]+result2$perm_control_12[12,]

# VS.Allele.QQ(Trad_case_1,Trad_case_2,Trad_control_1,Trad_control_2,
    # MC_case_1,MC_case_2,MC_control_1,MC_control_2,
    # main="cumulative distribution curve for allele model",
 # title=c("case_A","case_a","control_A","control_a"))
```

---

VS.CDC *plot cumulative distribution curve for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis*

---

## Description

plot cumulative distribution curve for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis

## Usage

```
VS.CDC(Trad_data, MC_data, Trad_col = "black", MC_col = "red",
    title = NULL, xlab = NULL, ylab = "cumulative probability")
```

## Arguments

| | |
|---|---|
| Trad_data | the return value of function 'meta.TradPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |
| MC_data | the return value of function 'meta.MCPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |
| Trad_col | the color for cumulative distribution curve of 'Trad_data'. Default value is 'black'. |

| | |
|---|---|
| MC_col | the color for cumulative distribution curve of 'MC_data'. Default value is 'red'. |
| title | the main title(on top). |
| xlab,ylab | X axis label. Y axis label, default value is 'cumulative probability'. |

## Details

Plotting cumulative distribution curve for the return value(e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc) of 'meta.TradPerm' and 'meta.MCPerm' is to compare the simulative data distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.QQ, VS.KS, VS.Allele.Hist, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]

## import data
# data(MetaGenotypeCount)
```

```
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]

# VS.CDC(Trad_case_1,MC_case_1,title="cumulative distribution cure for case_1")
# VS.CDC(result1$perm_Qp,result2$perm_Qp,title="cumulative distribution cure for Qp")
# VS.CDC(result1$perm_p,result2$perm_p,title="cumulative distribution cure for p")
```

---

`VS.Genotype.CDC` *separately plot cumulative distribution curve for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

Separately plot cumulative distribution curve for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study.

## Usage

```
VS.Genotype.CDC(Trad_case_11, Trad_case_12, Trad_case_22,
    Trad_control_11, Trad_control_12, Trad_control_22,
 MC_case_11, MC_case_12, MC_case_22,
 MC_control_11, MC_control_12, MC_control_22,
 Trad_col = "black", MC_col = "red",
 title = NULL, xlab = "Genotype count", ylab = "cumulative probability")
```

## Arguments

| | |
|---|---|
| `Trad_case_11` | a numeric vector, the return value 'perm_case_11' got by meta.TradPerm method for certain study. |
| `Trad_case_12` | a numeric vector, the return value 'perm_case_12' got by meta.TradPerm method for certain study. |
| `Trad_case_22` | a numeric vector, the return value 'perm_case_22' got by meta.TradPerm method for certain study. |
| `Trad_control_11` | a numeric vector, the return value 'perm_control_11' got by meta.TradPerm method for certain study. |
| `Trad_control_12` | a numeric vector, the return value 'perm_control_12' got by meta.TradPerm method for certain study. |

`Trad_control_22`
: a numeric vector, the return value 'perm_control_22' got by meta.TradPerm method for certain study.

`MC_case_11`
: a numeric vector, the return value 'perm_case_11' got by meta.MCPerm method for certain study.

`MC_case_12`
: a numeric vector, the return value 'perm_case_12' got by meta.MCPerm method for certain study.

`MC_case_22`
: a numeric vector, the return value 'perm_case_22' got by meta.MCPerm method for certain study.

`MC_control_11`
: a numeric vector, the return value 'perm_control_11' got by meta.MCPerm method for certain study.

`MC_control_12`
: a numeric vector, the return value 'perm_control_12' got by meta.MCPerm method for certain study.

`MC_control_22`
: a numeric vector, the return value 'perm_control_22' got by meta.MCPerm method for certain study.

`Trad_col`
: the color of cumulative distribution cure for Trad_case_11/Trad_case_12/Trad_case_22/ Trad_control_11/Trad_control_12/Trad_control_22. Default value is 'black'.

`MC_col`
: the color of cumulative distribution cure for MC_case_11/MC_case_12/MC_case_22/ MC_control_11/MC_control_12/MC_control_22. Default value is 'red'.

`title`
: the main title(on top).

`xlab,ylab`
: X axis label, default value is "Genotype count". Y axis label, default value is "cumulative probability".

## Details

Separately plotting cumulative distribution curve for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative genotype count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see `chisq.MCPerm`. TradPerm details see `chisq.TradPerm`.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

`meta.MCPerm`, `meta.TradPerm`, `chisq.MCPerm`, `chisq.TradPerm`, `VS.CDC`, `VS.KS`, `VS.Genotype.Hist`, `VS.Genotype.QQ`, `VS.Allele.CDC`, `PermMeta.LnOR.CDC`

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2

## plot study 12
# VS.Genotype.CDC(result1$perm_case_11[12,],result1$perm_case_12[12,],result1$perm_case_22[12,],
   # result1$perm_control_11[12,],result1$perm_control_12[12,],result1$perm_control_22[12,],
 # result2$perm_case_11[12,],result2$perm_case_12[12,],result2$perm_case_22[12,],
 # result2$perm_control_11[12,],result2$perm_control_12[12,],result2$perm_control_22[12,],
    # Trad_col="grey",MC_col="black", title="hist_plot for six genotype")
```

---

`VS.Genotype.Hist` *separately plot histplot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

Separately plot histplot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study.

## Usage

```
VS.Genotype.Hist(Trad_case_11, Trad_case_12, Trad_case_22,
```

```
   Trad_control_11, Trad_control_12, Trad_control_22,
MC_case_11, MC_case_12, MC_case_22,
MC_control_11, MC_control_12, MC_control_22,
Trad_col = "grey", MC_col = "black",
title = NULL, xlab = "Genotype count")
```

## Arguments

| | |
|---|---|
| Trad_case_11 | a numeric vector, the return value 'perm_case_11' got by meta.TradPerm method for certain study. |
| Trad_case_12 | a numeric vector, the return value 'perm_case_12' got by meta.TradPerm method for certain study. |
| Trad_case_22 | a numeric vector, the return value 'perm_case_22' got by meta.TradPerm method for certain study. |
| Trad_control_11 | a numeric vector, the return value 'perm_control_11' got by meta.TradPerm method for certain study. |
| Trad_control_12 | a numeric vector, the return value 'perm_control_12' got by meta.TradPerm method for certain study. |
| Trad_control_22 | a numeric vector, the return value 'perm_control_22' got by meta.TradPerm method for certain study. |
| MC_case_11 | a numeric vector, the return value 'perm_case_11' got by meta.MCPerm method for certain study. |
| MC_case_12 | a numeric vector, the return value 'perm_case_12' got by meta.MCPerm method for certain study. |
| MC_case_22 | a numeric vector, the return value 'perm_case_22' got by meta.MCPerm method for certain study. |
| MC_control_11 | a numeric vector, the return value 'perm_control_11' got by meta.MCPerm method for certain study. |
| MC_control_12 | a numeric vector, the return value 'perm_control_12' got by meta.MCPerm method for certain study. |
| MC_control_22 | a numeric vector, the return value 'perm_control_22' got by meta.MCPerm method for certain study. |
| Trad_col | the color of cumulative distribution cure for Trad_case_11/Trad_case_12/Trad_case_22/ Trad_control_11/Trad_control_12/Trad_control_22. Default value is 'grey'. |
| MC_col | the color of cumulative distribution cure for MC_case_11/MC_case_12/MC_case_22/ MC_control_11/MC_control_12/MC_control_22. Default value is 'black'. |
| title | the main title(on top). |
| xlab | X axis label, default value is 'genotype count'. |

## Details

Separately plotting histplot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative genotype count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.KS, VS.Genotype.CDC, VS.Genotype.QQ, VS.Allele.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
```

```
## plot study 12
# VS.Genotype.CDC(result1$perm_case_11[12,],result1$perm_case_12[12,],result1$perm_case_22[12,],
   # result1$perm_control_11[12,],result1$perm_control_12[12,],result1$perm_control_22[12,],
 # result2$perm_case_11[12,],result2$perm_case_12[12,],result2$perm_case_22[12,],
 # result2$perm_control_11[12,],result2$perm_control_12[12,],result2$perm_control_22[12,],
    # Trad_col="grey",MC_col="black", title="hist_plot for six genotype")
```

---

VS.Genotype.QQ *separately plot quantile-quantile plot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study*

---

## Description

Separately plot quantile-quantile plot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study.

## Usage

```
VS.Genotype.QQ(Trad_case_11, Trad_case_12, Trad_case_22,
    Trad_control_11, Trad_control_12, Trad_control_22,
 MC_case_11, MC_case_12, MC_case_22,
 MC_control_11, MC_control_12, MC_control_22,
 scatter_col = "black", line_col = "black",
 title = NULL, xlab = "Quantile of genotype count (TradPerm)",
 ylab = "Quantile of genotype count (MCPerm)")
```

## Arguments

| | |
|---|---|
| Trad_case_11 | a numeric vector, the return value 'perm_case_11' got by meta.TradPerm method for certain study. |
| Trad_case_12 | a numeric vector, the return value 'perm_case_12' got by meta.TradPerm method for certain study. |
| Trad_case_22 | a numeric vector, the return value 'perm_case_22' got by meta.TradPerm method for certain study. |
| Trad_control_11 | a numeric vector, the return value 'perm_control_11' got by meta.TradPerm method for certain study. |
| Trad_control_12 | a numeric vector, the return value 'perm_control_12' got by meta.TradPerm method for certain study. |
| Trad_control_22 | a numeric vector, the return value 'perm_control_22' got by meta.TradPerm method for certain study. |
| MC_case_11 | a numeric vector, the return value 'perm_case_11' got by meta.MCPerm method for certain study. |

| | |
|---|---|
| MC_case_12 | a numeric vector, the return value 'perm_case_12' got by meta.MCPerm method for certain study. |
| MC_case_22 | a numeric vector, the return value 'perm_case_22' got by meta.MCPerm method for certain study. |
| MC_control_11 | a numeric vector, the return value 'perm_control_11' got by meta.MCPerm method for certain study. |
| MC_control_12 | a numeric vector, the return value 'perm_control_12' got by meta.MCPerm method for certain study. |
| MC_control_22 | a numeric vector, the return value 'perm_control_22' got by meta.MCPerm method for certain study. |
| scatter_col | the color for scatter points of quantile-quantile plot, default value is 'black'. |
| line_col | the color of line which passes through the sample distribution probs quantiles, the first and third quartiles. Default value is 'black'. |
| title | the main title(on top). |
| xlab,ylab | X axis label, default value is "Quantile of genotype count (TradPerm)". Y axis label, default value is "Quantile of genotype count (MCPerm)". |

## Details

Separately plotting quantile-quantile plot for the return value(genotype count) of 'meta.TradPerm' and 'meta.MCPerm' for certain study is to compare the simulative allele count distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm..

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.QQ, VS.KS, VS.Genotype.CDC, VS.Genotype.Hist, VS.Allele.QQ, PermMeta.LnOR.qqnorm

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
```

```
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2

## plot study 12
# VS.Genotype.CDC(result1$perm_case_11[12,],result1$perm_case_12[12,],result1$perm_case_22[12,],
   # result1$perm_control_11[12,],result1$perm_control_12[12,],result1$perm_control_22[12,],
 # result2$perm_case_11[12,],result2$perm_case_12[12,],result2$perm_case_22[12,],
 # result2$perm_control_11[12,],result2$perm_control_12[12,],result2$perm_control_22[12,],
    # Trad_col="grey",MC_col="black", title="hist_plot for six genotype")
```

---

`VS.Hist` *plot histplot for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis*

---

## Description

plot histplot for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis.

## Usage

```
VS.Hist(Trad_data, MC_data, Trad_col = "grey", MC_col = "black",
    title = NULL, xlab = NULL)
```

## Arguments

| | |
|---|---|
| `Trad_data` | the return value of function 'meta.TradPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |
| `MC_data` | the return value of function 'meta.MCPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |

| | |
|---|---|
| Trad_col | the color for the histplot body of 'Trad_data', default value is 'grey'. |
| MC_col | the color for the histplot border of 'MC_data', default value is 'black'. |
| title | the main title(on top). |
| xlab | X axis label. |

## Details

Plotting histplot for the return value(e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc) of 'meta.TradPerm' and 'meta.MCPerm' is to compare the simulative data distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.QQ, VS.CDC, VS.KS, VS.Allele.Hist, VS.Genotype.Hist, PermMeta.LnOR.Hist, PermMeta.Hist

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]

## import data
```

```
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]

# VS.Hist(Trad_case_1,MC_case_1,title="Histplot for case_1")
# VS.Hist(result1$perm_Qp,result2$perm_Qp,title="Histplot for Qp")
# VS.Hist(result1$perm_p,result2$perm_p,title="Histplot for p")
```

---

VS.KS *Kolmogorov-Smirnov test for the return value of 'meta.TradPerm' and 'meta.MCPerm'*

---

## Description

Kolmogorov-Smirnov test for the return value of 'meta.TradPerm' and 'meta.MCPerm'.

## Usage

```
VS.KS(Trad_data, MC_data, scatter_alpha = 0.01, line_alpha = 0.001,
    scatter_col = "black", line_col = "red",
 xlab = NULL, ylab = "KS test p_value",
 title = "KS (Kolmogorov-Smirnov) test")
```

## Arguments

| | |
|---|---|
| Trad_data | a matrix with more than one line, each line is the return value of function 'meta.TradPerm'(e.g. 'perm_case_11' of certain study, 'perm_Qp', 'perm_p' etc). |
| MC_data | a matrix with more than one line, each line is the return value of function 'meta.MCPerm'(e.g. 'perm_case_11' of certain study, 'perm_Qp', 'perm_p' etc). |
| scatter_alpha | numeric indicates k-s test p.value less 'scatter_alpha' to be plotted. Default value is 0.01. |
| line_alpha | numeric indicates the threshold value of k-s test significance. Default value is 0.001. |
| scatter_col | the color of sctter points which to be plotted. Default value is 'black'. |
| line_col | the color of the line. Default value is 'red'. |
| xlab,ylab | X axis label. Y axis label, default value is 'KS test p_value'. |
| title | the main title(on top),default value is "KS (Kolmogorov-Smirnov) test". |

## Details

Kolmogorov-Smirnov test for the return value(e.g. 'perm_case_11', 'perm_Qp' etc.) of 'meta.TradPerm' and 'meta.MCPerm' to test the corresponding form the whole. Just plotting p.value less the 'scatter_alpha' will more clear to see the number of no-corresponding data.

## Value

KS_p a numeric vector, p.value of K-S test.

## Note

Parameter 'Trad_data' and 'MC_data' are matrix with more than one line.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

The two-sided one-sample distribution comes via Marsaglia, Tsang and Wang (2003).

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.CDC, VS.QQ VS.Allele.Hist, VS.Genotype.Hist

## Examples

```
## write the return value 'perm_case_11','perm_Qp' or 'perm_p' of function 'meta.TradPerm'
  # to file 'Trad_data.txt'
## write the return value 'perm_case_11','perm_Qp' or 'perm_p' of function 'meta.MCPerm'
  # to file 'MC_data.txt'
## when all meta analysis run over, read the file to K-S test
## if the line of file is separated by '\t'
# Trad=read.table("Trad_data.txt",sep="\t",header=FALSE)
# MC=read.table("MC_data.txt",sep="\t",header=FALSE)

# VS.KS(Trad, MC)
```

---

VS.QQ *plot quantile-quantile plot for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis.*

---

## Description

plot quantile-quantile plot for the return value of 'meta.TradPerm' and 'meta.MCPerm' for certain study or meta analysis.

## Usage

```
VS.QQ(Trad_data, MC_data, scatter_col = "black", line_col = "black", title = "QQ plot",
    xlab = "Quantile for TradPerm data)", ylab = "Quantile for MCPerm data")
```

## Arguments

| | |
|---|---|
| Trad_data | the return value of function 'meta.TradPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |
| MC_data | the return value of function 'meta.MCPerm', e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc. |
| scatter_col | the color for scatter points of quantile-quantile plot, default value is 'black'. |
| line_col | the color of line which passes through the sample distribution probs quantiles, the first and third quartiles. Default value is 'black'. |
| title | the main title(on top), default value is 'QQ plot'. |
| xlab,ylab | X axis label, default is "Quantile for TradPerm data)". Y axis label, default is "Quantile for MCPerm data". |

## Details

Plotting quantile-quantile plot for the return value(e.g. 'perm_case_11' of certain stuy, 'perm_Qp', 'perm_p' etc) of 'meta.TradPerm' and 'meta.MCPerm' is to compare the simulative data distribution got by TradPerm and MCPerm method whether are same.

MCPerm details see chisq.MCPerm. TradPerm details see chisq.TradPerm.

## Author(s)

Lanying Zhang and Yongshuai Jiang <jiangyongshuai@gmail.com>

## References

William S Noble(Nat Biotechnol.2009): How does multiple testing correction work?

Edgington. E.S.(1995): Randomization tests, 3rd ed.

## See Also

meta.MCPerm, meta.TradPerm, chisq.MCPerm, chisq.TradPerm, VS.Hist, VS.CDC, VS.KS, VS.Allele.QQ, VS.Genotype.QQ

## Examples

```
## import data
# data(MetaGenotypeData)
## delete first line which contains the names of each column
# temp=MetaGenotypeData[-1,];
# rowNum=nrow(temp)
# gen=matrix(0,nrow=rowNum,ncol=1);
# aff=matrix(0,nrow=rowNum,ncol=1);
# for(j in 1:rowNum){
```

```
 # gen[j,]=paste(temp[j,14],temp[j,15],sep=" ");
 # case_num=length(unlist(strsplit(temp[j,14],split=" ")));
 # control_num=length(unlist(strsplit(temp[j,15],split=" ")));
 # case_aff=paste(rep(2,case_num),collapse=" ");
 # control_aff=paste(rep(1,control_num),collapse=" ");
 # aff[j,]=paste(case_aff,control_aff,sep=" ");
# }
# result1=meta.TradPerm(gen,aff,split=" ",sep="/",naString="-",
    # model="allele",method="MH",repeatNum=1000)
# result1
## plot study 12
# Trad_case_1=2*result1$perm_case_11[12,]+result1$perm_case_12[12,]

## import data
# data(MetaGenotypeCount)
## delete the first line which is the names for columns.
# temp=MetaGenotypeCount[-1,,drop=FALSE]
# result=meta.MCPerm(case_11=as.numeric(temp[,14]),case_12=as.numeric(temp[,16]),
 # case_22=as.numeric(temp[,18]),control_11=as.numeric(temp[,15]),
 # control_12=as.numeric(temp[,17]),control_22=as.numeric(temp[,19]),
 # model="allele",method="MH",repeatNum=100000)
# result2
## plot study 12
# MC_case_1=2*result2$perm_case_11[12,]+result2$perm_case_12[12,]

# VS.QQ(Trad_case_1,MC_case_1,title="cumulative distribution cure for case_1")
# VS.QQ(result1$perm_Qp,result2$perm_Qp,title="cumulative distribution cure for Qp")
# VS.QQ(result1$perm_p,result2$perm_p,title="cumulative distribution cure for p")
```

# Index