R topics documented:

hist ................................................................. 1
PilotData-class .................................................. 2
pilotData ......................................................... 3
plotEffectSize ................................................... 4
plot ............................................................... 5
Power .................................................................. 5
SampleSize-class .................................................. 7
sampleSize ........................................................ 8
show-methods ...................................................... 10
Internal functions ................................................ 10

Index 11

hist  Histogram of p-values of the pilot data

Description

The function `hist` computes a histogram of p-values of the given pilot data. Two-sided p-values are computed either using the normal or Student t distribution based on the test statistics that are stored in an object of `PilotData`.

Usage

`hist(x, ...)`

Arguments

- `x` object of `PilotData`
- `...` additional arguments to `hist`

Details

p-values calculated under the null hypothesis (non-differentially expressed genes) are assumed to be uniformly distributed on [0,1]. p-values calculated under the alternative hypothesis (differentially expressed genes) are assumed to accummulated near zero. The height of the flat part of the histogram is an indication of how many hypothesis are calculated under the null hypothesis.
PilotData-class

Value

an object of class "histogram", see hist.

Author(s)

Maarten van Iterson

See Also

hist and plot

Examples

library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)
pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
hist(pd)

Description

The "PilotData" class is a container class. It contains all necessary information of the pilot data for performing the power and sample size analysis.

Objects from the Class

Objects can be created by calling the function pilotData.

Slots

name: name of the pilot data, type "character"

testStatistics: vector of test statistics, type "numeric"

sampleSize: effective sample size \( \left( \frac{1}{n_A} + \frac{1}{n_B} \right)^{-1} \) with \( n_A \) the sample size in group A, type "numeric"

pValues: vector of p-values, type "numeric"

dof: degrees of freedom only used for Student t distribution, type "numeric"

nullDist: null distribution either normal or Student t, type "character"

Methods

show,PilotData-method, hist and plot

Author(s)

Maarten van Iterson

See Also

pilotData, SampleSize-class
pilotData

pilotData

Description

The function pilotData initializes a PilotData-object. Information of the pilot data and the null distribution is stored e.g. name of pilot experiment, test statistics and number of samples used.

Usage

pilotData(name = "Unknown Experiment", testStatistics = double(1), sampleSizeA = double(1), sampleSizeB = double(1), dof = double(0), nullDist = c("normal", "student"))

Arguments

name character string giving the experiment name
testStatistics vector of type numeric containing the set of test statistics
sampleSizeA the samplesize of group A
sampleSizeB the samplesize of group B
dof degree of freedom for a Student t distribution
nullDist distribution under the null hypothesis either one of:
  • "normal",
  • "student", (only in case of method="Ruppert" see sampleSize)

Details

Based on the given null distribution two-sided p-values are calculated from the test statistics. Some additional checks on the data are performed. Once an object of PilotData is created the sampleSize-function can be called.

Value

object of class PilotData

Author(s)

Maarten van Iterson

See Also

hist, plot and PilotData use class?PilotData

Examples

library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)

pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
plotEffectSize  

Plots the density of effect sizes of the pilot data

**Description**

The function *plotEffectSize* plots density of effect sizes of the pilot data.

**Usage**

```r
plotEffectSize(x, threshold = 0, xlab = "effect size", ylab = "density of effect sizes", main, sub, ...)```

**Arguments**

- `x`  
  object of class `SampleSize-class`
- `threshold`  
  threshold for truncation of the density of effect-sizes. The threshold will be taken symmetrical around the y-axis.
- `xlab`  
  a title for the x axis
- `ylab`  
  a title for the y axis
- `main`  
  an overall title for the plot
- `sub`  
  a sub title for the plot
- `...`  
  additional arguments given to `plot` or `par`

**Details**

The density of effect sizes describes the effects observed in the pilot data. Usually a bimodal density is observed representing up- and down-regulated genes. The way in which the test statistics is calculated determines what is meant by up- and down-regulation. A small symmetrical region around zero can be defined that will be excluded from the density of effect sizes and thereby increases the estimated average power.

**References**


**Examples**

```r
library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)
pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
hist(pd)
plot(pd)
ss <- sampleSize(pd)
plotEffectSize(ss)
```
plot

Empirical cumulative distribution function of p-values of the pilot data

Description

The function plot plots an empirical cumulative distribution function of p-values of the pilot data. The test statistics that are given as input to PilotData-class are transformed to two-sided p-values either using the normal or Student t distribution.

Usage

plot(x, y, ...)

Arguments

x  
object of class PilotData-class

y  
unused item

... 
additional arguments given to plot or par

Details

Empirical cumulative distribution function of p-values. The line at angle of 45 degrees represents the theoretical CDF of a uniform distribution as expected when all genes are non-differentially expressed. An accumulation of p-values near zero indicates a certain number of differentially expressed genes.

Author(s)

Maarten van Iterson

Examples

library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)
pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
hist(pd)
plot(pd)

---

Power

Estimates the average power of the pilot data

Description

The function Power estimates the average power of the pilot data at a given false discovery rate. The average power can be estimated for sample sizes other then the pilot data.
Usage

```r
Power(x, threshold = 0, fdr = 0.1, samplesizes = NULL, plot = FALSE, type = "l", ylim = c(0, 1), xlim = c(0, 1), xlab = "p-value", ylab = "average power", main, sub, ...)
```

Arguments

- **x**: object of class `SampleSize-class`
- **threshold**: threshold for truncation of the density of effect-sizes. A small symmetrical region around zero can be defined that will be excluded from the density of effect sizes.
- **fdr**: numeric given the desired false discovery rate threshold, possibly a vector like: `c(0.1, 0.05)`
- **samplesizes**: numeric vector, giving the samplesizes for which the power will be estimated
- **plot**: logical if `TRUE` plots the power curve and intersection for the given fdr (default `plot=FALSE`).
- **type**: what type of plot should be drawn
- **ylim**: range of y values
- **xlim**: range of x values
- **xlab**: a title for the x axis
- **ylab**: a title for the y axis
- **main**: an overall title for the plot
- **sub**: a sub title for the plot
- **...**: additional arguments to `plot` or `par`

Details

Having estimated the proportion of non-differentially expressed genes and the density of effect-sizes an estimate of the average power is calculated at a given false discovery rate and sample size. Power and sample size analysis can be performed by estimating the average power for sample sizes other then the pilot data. Ferreira and Zwinderman (2006) show that the average power (the proportion of rejected hypothesis out of all alternative hypothesis) of the adaptive Benjamini-Hochberg procedure is estimated by the point where the CDF of p-values under the alternative hypothesis intersects the straight line with a slope determined by the given FDR and the estimated proportion of non-differentially expressed genes. Using `plot=TRUE` the CDF of p-values under the alternative hypothesis is shown with the intersecting lines for different FDRs and sample sizes. Don’t use more than 2 FDRs and more than 3 different sample sizes because the plot will then be unclear.

Value

- estimated average power, probably a named matrix with on the rows different sample sizes and on the columns the different false discovery rates.

Author(s)

- Maarten van Iterson
SampleSize-class

References


Examples

```r
library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)
pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
hist(pd)
plot(pd)
ss <- sampleSize(pd)
plotEffectSize(ss)
Power(ss)

## creating a plot estimate power vs. sample size
pwr <- Power(ss, plot = FALSE, samplesizes = c(5, 10, 15, 20), fdr=0.01)
plot(c(5, 10, 15, 20), pwr, ylim = c(0, 1), type = "b", ylab = "Power", xlab = "Sample size")
legend("bottomright", colnames(pwr), col=c(1:ncol(pwr)), pch=1, lty=1)

## creating a plot estimate power vs. sample size for different false discovery rates
pwr <- Power(ss, plot = FALSE, samplesizes = c(5, 10, 15, 20), fdr=c(0.01, 0.05))
matplot(c(5, 10, 15, 20), pwr, ylim = c(0, 1), type = "b", pch=1, ylab = "Power", xlab = "Sample size")
legend("bottomright", colnames(pwr), col=c(1:ncol(pwr)), pch=1, lty=1)
```

SampleSize-class  Class SampleSize

Description

By contructing an object of SampleSize the power and sample size analysis will be performed.

Objects from the Class

Objects can be created by calling the function `sampleSize`.

Slots

- **pi0**: proportion of non-differentially expressed genes, "named-list"
- **lambda**: vector containing the estimated density of effect sizes, "numeric"
- **theta**: vector containing the range over which the density of effect sizes is estimated, "numeric"
- **adjust**: is the density of effect sizes adjusted (adjust=TRUE) or not (adjust=FALSE), "logical"
- **method**: method used for the estimation of the density of effect sizes, "character"
- **bandwidth**: Default bandwidth given by \(\log_{10}(n)^{-\frac{1}{4}}\) where \(n\) is the number of genes.
- **kernel**: kernel used for the deconvolution estimator, "character"
- **nKnots**: number of knots used for the B-splines (only visible if method="Ruppert")
**bDegree**: B-spline degree (only visible if `method="Ruppert"`)  
**name**: name of the pilot data, type "character"  
**testStatistics**: vector of test statistics, type "numeric"  
**sampleSize**: effective sample size \( \left( \frac{1}{n_A} + \frac{1}{n_B} \right)^{-1} \) with \( n_A \) the sample size in group A, type "numeric"  
**pValues**: vector of p-values, type "numeric"  
**dof**: degrees of freedom only used for Student t distribution, type "numeric"  
**nullDist**: null distribution either normal or Student t, type "character"

**Extends**
Class `PilotData`, directly.

**Methods**
- `show`, `SampleSize-method`, `Power` and `plotEffectSize`

**Author(s)**
Maarten van Iterson

**See Also**
`pilotData`, `PilotData`

---

**sampleSize**  
`sampleSize` creates an object of class `SampleSize`

**Description**
The function `sampleSize` initializes a `SampleSize`-object. The density of effect-sizes is estimated using a deconvolution estimator or constrained optimization using B-splines.

**Usage**
`sampleSize(PilotData, method = c("Langaas", "Storey", "Ferreira", "Ruppert", "Userdefined"), from = -6, to = 6, resolution = 2^10, kernel = c("fan", "wand", "sinc"), pi0 = seq(0.1, 0.99, 0.01), adjust = TRUE, nKnots = 11, bDegree = 3, ...)`

**Arguments**
- **PilotData**: object of class `PilotData` on which the sample size and power analysis will be performed  
- **method**: character string giving the method for estimation of the fraction of non-differentially expressed genes either one of:  
  - "Langaas" (default)  
  - "Storey"  
  - "Ferreira"  
  - "Ruppert"  
  - "Userdefined"  
- **from**: Lower bound of the density of effect-sizes (the range should be symmetric)
sampleSize

to

resolution

the number of points on which the density of effect-sizes will be estimated (must be a power of 2)

kernel

the kernel type used in the deconvolution estimator either one of:

• "fan" (default)
• "wand"
• "sinc"

pi0

numeric or a vector of type numeric giving the fraction of non-differentially expressed genes. If method="Userdefined" a userdefined pi0 is obligated. If method="Ferreira" a range of values should be given e.g. seq(0.01, 0.99, 0.01)

adjust

is the density of effect sizes adjusted (adjust=TRUE, default) or not (adjust=FALSE)

nKnots

the number of knots used in Rupperts method for estimating the proportion of non-differentially expressed genes and density of effect-sizes

bDegree

degree of B-spline basis used in Rupperts method for estimating the proportion of non-differentially expressed genes and density of effect-sizes

... Additional parameters for method for the estimation of the fraction of non-differentially expressed genes (Doesn’t work yet!)

Details

The sampleSize function performes the estimation of the proportion of non-differentially expressed genes using one of the three methods, "Langaas", "Storey", "Ferreira" if method="Userdefined" a userdefined pi0 is needed and estimation of the proportion of non-differentially expressed genes will be skipped. A deconvolution estimator is implemented using the Fast Fourier Transform Algorithm fft() for estimations of the density of effect sizes. If method="Ruppert" constrained optimization using B-splines is used, for this method two additional packages needs to be installed namely quadprog and splines. Both the proportion of non-differentially expressed genes and the density of effect sizes are estimated with Ruppert’s method. In contrast to the orginal method of Ruppert et al. (2007) we made a modification on the estimation of the density of effect sizes allowing for negative effect sizes as well.

Value

Object of class SampleSize

Author(s)

Maarten van Iterson

References


See Also

convert, qvalue and SampleSize use class?SampleSize

Examples

library(multtest)
data(golub)
teststat <- mt.teststat(golub, golub.cl)
table(golub.cl)
pd <- pilotData(name="golub", testStatistics=teststat, sampleSizeA=11, sampleSizeB=27)
hist(pd)
plot(pd)
ss <- sampleSize(pd)

show-methods  Nicely showing objects of class pilotData and sampleSize

Description

Nicely showing objects from "PilotData" and "SampleSize"

Methods

object = "PilotData"  object from class "PilotData"
object = "SampleSize"  object from class "SampleSize"

Internal functions  internal functions

Description

internal functions; these are not to be called by the user.
Index

*Topic classes
  pilotData, 3
  PilotData-class, 2
  sampleSize, 8
  SampleSize-class, 7

*Topic hplot
  hist, 1
  plot, 5
  plotEffectSize, 4
  Power, 5

*Topic methods
  Internal functions, 10
  show-methods, 10

bspline (Internal functions), 10
constrainedOptimization (Internal functions), 10
convest, 10
deconvolution (Internal functions), 10
Dn (Internal functions), 10
Gnhat (Internal functions), 10
Gntilde (Internal functions), 10
hist, 1, 1, 2
hist,PilotData-method(hist), 1
Hntilde (Internal functions), 10

Internal functions, 10
par, 4–6
PilotData, 1–3, 8, 10
pilotData, 2, 3, 8
PilotData-class, 5
PilotData-class, 2
plot, 1, 2, 4, 5, 6
plot,ANY, ANY-method(plot), 5
plot,PilotData,missing-method (plot), 5
plot-methods (plot), 5
plotEffectSize, 4, 8
Power, 5, 8

psreg (Internal functions), 10
qvalue, 10
SampleSize, 8–10
sampleSize, 3, 7, 8
SampleSize-class, 2, 4, 6
SampleSize-class, 7
show,PilotData-method, 2
show,PilotData-method (show-methods), 10
show,SampleSize-method, 8
show,SampleSize-method (show-methods), 10
show-methods, 10