Package 'brt'

May 1, 2018

Type Package

Title Biological Relevance Testing

Version 1.3.0

Date 2018-05-01

Author Le Zheng[aut], Peng Yu[aut, cre]

Maintainer Le Zheng <lzheng.chn@gmail.com>

License GPL (>= 2)

VignetteBuilder knitr

Suggests knitr, rmarkdown, reshape2, vsn, DESeq2, pasilla

Depends R (>= 3.2.0)

Imports stats, ggplot2

RoxygenNote 6.0.1

Description Analyses of large-scale -omics datasets commonly use p-values as the indicators of statistical significance. However, considering p-value alone neglects the importance of effect size (i.e., the mean difference between groups) in determining the biological relevance of a significant difference. Here, we present a novel algorithm for computing a new statistic, the biological relevance testing (BRT) index, in the frequentist hypothesis testing framework to address this problem.

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-01 04:32:09 UTC

R topics documented:

`brt.test` *BRT test*

Description

BRT test

Usage

```
brt.test(x, y, hi, lo = -hi, var.equal = T, log_pvalue = F)
```

Arguments

`x`	a (non-empty) numeric vector of data values.
`y`	a (non-empty) numeric vector of data values.
`hi`	upper bound of the shift range (i.e. signficant if outside the range)
`lo`	lower bound of the shift range (i.e. if hi=lo=0, return t.test)
`var.equal`	a logical variable indicating whether to treat the two variances as being equal.If TRUE then the pooled variance is used to estimate the variance otherwise the Satterhwaite approximation to the degrees of freedom is used.
`log_pvalue`	brt.value is returned in log scale.

Examples

```
x=rnorm(10, 0, 1)
y=rnorm(10, 8, 2)
brt.test(x, y, hi=3)
```

`dtavg` *Average of the Student t Distribution*

Description

Average of the Student t Distribution

Usage

```
dtavg(x, df, hi = 1, lo = -hi, n = as.integer(ceiling(abs(hi - lo) * 10)),
  log = FALSE)

ptavg(x, df, hi = 1, lo = -hi, n = as.integer(ceiling(abs(hi - lo) * 10)),
  lower.tail = TRUE, log = FALSE)
```

Arguments

x	a vector
df	degrees of freedom
hi	upper bound of the shift range
lo	lower bound of the shift range
n	the number of bins for interpolation
log	the probability is in log-scale
lower.tail	use lower tail probablity

Examples

```
x=seq(from=-10, to=10, length.out=100)
ggplot2::qplot(x, ptavg(x, df=3, hi=3), geom='line')
```

---

logmeanexp *Mean of Numbers in Log-Scale*

Description

Mean of Numbers in Log-Scale

Usage

```
logmeanexp(x)
```

Arguments

x	a numeric vector

---

logsumexp *Sum of Numbers in Log-Scale*

Description

Sum of Numbers in Log-Scale

Usage

```
logsumexp(x)
```

Arguments

x	a numeric vector

tpval *The P-value of a t Test Base on a t-statistic.*

Description

The P-value of a t Test Base on a t-statistic.

Usage

```
tpval(x, df, log = FALSE)
```

Arguments

x	a t statistic
df	degrees of freedom
log	the probability is in log-scale

Examples

```
tpval(1, df=3)
exp(tpval(1, df=3, log=TRUE))
tpval(Inf, df=3)
tpval(0, df=3)
```

tpvalavg *Average of The Student t Distribution*

Description

Average of The Student t Distribution

Usage

```
tpvalavg(coefficients, hi, lo = -hi, se, df, n = as.integer(ceiling(abs(hi -
  lo) * 10)), log = FALSE)
```

Arguments

coefficients	a vector
hi	upper bound of the shift range
lo	lower bound of the shift range
se	standard error
df	degrees of freedom
n	the number of bins for interpolation
log	the probability is in log-scale

Examples

```
x=seq(from=0, to=30, length.out=100)

data=do.call(
  rbind
  , lapply(
    seq_len(10)
    , function(cutoff)
      rbind(
        data.frame(x, pval=tpvalavg(x, hi=1, se=1, df=3), cutoff=cutoff)
        )
    )
  )

ggplot2::qplot(x, log(pval), data=data, color=as.factor(cutoff),
linetype=as.factor(cutoff), geom='line')
tpvalavg(1, hi=1, se=1, df=3)
exp(tpvalavg(1, hi=1, se=1, df=3, log=TRUE))
```

`tpvalint` *Hypothesis testing using the Student t Distribution with H0: lo <= mu <= hi*

Description

Hypothesis testing using the Student t Distribution with H0: lo <= mu <= hi

Usage

```
tpvalint(coefficients, hi, lo = -hi, se, df, log = FALSE)
```

Arguments

`coefficients`	a vector
`hi`	upper bound
`lo`	lower bound
`se`	standard error
`df`	degrees of freedom
`log`	the probability is in log-scale

Examples

```
x=seq(from=-30, to=30, length.out=100)

data=do.call(
  rbind
  , lapply(
```

```
    seq_len(10)
    , function(cutoff)
      rbind(
       data.frame(x, pval=tpvalint(x, lo=-cutoff, hi=cutoff, se=1, df=3), cutoff=cutoff)
       )
    )
  )

ggplot2::qplot(x, pval, data=data, color=as.factor(cutoff), linetype=as.factor(cutoff), geom='line')
```

`tpvaltreat` *Hypothesis testing using the Student t Distribution with H0: abs(mu) <= delta*

Description

Hypothesis testing using the Student t Distribution with H0: abs(mu) <= delta

Usage

```
tpvaltreat(coefficients, delta, se, df, log = FALSE)
```

Arguments

`coefficients`	a vector
`delta`	a postive cutoff
`se`	standard error
`df`	degrees of freedom
`log`	the probability is in log-scale

Examples

```
x=seq(from=-30, to=30, length.out=100)

data=do.call(
  rbind
  , lapply(
    seq_len(10)
    , function(delta)
      rbind(
        data.frame(x, pval=tpvaltreat(x, delta=delta, se=1, df=3), delta=delta)
        )
    )
  )

ggplot2::qplot(x, pval, data=data, color=as.factor(delta), linetype=as.factor(delta), geom='line')
```

Index