**Package ‘brt’**

May 1, 2018

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Biological Relevance Testing</td>
</tr>
<tr>
<td>Version</td>
<td>1.3.0</td>
</tr>
<tr>
<td>Date</td>
<td>2018-05-01</td>
</tr>
<tr>
<td>Author</td>
<td>Le Zheng[aut], Peng Yu[aut, cre]</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Le Zheng <a href="mailto:lzheng.chn@gmail.com">lzheng.chn@gmail.com</a></td>
</tr>
<tr>
<td>License</td>
<td>GPL (&gt;= 2)</td>
</tr>
<tr>
<td>VignetteBuilder</td>
<td>knitr</td>
</tr>
<tr>
<td>Suggests</td>
<td>knitr, rmarkdown, reshape2, vsn, DESeq2, pasilla</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.2.0)</td>
</tr>
<tr>
<td>Imports</td>
<td>stats, ggplot2</td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>6.0.1</td>
</tr>
</tbody>
</table>

**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:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Documented</th>
</tr>
</thead>
<tbody>
<tr>
<td>brt.test</td>
<td>2</td>
</tr>
<tr>
<td>dtavg</td>
<td>2</td>
</tr>
<tr>
<td>logmeanexp</td>
<td>3</td>
</tr>
<tr>
<td>logsumexp</td>
<td>3</td>
</tr>
<tr>
<td>tpval</td>
<td>4</td>
</tr>
<tr>
<td>tpvalavg</td>
<td>4</td>
</tr>
<tr>
<td>tpvalavg</td>
<td>5</td>
</tr>
<tr>
<td>tpvaltreat</td>
<td>6</td>
</tr>
</tbody>
</table>
# brt.test

**BRT test**

**Description**
BRT test

**Usage**

\[
\text{brt.test}(x, y, hi, lo = -hi, \text{var.equal} = \text{T}, \text{log\_pvalue} = \text{F})
\]

**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. significant 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 Satterwaite 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**

\[
\text{dtavg}(x, df, hi = 1, lo = -hi, n = \text{as.integer}\left(\text{ceiling}\left(\text{abs}\left(hi - lo\right) * 10\right)\right), \text{log} = \text{FALSE})
\]

\[
\text{ptavg}(x, df, hi = 1, lo = -hi, n = \text{as.integer}\left(\text{ceiling}\left(\text{abs}\left(hi - lo\right) * 10\right)\right), \text{lower\_tail} = \text{TRUE}, \text{log} = \text{FALSE})
\]
**logmeanexp**

**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 probability

**Examples**

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

---

**Description**

Mean of Numbers in Log-Scale

**Usage**

```r
logmeanexp(x)
```

**Arguments**

- `x` a numeric vector

---

**logsumexp**

**Description**

Sum of Numbers in Log-Scale

**Usage**

```r
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
tpvalint

Examples

\begin{verbatim}
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')

exp(tpvalavg(1, hi=1, se=1, df=3, log=TRUE))
\end{verbatim}

\begin{verbatim}

tpvalint

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(

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(

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 positive 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

*Topic **distribution**
  dtavg, 2
  tpval, 4
  tpvalavg, 4
*Topic **htest**
  tpvalint, 5
  tpvaltreat, 6
*Topic **math**
  logmeanexp, 3
  logsumexp, 3

brt.test, 2

dtavg, 2

logmeanexp, 3
logsumexp, 3

ptavg (dtavg), 2

tpval, 4
tpvalavg, 4
tpvalint, 5
tpvaltreat, 6