

Package ‘wtest’

September 3, 2019

Type Package

Title The W-Test for Genetic Interactions Testing

Version 3.2

Author Rui Sun, Maggie Haitian Wang

Maintainer Rui Sun <rsunzju@gmail.com>

Description Perform the calculation of W-test, diagnostic checking, calculate minor allele frequency (MAF) and odds ratio.

License GPL-2

LazyData TRUE

RoxygenNote 6.1.1

NeedsCompilation yes

Encoding UTF-8

Repository CRAN

Date/Publication 2019-09-03 07:10:02 UTC

R topics documented:

CpG.pos	2
diabetes.geno	2
genotype	3
hf	3
hf.snps.meth	4
maf	5
methylation	6
methylation.recode	6
odds.ratio	7
phenotype1	7
phenotype2	8
SNP.pos	8
w.diagnosis	9
w.qqplot	10
wtest	11

wtest.high	14
wtest.snps.meth	16

Index	19
--------------	-----------

CpG.pos	<i>Position of CpG Sites for the methylation Data</i>
---------	---

Description

This dataset contains pseudo positions of 200 CpG sites.

Usage

```
data(CpG.pos)
```

Format

The format is: CpG sites by rows; column 1: names of CpG sites; column 2: positions of CpG sites.

diabetes.geno	<i>Genotype Data of Candidate Diabetes Genes</i>
---------------	--

Description

A data frame contains 23 SNPs for 115 individuals.

Usage

```
data(diabetes.geno)
```

Format

The format is: subjects by rows and genotypes by columns.

References

Wang, M. H., Li, J., Yeung, V. S. Y., Zee, B. C. Y., Yu, R. H. Y., Ho, S., & Waye, M. M. Y. (2014). Four pairs of gene-gene interactions associated with increased risk for type 2 diabetes (CDKN2BAS-KCNJ11), obesity (SLC2A9-IGF2BP2, FTO-APOA5), and hypertension (MC4R-IGF2BP2) in Chinese women. *Meta Gene*, 2, 384-391. <http://doi.org/10.1016/j.mgene.2014.04.010>

genotype	<i>Example Genotype Data</i>
----------	------------------------------

Description

This simulated data frame contains 300 individuals and 200 SNPs.

Usage

```
data(genotype)
```

Format

The format is: subjects by rows, and genotype by columns.

hf	<i>Parameter Estimation for W-test Probability Distribution</i>
----	---

Description

Estimate parameters (h and f) for W-test.

Usage

```
hf(data, w.order, B = 400, n.sample = nrow(data),
    n.marker = "default.nmarker")
```

Arguments

data	a data frame or matrix containing genotypes in the columns and subjects in the rows. Genotypes should be coded as (0, 1, 2) or (0, 1).
w.order	a numeric number. $w.order = 1$ gives main effect calculation. $w.order = 2$ gives pairwise interaction calculation. $w.order > 2$ gives high order interaction calculation.
B	a numeric number specifying the number of replicates. Default is 400.
n.sample	a numeric number specifying the number of samples to be used for estimating parameters. Default is the total number of samples in the data.
n.marker	a numeric value, the number of biomarkers to include in bootstrapping. For $order = 1$, the default = $\min(P, 1000)$, and for $order = 2$, default = $\min(P, 50)$. P is the total number of markers in the data.

Value

a set of h and f values indexed by k , estimated automatically. For main effect, k is the number of levels of a predictor variable. For interactions, k is the number of categorical combinations of a variable pair.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research*. doi:10.1093/nar/gkw347.

See Also

[wtest](#), [w.diagnosis](#), [w.qqplot](#)

Examples

```
data(diabetes.geno)

# Please note that parameter B is recommended to be greater than 400.
# For high order interaction analysis (w.order > 2), it is recommended to use default n.sample.
hf1 <- hf(data = diabetes.geno, w.order = 1, B = 100)
hf2 <- hf(data = diabetes.geno, w.order = 2, B = 80)
```

hf.snps.meth

Parameter Estimation for W-test Probability Distribution in Gene-methylation Data

Description

Estimate parameters (h and f) for W-test.

Usage

```
hf.snps.meth(B = 400, geno, meth, y, geno.pos, meth.pos, window.size,
             n.sample = nrow(geno), n.pair = 1000)
```

Arguments

B	a numeric number specifying the number of bootstrapping times. Default is 400.
geno	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1). SNP names should be stored as column names.
meth	a data frame or matrix containing methylation data in the columns. Methylation data should be recoded as (0, 1, 2) or (0, 1). Names of CpG sites should be stored as column names.
y	a numeric vector of 0 or 1.
geno.pos	a data frame containing SNP names and positions in two columns.
meth.pos	a data frame containing CpG names and positions in two columns.

window.size	a numeric number specifying the size of genome distance. Interaction of the SNPs and CpG sites located within the size of genome distance will be evaluated exhaustively.
n.sample	a numeric number specifying the number of samples to be included for estimating parameters. Default is the total number of samples.
n.pair	a numeric value, the number of SNP-CpG pairs to use in bootstrapping. Default = $\min(P, 1000)$. P is the total number of pairs within the window.size.

Value

a set of h and f values indexed by k , estimated automatically. Variable k is the number of categorical combinations of a variable pair.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research*. doi:10.1093/nar/gkw347.

Examples

```
data(SNP.pos)
data(CpG.pos)
data(genotype)
data(methylation)
data(phenotype2)

# Please note that parameter B is recommended to be greater than 400.
hf.pair <- hf.snps.meth(B = 80, geno = genotype, meth = methylation, y = phenotype2,
                      geno.pos = SNP.pos, meth.pos = CpG.pos, window.size = 1000)
```

maf

Minor Allele Frequency

Description

Calculate minor allele frequency.

Usage

```
maf(data, which.snp = NULL)
```

Arguments

`data` a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1).

`which.snp` a numeric value, indicating which SNP to calculate. When `which.snp = NULL`, MAF of all the markers is calculated. Default is `NULL`.

Value

The MAF of one marker.

Examples

```
data(diabetes.geno)
result <- maf(diabetes.geno, which.snp=10)
```

methylation	<i>Example Methylation Data</i>
-------------	---------------------------------

Description

This data frame contains 300 samples and 200 CpG sites.

Usage

```
data(methylation)
```

Format

The format is: subjects by rows and methylation by columns.

methylation.recode	<i>Recode Methylation Data</i>
--------------------	--------------------------------

Description

Code a CpG variable into two levels (high and low) by the two-mean clustering method.

Usage

```
methylation.recode(data)
```

Arguments

`data` a data frame or matrix contains methylation data in the columns.

Examples

```
data(methylation)
data.recoded <- methylation.recode(methylation)
```

odds.ratio	<i>Odds Ratio</i>
------------	-------------------

Description

Calculate odds ratio for a single SNP or a pair of SNPs. Single marker odds ratio is computed by contingency table as the odds of disease at minor allele vs the odds of diseases at major allele. Odds ratio of a pair of SNPs is calculated by the Logistic Regression.

Usage

```
odds.ratio(data, y, w.order, which.marker)
```

Arguments

data	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1), according to minor allele count.
y	binary values.
w.order	a numeric number taking values 1 or 2. If w.order = 1, odds ratio of main effect is calculated. If w.order = 2, odds ratio of pairwise interaction is calculated.
which.marker	a numeric vector, when w.order = 1, a single value indicating the column index of the variable to calculate; when w.order = 2, a vector indicating the column index of a SNP-pair to calculate.

Value

The odds ratio of a SNP or a SNP-pair.

Examples

```
data(diabetes.geno)
data(phenotype1)
y <- as.numeric(phenotype1)
OR.sn4.sn8 <- odds.ratio(diabetes.geno, y, w.order=2, which.marker = c(4,8))
OR.sn4 <- odds.ratio(diabetes.geno, y, w.order = 1, which.marker = 4)
```

phenotype1	<i>Phenotype of the diabetes.geno Data</i>
------------	--

Description

A binary variable indicate the status of 115 individuals (1 = affected, 0 = unaffected).

Usage

```
data(phenotype1)
```

Format

The format is: 115 rows and 1 column.

References

Wang, M. H., Li, J., Yeung, V. S. Y., Zee, B. C. Y., Yu, R. H. Y., Ho, S., & Waye, M. M. Y. (2014). Four pairs of gene-gene interactions associated with increased risk for type 2 diabetes (CDKN2BAS-KCNJ11), obesity (SLC2A9-IGF2BP2, FTO-APOA5), and hypertension (MC4R-IGF2BP2) in Chinese women. *Meta Gene*, 2, 384-391. <http://doi.org/10.1016/j.mgene.2014.04.010>

phenotype2

Simulated Phenotype of the genotype-methylation Data

Description

The phenotype of the 300 individuals (1 = affected, 0 = unaffected).

Usage

```
data(phenotype2)
```

Format

The format is: 300 rows and 1 column.

SNP.pos

Pseudo Position of SNPs for the genotype Data

Description

Positions of 200 SNPs.

Usage

```
data(SNP.pos)
```

Format

The format is: SNPs by rows; column 1: names of SNPs, column 2: positions of SNPs.

w.diagnosis

*W-test Probability Distribution Diagnostic Plot***Description**

Diagnostic checking of W-test probability distribution estimation.

Usage

```
w.diagnosis(data, w.order = c(1, 2), n.rep = 10,
  n.sample = nrow(data), n.marker = ncol(data), hf1 = "default.hf1",
  hf2 = "default.hf2", ...)
```

Arguments

data	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1).
w.order	an integer value of 0 or 1. w.order = 1 gives main effect calculation; w.order = 2 gives pairwise calculation.
n.rep	a numeric value, the number of bootstrapping times.
n.sample	a numeric value, the number of samples to use in bootstrapping. Default is the total number of samples in the data.
n.marker	a numeric value, the number of markers to use in bootstrapping. Default is the total number of markers.
hf1	<i>h</i> and <i>f</i> values to calculate main effect, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 3. Needed when w.order = 1.
hf2	<i>h</i> and <i>f</i> values to calculate interaction associations, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 9. Needed when w.order = 2.
...	graphical parameters.

Details

This function evaluates the input W values of main or interaction effects using a set of null Y by the W-test, and the evaluation is performed in several bootstrap samples to achieve fast and stable output. The W histogram and its theoretical Chi-squared distribution density with *f* degrees of freedom are plotted indexed by *k*. Close overlaying of the histogram and the probability density curve indicates that the estimated *h* and *f* give a good test statistic probability distribution.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research*. doi:10.1093/nar/gkw347.

See Also

[wtest](#), [hf](#), [w.qqplot](#)

Examples

```
data(diabetes.geno)
# Please note that parameter B is recommended to be greater than 400.
hf1 <- hf(data = diabetes.geno, w.order = 1, B = 100)
hf2 <- hf(data = diabetes.geno, w.order = 2, B = 50)
w.diagnosis(diabetes.geno, w.order = 1, n.rep = 100, hf1 = hf1, main=NULL, xlab=NULL, ylab=NULL)
w.diagnosis(diabetes.geno, w.order = 2, n.rep = 100, hf2 = hf2, main=NULL, xlab=NULL, ylab=NULL)
```

w.qqplot

W P-values Diagnosis by Q-Q Plot

Description

Draw a Q-Q plot for W-test

Usage

```
w.qqplot(data, y, w.order = c(1, 2), input.poolsize = 200,
         hf1 = "default.hf1", hf2 = "default.hf2", ...)
```

Arguments

data	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1).
y	a numeric vector of 0 or 1.
w.order	a numeric number taking values 1 or 2. w.order = 1 gives main effect Q-Q plot. w.order = 2 gives interaction Q-Q plot.
input.poolsize	a numeric number; The maximum number of SNPs to calculate the Q-Q plot. Default is 200. The input.poolsize is suggested to set as 1000 for w.order = 1, and 200 for w.order = 2.
hf1	<i>h</i> and <i>f</i> values to calculate main effect, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 3. Needed when w.order = 1.
hf2	<i>h</i> and <i>f</i> values to calculate interaction associations, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 9. Needed when w.order = 2.
...	graphical parameters.

Details

With a given data and y, the p-value of W-test is calculated at given *h* and *f* values, which are plotted against the theoretical distribution.

Value

Q-Q plot

Examples

```

data(diabetes.geno)
data(phenotype1)
## Step 1. HF Calculation
# Please note that parameter B is recommended to be greater than 400.
hf1<-hf(data = diabetes.geno, w.order = 1, B = 200)

## Step 2. Q-Q Plot
w.qqplot(data = diabetes.geno, y = phenotype1, w.order = 1, hf1 = hf1, cex =.5)
abline(0,1)

```

wtest

*W-test***Description**

This function performs the *W*-test to calculate main effect or pairwise interactions in case-control studies for categorical data sets. The test measures target variables' distributional difference between cases and controls via a combined log of odds ratio. It follows a Chi-squared probability distribution with data-adaptive degrees of freedom. For pairwise interaction calculation, the user has 3 options: (1) calculate a single pair's *W*-value, (2) calculate pairwise interaction for a list of variables, which p-values are smaller than a threshold (input.pval); (3) calculate the pairwise interaction exhaustively for all variables. For both main and interaction calculation, the output can be filtered by p-values, such that only sets with smaller p-value than a threshold (output.pval) will be returned. An extension of the *W*-test for rare variant analysis is available in *zfa* package.

Usage

```

wtest(data, y, w.order = c(1, 2), hf1 = "default.hf1",
      hf2 = "default.hf2", which.marker = NULL, output.pval = NULL,
      sort = TRUE, input.pval = 0.1, input.poolsize = 150)

```

Arguments

data	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1).
y	a numeric vector of 0 or 1.
w.order	an integer value of 0 or 1. w.order = 1 for main effect calculation; w.order = 2 for pairwise calculation.
hf1	<i>h</i> and <i>f</i> values to calculate main effect, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 3. Needed when w.order = 1.
hf2	<i>h</i> and <i>f</i> values to calculate interaction associations, organized as a matrix, with columns (<i>k</i> , <i>h</i> , <i>f</i>), <i>k</i> = 2 to 9. Needed when w.order = 2.

<code>which.marker</code>	a numeric vector, when <code>w.order = 1</code> , a single value indicating the column index of a SNP to calculate, when <code>w.order = 2</code> , a vector indicating the column index of a SNP-pair to calculate. Default <code>which.marker = NULL</code> means main or interaction effect will be calculated exhaustively.
<code>output.pval</code>	a p-value threshold for filtering the output. If <code>NULL</code> , all the results will be listed; otherwise, the function will only output the results with p-values smaller than the <code>output.pval</code> .
<code>sort</code>	a logical value indicating whether or not to sort the output by p-values in ascending order. Default = <code>TRUE</code> .
<code>input.pval</code>	a p-value threshold to select markers for pairwise calculation, used only when <code>w.order = 2</code> . When specified, only markers with main effect p-value smaller than <code>input.pval</code> will be passed to interaction effect calculation. Default = <code>0.10</code> . Set <code>input.pval = NULL</code> or <code>1</code> for exhaustive pairwise calculation.
<code>input.poolsize</code>	an integer, with value less than the number of input variables. It is an optional filter to control the maximum number of variables to include in pairwise calculation, used only when <code>w.order = 2</code> . When specified, the function selects top <code>input.poolsize</code> number of variables to calculate pairwise interactions. It can be used separately or jointly with <code>input.pval</code> , whichever gives smaller input variable pool size. Default = <code>50</code> . Set <code>input.poolsize = NULL</code> for exhaustive pairwise calculation. It can be useful for data exploration, when there are a large number of variables with extremely small main effect p-values.

Details

W-test is a model-free statistical test to measure main effect or pairwise interactions in case-control studies with categorical variables. Theoretically, the test statistic follows a Chi-squared distribution with f degrees of freedom. The data-adaptive degree of freedom f , and a scalar h in the test statistics allow the W-test to correct for distributional bias due to sparse data and small sample size. Let k be the number of columns of the 2 by k contingency table formed by a single variable or a variable pair. When the sample size is large and there is no population stratification, the h and f will approximate well to the theoretical value $h = (k-1)/k$, and $f = k-1$. When sample size is small and there is population stratification, the h and f will vary to correct for distributional bias caused by the data structure.

When `w.order = 2`, the `wtest()` will automatically calculate the main effect first and then do a pre-filter before calculating interactions. This filtering is to avoid overloading the memory before having a better understanding of the data. User can specify a smaller `input.pval` such as `0.05` or `0.001` for less output, or `input.pval=1` or `NULL` for exhaustive pairwise calculation. Another optional filter is `input.poolsize`. It will take the top `input.poolsize` number of variables to calculate pairwise effect exhaustively, selected by smallest p-value; when used together with `input.pval`, the smaller set will be passed to pairwise calculation.

Value

An object "wtest" containing:

`order` the "w.order" specified.

results	When <code>w.order = 1</code> , the test results include: the ID of SNP, the W value, k , and p-value. When <code>w.order = 2</code> and <code>which.marker = NULL</code> , the test results include: (information of the pair, column 1-5) [SNP1 name, SNP2, name, W-value, k, p-value]; (Information of the first variable in the pair, column 6-8) [W-value, k, p-value]; (Information of the second variable in the pair, column 9-11) [W-value, k, p-value].
hf1	The h and f values used in main effect calculation.
hf2	The h and f values used in pairwise interaction calculation.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research*. doi:10.1093/nar/gkw347.

Maggie Haitian Wang, Haoyi Weng, Rui Sun, Jack Lee, William K.K. Wu, Ka Chun Chong, Benny C.Y. Zee. (2017). A Zoom-Focus algorithm (ZFA) to locate the optimal testing region for rare variant association tests. *Bioinformatics*, 33(15), 2330-2336.

See Also

[hf](#), [w.diagnosis](#), [w.qqplot](#)

Examples

```
data(diabetes.geno)
data(phenotype1)

## Step 1. HF Calculation
# Please note that parameter B is recommended to be greater than 400.
hf1 <- hf(data = diabetes.geno, w.order = 1, B = 100)
hf2 <- hf(data = diabetes.geno, w.order = 2, B = 50)

## Step 2. W-test Calculation
w1 <- wtest(diabetes.geno, phenotype1, w.order = 1, hf1 = hf1)
w2 <- wtest(diabetes.geno, phenotype1, w.order = 2, input.pval = 0.3,
            input.poolsize = 50, output.pval = 0.01, hf1 = hf1, hf2 = hf2)
w.pair <- wtest(diabetes.geno, phenotype1, w.order = 2, which.marker = c(10,13), hf2 = hf2)
```

wtest.high

*W-test for High Order Interaction Analysis***Description**

This function performs the W-test to calculate high-order interactions in case-control studies for categorical data sets. The test measures target variables' distributional difference between cases and controls via a combined log of odds ratio. It follows a Chi-squared probability distribution with data-adaptive degrees of freedom. For high-order interaction calculation, the user has 3 options: (1) calculate W-test of a set of SNPs, (2) calculate high-order interaction for a list of variables, which p-values are smaller than a threshold (`input.pval`); (3) calculate high-order interaction exhaustively for all variables. Output can be filtered by p-values, such that only sets with smaller p-value than a threshold (`output.pval`) will be returned.

Usage

```
wtest.high(data, y, w.order = 3, hf1 = "default.hf1",
           hf.high.order = "default.high", which.marker = NULL,
           output.pval = NULL, sort = TRUE, input.pval = 0.1,
           input.poolsize = 10)
```

Arguments

<code>data</code>	a data frame or matrix containing genotypes in the columns. Genotypes should be coded as (0, 1, 2) or (0, 1).
<code>y</code>	a numeric vector of 0 or 1.
<code>w.order</code>	an integer value, indicating the order of high-way interactions. For example, <code>w.order = 3</code> for three-way interaction analysis.
<code>hf1</code>	<i>h</i> and <i>f</i> values to calculate main effect, organized as a matrix, with columns (<i>k, h, f</i>), <i>k</i> = 2 to 3.
<code>hf.high.order</code>	<i>h</i> and <i>f</i> values to calculate high-order interactions, organized as a matrix, with columns (<i>k, h, f</i>), where <i>k</i> is the number of genotype combinations of a set of SNPs.
<code>which.marker</code>	a numeric vector indicating the column index of a set of SNPs to calculate. Default <code>which.marker = NULL</code> gives an exhaustively high-order interaction calculation.
<code>output.pval</code>	a p-value threshold for filtering the output. If <code>NULL</code> , all the results will be listed; otherwise, the function will only output the results with p-values smaller than the <code>output.pval</code> .
<code>sort</code>	a logical value indicating whether or not to sort the output by p-values in ascending order. Default = <code>TRUE</code> .
<code>input.pval</code>	a p-value threshold to select markers for high-order interaction calculation, used only when <code>w.order > 2</code> . When specified, only markers with main effect p-value smaller than <code>input.pval</code> will be passed to interaction effect calculation. Default = 0.10. Set <code>input.pval = NULL</code> or 1 for exhaustive calculation.

`input.poolsize` an integer, with value less than the number of input variables. It is an optional filter to control the maximum number of variables to include in high-order interaction calculation, used only when `w.order > 2`. When specified, the function selects top `input.poolsize` number of variables to calculate interactions. It can be used separately or jointly with `input.pval`, whichever gives smaller input pool size. Default = 10. Set `input.poolsize = NULL` for exhaustive calculation. It can be useful for data exploration, when there are a large number of variables with extremely small main effect p-values.

Details

W-test is a model-free statistical test originally proposed to measure main effect or pairwise interactions in case-control studies with categorical variables. It can be extended to high-order interaction detection by the `wtest.high()` function. Theoretically, the test statistic follows a Chi-squared distribution with f degrees of freedom. The data-adaptive degree of freedom f , and a scalar h in the test statistics allow the W-test to correct for distributional bias due to sparse data and small sample size. Let k be the number of columns of the 2 by k contingency table formed by a single variable or a variable pair. When the sample size is large and there is no population stratification, the h and f will approximate well to the theoretical value $h = (k-1)/k$, and $f = k-1$. When sample size is small and there is population stratification, the h and f will vary to correct for distributional bias caused by the data structure.

When `w.order > 2`, the `wtest()` will automatically calculate the main effect first and then do a pre-filter before calculating interactions. This filtering is to avoid overloading the memory before having a better understanding of the data. User can specify a smaller `input.pval` such as 0.05 or 0.001 for less output, or `input.pval=1` or `NULL` for exhaustive high-order interaction calculation. Another optional filter is `input.poolsize`. It will select the top `input.poolsize` number of variables, ranked by p-values, to calculate high-order interactions. When used together with `input.pval`, the algorithm selects the smaller set in the high-order calculation.

Value

An object "wtest" containing:

<code>order</code>	the "w.order" specified.
<code>results</code>	When <code>order > 2</code> and <code>which.marker = NULL</code> , the test results include: (information of a set) [SNPs name, W-value, k, p-value]; (Information of the first variable in the set) [W-value, k, p-value]; (Information of the second variable in the set) [W-value, k, p-value] ...
<code>hf1</code>	The h and f values used in main effect calculation.
<code>hf2</code>	The h and f values used in high-order interaction calculation.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. *Nucleic Acids Research*. doi:10.1093/nar/gkw347.

See Also

[hf](#), [w.diagnosis](#), [w.qqplot](#)

Examples

```
data(diabetes.geno)
data(phenotype1)

## Step 1. HF Calculation
# Please note that parameter B is recommended to be greater than 400 for w.order = 1 or 2.
# For high order interaction analysis (w.order > 2), it is recommended to use default n.sample.
hf1 <- hf(data = diabetes.geno, w.order = 1, B = 100)
hf.high <- hf(data = diabetes.geno, w.order = 3, B = 30, n.marker = 10)

## Step 2. W-test Calculation
w1 <- wtest.high(diabetes.geno, phenotype1, w.order = 1, hf1 = hf1)
w3 <- wtest.high(diabetes.geno, phenotype1, w.order = 3, input.pval = 0.3,
                 input.poolsize = 50, output.pval = 0.5, hf1 = hf1, hf.high.order = hf.high)
w.set <- wtest.high(diabetes.geno, phenotype1, w.order = 3, which.marker = c(10,13,20),
                   hf.high.order = hf.high)
```

wtest.snps.meth

W-test for Gene-methylation Interaction Analysis

Description

Calculate cis-gene-methylation interaction of a (SNP, CpG) pair in user-defined window, and can run in a genome-wide manner. The output can be filtered by p-values, such that only sets with smaller p-value than the threshold (`output.pval`) will be returned.

Usage

```
wtest.snps.meth(geno, meth, y, geno.pos, meth.pos, window.size = 10000,
               hf = "default.hf", output.pval = NULL, sort = TRUE,
               which.marker = NULL)
```

Arguments

`geno` a data frame or matrix containing genotypes in the columns and subjects in the rows. Genotypes should be coded as (0, 1, 2) or (0, 1). SNP names should be stored as column names of the data.

meth	a data frame or matrix containing methylation data in the columns. Methylation data should be recoded as (0, 1, 2) or (0, 1). Names of CpG sites should be stored as column names of the data.
y	a numeric vector of 0 or 1.
geno.pos	a data frame containing SNP names and positions in two columns.
meth.pos	a data frame containing CpG names and positions in two columns.
window.size	a numeric number specifying the size of genome distance. Interaction effects of the SNPs and CpG sites located within the size of genome distance will be evaluated exhaustively.
hf	h and f values to calculate gene-methylation interaction associations, organized as a matrix, with columns (k, h, f), $k = 2$ to 6.
output.pval	a p-value threshold for filtering the output. If NULL, all results will be listed; otherwise, the function will only output the results with p-values smaller than output.pval.
sort	a logical value indicating whether or not to sort the output by p-values in ascending order. Default = TRUE.
which.marker	a vector indicating the column index of a SNP-CpG pair to calculate. Default which.marker = NULL means interaction pairs located within window.size will be calculated exhaustively.

Details

Calculate cis-gene-methylation interaction of a (SNP, CpG) pair in user-defined window, and can run in a genome-wide manner. The output can be filtered by p-values, such that only sets with smaller p-value than the threshold (output.pval) will be returned.

Value

An object "wtest.snps.meth" containing:

results	The test results include: SNP name, CpG name, SNP position, CpG position, W value, k , and p-value.
hf	The h and f values used for each k in pairwise calculation, where $k = 2$ to 6.

Author(s)

Rui Sun, Maggie Haitian Wang

References

Maggie Haitian Wang, Rui Sun, Junfeng Guo, Haoyi Weng, Jack Lee, Inchi Hu, Pak Sham and Benny C.Y. Zee (2016). A fast and powerful W-test for pairwise epistasis testing. Nucleic Acids Research. doi:10.1093/nar/gkw347.

See Also

[wtest](#), [hf.snps.meth](#)

Examples

```
data(SNP.pos)
data(CpG.pos)
data(genotype)
data(methylation)
data(phenotype2)

w <- 13000

# Recode methylation data
methylation <- methylation.recode(methylation)

## Step 1. HF Calculation.
# Please note that parameter B is recommended to be greater than 400.
hf.pair <- hf.snps.meth(B = 80, geno = genotype, meth = methylation, y = phenotype2,
                       geno.pos = SNP.pos, meth.pos = CpG.pos, window.size = w)

## Step 2. Application
result <- wtest.snps.meth(geno = genotype, meth = methylation, y = phenotype2, geno.pos = SNP.pos,
                        meth.pos = CpG.pos, window.size = w, hf = hf.pair, output.pval = 0.1)
```

Index

*Topic **datasets**

CpG.pos, [2](#)
diabetes.geno, [2](#)
genotype, [3](#)
methylation, [6](#)
phenotype1, [7](#)
phenotype2, [8](#)
SNP.pos, [8](#)

CpG.pos, [2](#)

diabetes.geno, [2](#)

genotype, [3](#)

hf, [3](#), [10](#), [13](#), [16](#)
hf.snps.meth, [4](#), [17](#)

maf, [5](#)
methylation, [6](#)
methylation.recode, [6](#)

odds.ratio, [7](#)

phenotype1, [7](#)
phenotype2, [8](#)

SNP.pos, [8](#)

w.diagnosis, [4](#), [9](#), [13](#), [16](#)
w.qqplot, [4](#), [10](#), [10](#), [13](#), [16](#)
wtest, [4](#), [10](#), [11](#), [17](#)
wtest.high, [14](#)
wtest.snps.meth, [16](#)