Package ‘diffMeanVar’

October 5, 2017

Type Package

Title Detecting Gene Probes with Different Means or Variances Between Two Groups

Version 0.0.6

Date 2017-10-05

Author Xuan Li [aut, ctb], Weiliang Qiu [aut, cre], Yuejiao Fu [aut, ctb], Xiaogang Wang [aut, ctb]

Maintainer Weiliang Qiu <stwxq@channing.harvard.edu>

Depends R (>= 3.4.0), Biobase, missMethyl, stats

Imports MASS, graphics, lawstat, methylumi, grDevices

biocViews Bioinformatics, DifferentialExpression


License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-05 14:32:49 UTC

R topics documented:

AWvarTest ................................................................. 2
BFtest ................................................................. 3
**AWvarTest**

Test for equality of variance based on Ahn and Wang’s (2013) score test.

**Description**

Test for equality of variance based on Ahn and Wang’s (2013) score test.

**Usage**

`AWvarTest(value, group)`

**Arguments**

- **value**: numeric. Measurements to be compared between two groups.
- **group**: numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).

**Value**

A list with 6 elements:

- **U2**: score statistic
- **varU2**: estimated variance of the score statistic
- **T2**: score statistic $U_2 / \text{var}(U_2)$
- **pval**: p-value of the score test
- **z**: squared deviation of value from mean value
- **zbar**: sample average of z
Tests for equality of variance based on Brown and Forsythe’s test

Description

Test for equality of variance based on Brown and Forsythe’s test.

Usage

BFtest(value, group)

Arguments

value numeric. Measurements to be compared between two groups.

group numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).
equalMeanTestWrapper

**Value**

A list with 2 elements:

- **stat**: test statistic value
- **pval**: pvalue of the score test

**Author(s)**

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

**References**


**Examples**

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpGs = 100, nCases = 20, nControls = 20,
                          df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
                          eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = BFTest(value = dat[1,], group = pDat$memsubj)
print(names(res))
print(res)
```

**Description**

Wrapper function to test for equality of mean by using the score test of logistic regression for whole genome.
Usage

equalMeanTestWrapper(
  es,
  grpVar = "group",
  meanTestFunc = scoreTestMean,
  esFlag = "es",
  pvalAdjMethod = "fdr",
  alpha = 0.05,
  nTop = 20,
  probeID.var = "ProbeID",
  gene.var = "Symbol",
  chr.var = "Chromosome",
  applier = lapply,
  verbose = FALSE)

Arguments

es An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.

grpVar character string. The name of the phenotype variable indicating arrays' group membership. 0 means control and 1 means case.

meanTestFunc A user-defined function to test for equality of mean. Available functions provided by diffMeanVar package include: scoreTestMean. The function should contain 2 and only 2 inputs: value and group, where group is a binary vector indicating if a subject is a case (group=1) or control (group=0) and value is a continuous-type variable. The goal of the function is to test if the mean of value for cases is the same as that for controls. The function should return a list object containing at least 2 elements: stat and pval. For each probe stored in es, the function will be applied to test if the probe is different in means between cases and controls.

esFlag character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.

pvalAdjMethod character string. Indicating which p-value adjustment will be used to control for multiple testing.

alpha numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially mean.

nTop integer. Specifying the number of top probes to be displayed if verbose=TRUE

probeID.var character string. Feature annotation variable indicating probe ID.

gene.var character string. Feature annotation variable indicating gene symbol.

chr.var character string. Feature annotation variable indicating chromosome number.

applier function name to do apply operation.

verbose logical. Indicating if intermediate results should be output to screen.
Value

A list of 2 elements. The first element frame is unsorted data frame; the second element frame.s is a sorted data frame object storing the analysis results and containing the following columns: probe (probe id), stat (test statistic), pval (raw p-value), p.adj (adjusted p-value), gene (gene symbol), chr (chromosome number), and pos (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "mean",
eps = 1.0e-3, applicer = lapply)
print(es.sim)

res.equalMeanTestWrapper = equalMeanTestWrapper(
es = es.sim,
grpVar = "memSubj",
meanTest Func = scoreTestMean,
esFlag = "es",
pvalAdjMethod = "fdr",
alpha = 0.05,
nTop = 20,
probeID.var = "probe",
gene.var = "gene",
chr.var = "chr",
applicer=lapply,
verbose=TRUE)
```
equalVarTestWrapper

Wrapper function to test for equality of variance by using the score test of logistic regression for whole genome

Description

Wrapper function to test for equality of variance by using the score test of logistic regression for whole genome.

Usage

equalVarTestWrapper(
es,.grpVar = "group", 
varTestFunc = AWvarTest, 
esFlag = "es", 
pvalAdjMethod = "fdr", 
alpha = 0.05, 
nTop = 20, 
probeID.var = "ProbeID", 
gene.var = "Symbol", 
chr.var = "Chromosome", 
applier = lapply, 
verbose = FALSE)

Arguments

es An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.

grpVar character string. The name of the phenotype variable indicating arrays’ group membership. 0 means control and 1 means case.

varTestFunc A user-defined function to test for equality of variance. Available functions provided by diffMeanVar package include: iAWvar.BF, iAWvar.Levene, iAWvar.TrimMean, AWvarTest,BFtest,FTtest,LeveneTest, and TrimMeanLeveneTest. The function should contain 2 and only 2 inputs: value and group, where group is a binary vector indicating if a subject is a case (group=1) or control (group=0) and value is a continuous-type variable. The goal of the function is to test if the variance of value for cases is the same as that for controls. The function should return a list object containing at least 2 elements: stat and pval. For each probe stored in es, the function will be applied to test if the probe is differentially variable between cases and controls.

esFlag character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.

pvalAdjMethod character string. Indicating which p-value adjustment will be used to control for multiple testing.
alpha numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially variable.

nTop integer. Specifying the number of top probes to be displayed if verbose=TRUE

probeID.var character string. Feature annotation variable indicating probe ID.

gene.var character string. Feature annotation variable indicating gene symbol.

chr.var character string. Feature annotation variable indicating chromosome number.

applier function name to do apply operation.

verbose logical. Indicating if intermediate results should be output to screen.

Value

A list of 2 elements. The first element frame is unsorted data frame; the second element frame.s is a sorted data frame object storing the analysis results and containing the following columns: probe (probe id), stat (test statistic), pval (raw p-value), p.adj (adjusted p-value), gene (gene symbol), chr (chromosome number), and pos (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.TDist(nCpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)

res.equalVarTestWrapper = equalVarTestWrapper(
es = es.sim,
grpVar = "memSubj",
varTestFunc = AWvarTest,
esFlag = "es",
)
```
pvalAdjustMethod = "fdr",
alpha = 0.05,
nTop = 20,
probeID.var = "probe",
gene.var = "gene",
chr.var = "chr",
applier=lapply,
verbose=TRUE)

---

**FTest**

*Test for equality of variance based on F test*

---

**Description**

Test for equality of variance based on F test.

**Usage**

FTest(value, group)

**Arguments**

- **value**: numeric. Measurements to be compared between two groups.
- **group**: numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).

**Value**

A list with 2 elements:

- **stat**: test statistic value
- **pval**: pvalue of the score test

**Author(s)**

Xuan Li <rixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

**References**


Examples

```r
# generate simulated data set from t distribution
define.seed(1234567)
es.sim = genSimData.tDistr(ncpG = 100, nCases = 20, nControls = 20,
                          df = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
                          eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = FTest(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
```

---

**genSimData.tDistr**  
Generating simulated data set from t distributions

**Description**

Generating simulated data set from t distributions.

**Usage**

```r
genSimData.tDistr(
  nCpgs,
  nCases,
  nControls,
  df0 = 10,
  ncp0 = 0,
  df1 = 6,
  ncp1 = 2.393,
  testPara = "var",
  outlierFlag = FALSE,
  eps = 0.001,
  applier = lapply)
```

**Arguments**

- `nCpgs`  
  integer. Number of genes.
- `nCases`  
  integer. Number of cases.
- `nControls`  
  integer. Number of controls.
- `df0`  
  integer. Degree of freedom for group 0 (controls).
- `ncp0`  
  numeric. Non-centrality parameter for group 0 (controls).
df1  integer. Degree of freedom for group 1 (cases).
ncp1  numeric. Non-centrality parameter for group 1 (cases).
testPara  character string. Indicating if the test is for testing equal mean (testPara="mean"), for testing equal variance (testPara="var"), or for testing both equal mean and equal variance (testPara="both").
outlierFlag  logical. Indicating if outliers would be generated. If outlierFlag=TRUE, then we followed Phipson and Oshlack’s (2014) simulation studies to generate one outlier for each CpG site by replacing the DNA methylation level of one diseased subject by the maximum of the DNA methylation levels of all CpG sites.
eps  numeric. If \(|mean_0 - mean_1| < eps\) then we regard \(mean_0 = mean_1\). Similarly, if \(|var_0 - var_1| < eps\) then we regard \(var_0 = var_1\). \(mean_0\) and \(var_0\) are the mean and variance of the t distribution for controls. \(mean_1\) and \(var_1\) are the mean and variance of the t distribution for cases.
apliier  function name to do apply operation.

Value
An ExpressionSet object. The phenotype data of the ExpressionSet object contains 2 columns: arrayID (array id) and memSubj (subject membership, i.e., case (memSubj=1) or control (memSubj=0)). The feature data of the ExpressionSet object contains 4 elements: probe (probe id), gene (psuedo gene symbol), chr (psuedo chromosome number), and memGenes (indicating if a gene is differentially expressed (when testPara="mean") or indicating if a gene is differentially variable (when testPara="var"))

Author(s)
Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yue-jiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples
```r
# generate simulated data set from t distribution
set.seed(1234567)
```
iAWvar.BF

Test for equality of variance based on improved Ahn and Wang's (2013) score test

Description

Test for equality of variance based on improved Ahn and Wang's (2013) score test.

Usage

iAWvar.BF(value, group)

Arguments

value numeric. Measurements to be compared between two groups.

group numeric. Subject's group membership. Must be binary (i.e., taking values 0 or 1).

Value

A list with 6 elements:

U2 score statistic

varU2 estimated variance of the score statistic

T2 score statistic \( U_2^2 / \text{var}(U_2) \)

pval pvalue of the score test

z absolute deviation of value from median of value

zbar sample average of z

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>
References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(ncpgs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = iAWvar.BF(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
```

---

### iAWvar.Levene

Test for equality of variance based on improved Ahn and Wang's (2013) score test

---

Description

Test for equality of variance based on improved Ahn and Wang’s (2013) score test.

Usage

\[ iAWvar.Levene(value, group) \]

Arguments

- **value** numeric. Measurements to be compared between two groups.
- **group** numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).
Value

A list with 6 elements:

- \( U_2 \) score statistic
- \( \text{var}(U_2) \) estimated variance of the score statistic
- \( T_2 \) score statistic \( U_2^2 / \text{var}(U_2) \)
- \( p\text{val} \) p-value of the score test
- \( z \) absolute deviation of value from mean value
- \( z\text{bar} \) sample average of \( z \)

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yue-jiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(ncpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = iAWvar.Levene(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
```
iAWvar.TrimMean

Test for equality of variance based on improved Ahn and Wang’s (2013) score test

Description

Test for equality of variance based on improved Ahn and Wang’s (2013) score test.

Usage

iAWvar.TrimMean(value, group, trim.alpha = 0.25)

Arguments

- value: numeric. Measurements to be compared between two groups.
- group: numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).
- trim.alpha: numeric. Indicating proportion of data points to be trimmed from both ends before calculating sample mean.

Value

A list with 6 elements:

- u2: score statistic
- varU2: estimated variance of the score statistic
- t2: score statistic $U_2^2 / \text{var}(U_2)$
- pval: pvalue of the score test
- z: absolute deviation of value from trimmed-mean of value
- zbar: sample average of z

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.TDistr(ncpgs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = iAWVar.TrimMean(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
```

**jointKSTestWrapper**

*Wrapper function to test for equality of two distributions by using KS test for whole genome*

**Description**

Wrapper function to test for equality of two distributions by using KS test for whole genome

**Usage**

```r
jointKSTestWrapper(
es,
grpVar = "group",
esFlag = "es",
pvalAdjMethod = "fdr",
alpha = 0.05,
nTop = 20,
probeID.var = "ProbeID",
gene.var = "Symbol",
chr.var = "Chromosome",
applier = lapply,
verbose = FALSE)
```

**Arguments**

- `es` An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.
- `grpVar` character string. The name of the phenotype variable indicating arrays’ group membership. 0 means control and 1 means case.
esFlag character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.

pvalAdjMethod character string. Indicating which p-value adjustment will be used to control for multiple testing.

alpha numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially variable.

nTop integer. Specifying the number of top probes to be displayed if verbose=TRUE

probeID.var character string. Feature annotation variable indicating probe ID.

gene.var character string. Feature annotation variable indicating gene symbol.

chr.var character string. Feature annotation variable indicating chromosome number.

applier function name to do apply operation.

verbose logical. Indicating if intermediate results should be output to screen.

Value

A list of 2 elements. The first element frame is unsorted data frame; the second element frame.s is a sorted data frame object storing the analysis results and containing the following columns: probe (probe id), stat (test statistic), pval (raw p-value), p.adj (adjusted p-value), gene (gene symbol), chr (chromosome number), and pos (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "both",
eps = 1.0e-3, applier = lapply)
```r
print(es.sim)

res.jointKstestWrapper = jointKstestWrapper(
  es = es.sim,
  grpVar = "memSubj",
  esFlag = "es",
  pvalAdjMethod = "fdr",
  alpha = 0.05,
  nTop = 20,
  probeID.var = "probe",
  gene.var = "gene",
  chr.var = "chr",
  applyl=lapply,
  verbose=TRUE)
```

---

**jointLRTTestWrapper**  
*Wrapper function to test for equality of mean or variance by using likelihood ratio test for whole genome*

---

**Description**

Wrapper function to test for equality of mean or variance by using likelihood ratio test for whole genome.

**Usage**

```r
jointLRTTestWrapper(
  es,
  grpVar = "group",
  esFlag = "es",
  pvalAdjMethod = "fdr",
  alpha = 0.05,
  nTop = 20,
  probeID.var = "ProbeID",
  gene.var = "Symbol",
  chr.var = "Chromosome",
  applyl = lapply,
  verbose = FALSE)
```

**Arguments**

- **es**  
  An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.

- **grpVar**  
  character string. The name of the phenotype variable indicating arrays’ group membership. 0 means control and 1 means case.
esFlag character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.

pvalAdjMethod character string. Indicating which p-value adjustment will be used to control for multiple testing.

alpha numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially variable.

nTop integer. Specifying the number of top probes to be displayed if verbose=TRUE

probeID.var character string. Feature annotation variable indicating probe ID.

gene.var character string. Feature annotation variable indicating gene symbol.

chr.var character string. Feature annotation variable indicating chromosome number.

applier function name to do apply operation.

verbose logical. Indicating if intermediate results should be output to screen.

Value

A list of 2 elements. The first element frame is unsorted data frame; the second element frame.s is a sorted data frame object storing the analysis results and containing the following columns: probe (probe id), stat (test statistic), pval (raw p-value), p.adj (adjusted p-value), gene (gene symbol), chr (chromosome number), and pos (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpGs = 100, nCases = 20, nControls = 20, 
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "both", 
es = 1.0e-3, applier = lapply)
Wrapper function to test for equality of mean or variance simultaneously by using score tests.

Description

Wrapper function to test for equality of mean or variance simultaneously by using score tests.

Usage

```r
jointScoreTestsWrapper(
  es,
  grpVar = "group",
  meanTestFunc = scoreTestMean,
  varTestFunc = AWVvarTest,
  esFlag = "es",
  pvalAdjMethod = "fdr",
  alpha = 0.05,
  nTop = 20,
  probeID.var = "ProbeID",
  gene.var = "Symbol",
  chr.var = "Chromosome",
  applier = lapply,
  verbose = FALSE)
```

Arguments

- `es` : An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.
grpVar character string. The name of the phenotype variable indicating arrays’ group membership. 0 means control and 1 means case.

meanTestFunc A user-defined function to test for equal mean. Available functions provided by diffMeanVar package include: scoreTestMean. The function should contain 2 and only 2 inputs: value and group, where group is a binary vector indicating if a subject is a case (group=1) or control (group=0) and value is a continuous-type variable. The goal of the function is to test if the mean of value for cases is the same as that for controls. The function should return a list object containing at least 2 elements: stat and pval. For each probe stored in es, the function will be applied to test if the probe is differentially expressed between cases and controls.

varTestFunc A user-defined function to test for equality of variance. Available functions provided by diffMeanVar package include: iawvar.BF, iawvar.Levene, iawvar.TrimMean, AVarTest, BFTest, FTest, LeveneTest, and TrimMeanLeveneTest. The function should contain 2 and only 2 inputs: value and group, where group is a binary vector indicating if a subject is a case (group=1) or control (group=0) and value is a continuous-type variable. The goal of the function is to test if the variance of value for cases is the same as that for controls. The function should return a list object containing at least 2 elements: stat and pval. For each probe stored in es, the function will be applied to test if the probe is differentially variable between cases and controls.

esFlag character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.

pvalAdjMethod character string. Indicating which p-value adjustment will be used to control for multiple testing.

alpha numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially variable.

nTop integer. Specifying the number of top probes to be displayed if verbose=TRUE

probeID.var character string. Feature annotation variable indicating probe ID.

gene.var character string. Feature annotation variable indicating gene symbol.

chr.var character string. Feature annotation variable indicating chromosome number.

applier function name to do apply operation.

verbose logical. Indicating if intermediate results should be output to screen.

Value

A list of 2 elements. The first element frame is unsorted data frame; the second element frame.s is a sorted data frame object storing the analysis results and containing the following columns: probe (probe id), stat (test statistic), pval (raw p-value), p. adj (adjusted p-value), gene (gene symbol), chr (chromosome number), and pos (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>
LeveneTest

Test for equality of variance based on Levene test

Description

Test for equality of variance based on Levene test.

Usage

LeveneTest(value, group)
**LeveneTest**

**Arguments**

- **value** numeric. Measurements to be compared between two groups.
- **group** numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).

**Value**

A list with 2 elements:

- **stat** test statistic value
- **pval** pvalue of the score test

**Author(s)**

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yue-jiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

**References**


**Examples**

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(ncpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
pDat(pDat[1:2,])

res = LeveneTest(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
```
**POTestWrapper**

*Wrapper function to test for equality of variance by using Phipson and Oshlack’s (2014) methods*

**Description**

Wrapper function to test for equality of variance by using Phipson and Oshlack’s (2014) methods

**Usage**

```r
POTestWrapper(
  es,
  grpVar = "group",
  type = "AD",
  esFlag = "es",
  pvalAdjMethod = "fdr",
  alpha = 0.05,
  nTop = 20,
  probeID.var = "ProbeID",
  gene.var = "Symbol",
  chr.var = "Chromosome",
  applier = lapply,
  verbose = FALSE)
```

**Arguments**

- `es` An ExpressionSet object storing gene expression/DNA methylation data, phenotype data, and feature annotation.
- `grpVar` character string. The name of the phenotype variable indicating arrays’ group membership. 0 means control and 1 means case.
- `type` character string indicating if AD method or SQ method would be used.
- `esFlag` character string. Indicating if es is an ExpressionSet object or MethylSet object. The program will use exprs function to extract gene expression data or use betas function to extract methylation data.
- `pvalAdjMethod` character string. Indicating which p-value adjustment will be used to control for multiple testing.
- `alpha` numeric. Cutoff for p-value or adjusted p-value to determine if a probe is differentially variable.
- `nTop` integer. Specifying the number of top probes to be displayed if verbose=TRUE
- `probeID.var` character string. Feature annotation variable indicating probe ID.
- `gene.var` character string. Feature annotation variable indicating gene symbol.
- `chr.var` character string. Feature annotation variable indicating chromosome number.
- `applier` function name to do apply operation.
- `verbose` logical. indicating if intermediate results should be output to screen.
Value

A list of 2 elements. The first element `frame` is unsorted data frame; the second element `frame.s` is a sorted data frame object storing the analysis results and containing the following columns: `probe` (probe id), `stat` (test statistic), `pval` (raw p-value), `p.adj` (adjusted p-value), `gene` (gene symbol), `chr` (chromosome number), and `pos` (position of a probe in unsorted data frame).

The data frame is sorted based on the descending order of the absolute value of the test statistic.

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yue-jiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

```r
# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpgs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 5, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)

res.POTestWrapper = POTestWrapper(
es = es.sim,
grpVar = "memSubj",
type = "SQ",
esFlag = "es",
pvalAdjMethod = "fdr",
alpha = 0.05,
nTop = 20,
probeID.var = "probe",
gene.var = "gene",
chr.var = "chr",
applier=lapply,
verbose=TRUE)
```

**scoreTestMean**

*Test for equality of mean based on the score test of logistic regression*

**Description**

Test for equality of mean based on the score test of logistic regression.
Usage

scoreTestMean(value, group)

Arguments

value numeric. Measurements to be compared between two groups.
group numeric. Subject’s group membership. Must be binary (i.e., taking values 0 or 1).

Value

A list with 6 elements:

- U1 score statistic
- varU1 estimated variance of the score statistic
- T1 score statistic $U_1^2 / \text{var}(U_1)$
- pval p-value of the score test
- x equal to the input value
- xbar sample average of x

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yuejiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpGs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applicer = lapply)
print(es.sim)
print(exprs(es.sim)[1:2,1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = scoreTestMean(value = dat[,1], group = pDat$memSubj)
print(names(res))
print(res)
TrimMeanLeveneTest

Test for equality of variance based on trimmed-mean based Levene test

Description

Test for equality of variance based on trimmed-mean based Levene test.

Usage

TrimMeanLeveneTest(value, group, trim.alpha = 0.25)

Arguments

value numeric. Measurements to be compared between two groups.

Group numeric. Subject's group membership. Must be binary (i.e., taking values 0 or 1).

Trim.alpha numeric. Indicating proportion of data points to be trimmed from both ends before calculating sample mean.

Value

A list with 2 elements:

stat test statistic value

pval pvalue of the score test

Author(s)

Xuan Li <lixuan0759@mathstat.yorku.ca>, Weiliang Qiu <stwxq@channing.harvard.edu>, Yue-jiao Fu <yuejiao@mathstat.yorku.ca>, Xiaogang Wang <stevenw@mathstat.yorku.ca>

References


Examples

# generate simulated data set from t distribution
set.seed(1234567)
es.sim = genSimData.tDistr(nCpgs = 100, nCases = 20, nControls = 20,
df0 = 10, ncp0 = 0, df1 = 6, ncp1 = 2.393, testPara = "var",
eps = 1.0e-3, applier = lapply)
print(es.sim)
print(exprs(es.sim)[1:2:1:3])

# do AW score test for the first probe
dat = exprs(es.sim)
pDat = pData(es.sim)
print(pDat[1:2,])

res = TrimMeanLeveneTest(value = dat[1,], group = pDat$memSubj)
print(names(res))
print(res)
Index

**Topic method**

AWVarTest, 2
BFtest, 3
equalMeanTestWrapper, 4
equalVarTestWrapper, 7
FTest, 9
genSimData.tDistr, 10
iAWVar.BF, 12
iAWVar.Levene, 13
iAWVar.TrimMean, 15
jointKSTestWrapper, 16
jointLRTTestWrapper, 18
jointScoreTestsWrapper, 20
LeveneTest, 22
POTestWrapper, 24
scoreTestMean, 25
TrimMeanLeveneTest, 27