Package ‘sgpv’

July 6, 2020

Title Calculate Second-Generation p-Values and Associated Measures

Date 2020-06-23

Version 1.0.0

Maintainer Valerie Welty <valerie.welty@vanderbilt.edu>

Depends R (>= 3.3.3)

License MIT + file LICENSE

Description Computation of second-generation p-values as described in Blume et al. (2018) <doi:10.1371/journal.pone.0188299> and Blume et al. (2019) <doi:10.1080/00031305.2018.1 ditional functions which provide power and type I error calculations, create graphs (particularly suited for large-scale inference usage), and a function to estimate false discovery rates based on second-generation p-value inference.

Encoding UTF-8

LazyData false

RoxygenNote 7.1.0

NeedsCompilation no

Author Valerie Welty [aut, cre],
Thomas Stewart [aut],
Robert Greevy, Jr. [aut],
Lucy D'Agostino McGowan [aut],
Jeffrey Blume [aut]

Repository CRAN

Date/Publication 2020-07-06 13:40:05 UTC

R topics documented:

`fdrisk` *False Discovery Risk for Second-Generation p-Values*

Description

This function computes the false discovery risk (sometimes called the "empirical bayes FDR") for a second-generation *p*-value of 0, or the false confirmation risk for a second-generation *p*-value of 1.

Usage

```
fdrisk(
  sgpval = 0,
  null.lo,
  null.hi,
  std.err,
  interval.type,
  interval.level,
  pi0 = 0.5,
  null.weights,
  null.space,
  alt.weights,
  alt.space
)
```

Arguments

`sgpval` The observed second-generation *p*-value. Default is 0, which gives the false discovery risk.

`null.lo` The lower bound of the indifference zone (null interval) upon which the second-generation *p*-value was based

`null.hi` The upper bound for the indifference zone (null interval) upon which the second-generation *p*-value was based

`std.err` Standard error of the point estimate

`interval.type` Class of interval estimate used. This determines the functional form of the power function. Options are `confidence` for a $(1-\alpha)100\%$ confidence interval and `likelihood` for a $1/k$ likelihood support interval (`credible` not yet supported).

`interval.level` Level of interval estimate. If `interval.type` is `confidence`, the level is α. If `interval.type` is `likelihood`, the level is $1/k$ (not k).

`pi0` Prior probability of the null hypothesis. Default is 0.5.

`null.weights` Probability distribution for the null parameter space. Options are currently `Point`, `Uniform`, and `TruncNormal`.

`null.space` Support of the null probability distribution. If `null.weights` is `Point`, then `null.space` is a scalar. If `null.weights` is `Uniform`, then `null.space` is a vector of length two.

`alt.weights` Probability distribution for the alternative parameter space. Options are currently `Point`, `Uniform`, and `TruncNormal`.

`alt.space` Support for the alternative probability distribution. If `alt.weights` is `Point`, then `alt.space` is a scalar. If `alt.weights` is `Uniform`, then `alt.space` is a vector of length two.

Details

When possible, one should compute the second-generation p-value and FDR/FCR on a scale that is symmetric about the null hypothesis. For example, if the parameter of interest is an odds ratio, inputs `pt.est`, `std.err`, `null.lo`, `null.hi`, `null.space`, and `alt.space` are typically on the log scale.

If `TruncNormal` is used for `null.weights`, then the distribution used is a truncated Normal distribution with mean equal to the midpoint of `null.space`, and standard deviation equal to `std.err`, truncated to the support of `null.space`. If `TruncNormal` is used for `alt.weights`, then the distribution used is a truncated Normal distribution with mean equal to the midpoint of `alt.space`, and standard deviation equal to `std.err`, truncated to the support of `alt.space`. Further customization of these parameters for the truncated Normal are currently not possible, although they may be implemented in future versions.

Value

Numeric scalar representing the False discovery risk (FDR) or false confirmation risk (FCR) for the observed second-generation p-value. If `sgpval` = 0, the function returns false discovery risk (FDR). If `sgpval` = 1, the function returns false confirmation risk (FCR).

References

Blume JD, Greevy RA Jr., Welty VF, Smith JR, Dupont WD (2019). An Introduction to Second-generation p-values. *The American Statistician*. 73:sup1, 157-167, DOI: https://doi.org/10.1080/00031305.2018.1537893

Blume JD, D'Agostino McGowan L, Dupont WD, Greevy RA Jr. (2018). Second-generation p-values: Improved rigor, reproducibility, & transparency in statistical analyses. *PLoS ONE* 13(3): e0188299. https://doi.org/10.1371/journal.pone.0188299

See Also

`sgpvalue`,`sgpower`,`plotsgpv`

Examples

```
# false discovery risk with 95% confidence level
fdrisk(sgpval = 0,  null.lo = log(1/1.1), null.hi = log(1.1),  std.err = 0.8,
  null.weights = 'Uniform', null.space = c(log(1/1.1), log(1.1)),
  alt.weights = 'Uniform',  alt.space = 2 + c(-1,1)*qnorm(1-0.05/2)*0.8,
  interval.type = 'confidence',  interval.level = 0.05)

# false discovery risk with 1/8 likelihood support level
fdrisk(sgpval = 0,  null.lo = log(1/1.1), null.hi = log(1.1),  std.err = 0.8,
  null.weights = 'Point', null.space = 0,  alt.weights = 'Uniform',
  alt.space = 2 + c(-1,1)*qnorm(1-0.041/2)*0.8,
  interval.type = 'likelihood',  interval.level = 1/8)

## with truncated normal weighting distribution
fdrisk(sgpval = 0,  null.lo = log(1/1.1), null.hi = log(1.1),  std.err = 0.8,
  null.weights = 'Point', null.space = 0,  alt.weights = 'TruncNormal',
  alt.space = 2 + c(-1,1)*qnorm(1-0.041/2)*0.8,
  interval.type = 'likelihood',  interval.level = 1/8)

# false discovery risk with LSI and wider null hypothesis
fdrisk(sgpval = 0,  null.lo = log(1/1.5), null.hi = log(1.5),  std.err = 0.8,
  null.weights = 'Point', null.space = 0,  alt.weights = 'Uniform',
  alt.space = 2.5 + c(-1,1)*qnorm(1-0.041/2)*0.8,
  interval.type = 'likelihood',  interval.level = 1/8)

# false confirmation risk example
fdrisk(sgpval = 1,  null.lo = log(1/1.5), null.hi = log(1.5),  std.err = 0.15,
  null.weights = 'Uniform', null.space = 0.01 + c(-1,1)*qnorm(1-0.041/2)*0.15,
  alt.weights = 'Uniform',  alt.space = c(log(1.5), 1.25*log(1.5)),
  interval.type = 'likelihood',  interval.level = 1/8)
```

leukstats *Test Statistics from Gloub (1999) Leukemia data set*

Description

Data are from 7218 gene specific t-tests for a difference in mean expression (on the log scale; AML versus ALL) in the Gloub data set (1999). Data are from 72 patients using a pooled t-test (df=70). Included in the dataframe are the following: t-statistic (`t.stat`), p-value (`p.value`), CI lower limit (`ci.lo`), CI upper limit (`ci.hi`), estimate (`estimate`), standard error (`se`).

Usage

```
data(leukstats)
```

Format

An object of class `data.frame`. Includes the following: t-statistic (`t.stat`), p-value (`p.value`), CI lower limit (`ci.lo`), CI upper limit (`ci.hi`), estimate (`estimate`), standard error (`se`).

Source

https://github.com/ramhiser/datamicroarray/wiki/Golub-(1999)

References

Gloub (1999) and used in Blume et. al. (2018) PlosONE.

Blume JD, D'Agostino McGowan L, Dupont WD, Greevy RA Jr. (2018). Second-generation *p*-values: Improved rigor, reproducibility, & transparency in statistical analyses. *PLoS ONE* 13(3): e0188299. https://doi.org/10.1371/journal.pone.0188299

Examples

```
data(leukstats)
order(leukstats$p.value)
```

plotsgpv *Second-Generation p-Value Plotting*

Description

This function displays user supplied interval estimates (support intervals, confidence intervals, credible intervals, etc.) according to its associated second-generation *p*-value ranking.

Usage

```
plotsgpv(
  est.lo,
  est.hi,
  null.lo,
  null.hi,
  set.order = "sgpv",
  x.show = NA,
  null.col = rgb(208, 216, 232, maxColorValue = 255),
  int.col = c("cornflowerblue", "firebrick3", "darkslateblue"),
  int.pch = NA,
  int.cex = 0.4,
  plot.axis = c(TRUE, TRUE),
  null.pt = NA,
  outline.zone = TRUE,
  title.lab = "Title",
  x.lab = "Position (by set.order)",
  y.lab = "Outcome label",
  legend.on = TRUE
)
```

Arguments

`est.lo` A numeric vector of lower bounds of interval estimates. Values must be finite for interval to be drawn. Must be of same length as `est.hi`.

`est.hi` A numeric vector of upper bounds of interval estimates. Values must be finite for interval to be drawn. Must be of same length as `est.lo`.

`null.lo` A scalar representing the lower bound of null interval (indifference zone). Value must be finite.

`null.hi` A scalar representing the upper bound of null interval (indifference zone). Value must be finite.

`set.order` A numeric vector giving the desired order along the x-axis. If `set.order` is set to `sgpv`, the second-generation *p*-value ranking is used. If `set.order` is set to `NA`, the original input ordering is used.

`x.show` A scalar representing the maximum ranking on the x-axis that is displayed. Default is to display all intervals.

`null.col` Coloring of the null interval (indifference zone). Default is Hawkes Blue: `rgb(208,216,232,maxColorVa`

`int.col` Coloring of the intervals according to SGPV ranking. Default is `c("cornflowerblue","firebrick3","` for SGPVs of 0, in $(0, 1)$, and 1 respectively.

`int.pch` Plotting symbol for interval endpoints. Default is `NA`, no symbol. Use `16` for small endpoints.

`int.cex` Size of plotting symbol for interval endpoints. Default is 0.4.

`plot.axis` Toggle for default axis plotting. Default is `c(TRUE,TRUE)` for $(x - axis, y - axis)$ respectively.

`null.pt` A scalar representing a point null hypothesis. Default is `NA`. If set, the function will draw a horizontal dashed black line at this location.

`outline.zone` Toggle for drawing a slim white outline around the null zone. Helpful visual aid when plotting many intervals. Default is `TRUE`.

`title.lab` Title text.

`x.lab` x-axis label.

`y.lab` y-axis label.

`legend.on` Toggle for plotting the legend. Default is `TRUE`.

Details

Use `set.order` to provide the classical p-value ranking. For example, if `pvalue.vector` is a vector of classical p-values, then set `set.order=order(pvalue.vector)` to sort the x-axis according to p-value rank.

Interval estimates with infinite or undefined limits should be manually truncated or avoided altogether. While the sgpvalue funciton will handle these cases, this function assumes they have been truncated or removed because there is no standard way to plot them.

References

Blume JD, Greevy RA Jr., Welty VF, Smith JR, Dupont WD (2019). An Introduction to Second-generation *p*-values. *The American Statistician*. 73:sup1, 157-167, DOI: https://doi.org/10.1080/00031305.2018.1537893

Blume JD, D'Agostino McGowan L, Dupont WD, Greevy RA Jr. (2018). Second-generation *p*-values: Improved rigor, reproducibility, & transparency in statistical analyses. *PLoS ONE* 13(3): e0188299. https://doi.org/10.1371/journal.pone.0188299

See Also

sgpvalue,sgpower,fdrisk

Examples

```
# Use leukstats data
data(leukstats)
plotsgpv(est.lo=leukstats$ci.lo, est.hi=leukstats$ci.hi,
null.lo=-0.3, null.hi=0.3,
set.order=order(leukstats$p.value),
x.show=7000,
plot.axis=c(”TRUE”,”FALSE”),
null.pt=0, outline.zone=TRUE,
title.lab="Leukemia Example", y.lab="Fold Change (base 10)”,
x.lab="Classical p-value ranking”,
legend.on=TRUE)
axis(side=2,at=round(log(c(1/1000,1/100,1/10,1/2,1,2,10,100,1000),
base=10),2),labels=c(”1/1000”,”1/100”,”1/10”,”1/2”,1,2,10,100,1000),
las=2)
```

sgpower *Power functions for Second-Generation p-Values*

Description

Calculate power and type I error values from significance testing based on second-generation p-values as the inferential metric.

Usage

```
sgpower(true, null.lo, null.hi, std.err = 1, interval.type, interval.level)
```

Arguments

`true` The true value for the parameter of interest at which to calculate power. Note that this is on the absolute scale of the parameter, and not the standard deviation or standard error scale.

`null.lo` The lower bound of the indifference zone (null interval) upon which the second-generation *p*-value is based

`null.hi` The upper bound for the indifference zone (null interval) upon which the second-generation *p*-value is based

`std.err` Standard error for the distribution of the estimator for the parameter of interest. Note that this is the standard deviation for the estimator, not the standard deviation parameter for the data itself. This will be a function of the sample size(s).

`interval.type` Class of interval estimate used for calculating the SGPV. Options are `confidence` for a $(1-\alpha)100\%$ confidence interval and `likelihood` for a $1/k$ likelihood support interval (`credible` not yet supported)

`interval.level` Level of interval estimate. If `interval.type` is `confidence`, the level is α. If `interval.type` is `likelihood`, the level is $1/k$ (not k).

Value

A list containing the following components:

`power.alt` Probability of SGPV = 0 calculated assuming the parameter is equal to `true`. That is, `power.alt`= $P(SGPV = 0|\theta =$ `true`).

`power.inc` Probability of 0 < SGPV < 1 calculated assuming the parameter is equal to `true`. That is, `power.inc`= $P(0 < SGPV < 1|\theta =$ `true`).

`power.null` Probability of SGPV = 1 calculated assuming the parameter is equal to `true`. That is, `power.null`= $P(SGPV = 1|\theta =$ `true`).

`‘type I error summaries‘` Named vector that includes different ways the type I error may be summarized for an interval null hypothesis. `min` is the minimum type I error over the range (`null.lo`, `null.hi`), which occurs at the midpoint of (`null.lo`, `null.hi`). `max` is the maximum type I error over the range (`null.lo`, `null.hi`), which occurs at the boundaries of the null hypothesis, `null.lo` and `null.hi`. `mean` is the average type I error (unweighted) over the range (`null.lo`, `null.hi`). If 0 is included in the null hypothesis region, then `` `type I error summaries` `` also contains `at 0`, the type I error calculated assuming the true parameter value θ is equal to 0.

References

Blume JD, Greevy RA Jr., Welty VF, Smith JR, Dupont WD (2019). An Introduction to Second-generation *p*-values. *The American Statistician*. 73:sup1, 157-167, DOI: https://doi.org/10.1080/00031305.2018.1537893

Blume JD, D'Agostino McGowan L, Dupont WD, Greevy RA Jr. (2018). Second-generation *p*-values: Improved rigor, reproducibility, & transparency in statistical analyses. *PLoS ONE* 13(3): e0188299. https://doi.org/10.1371/journal.pone.0188299

See Also

`fdrisk`,`sgpvalue`,`plotsgpv`

Examples

```
sgpower(true=2, null.lo=-1, null.hi=1, std.err=1, interval.type='confidence',
 'interval.level'=0.05)

sgpower(true=0, null.lo=-1, null.hi=1, std.err=1, interval.type='confidence',
 'interval.level'=0.05)

# plot the power curve
sigma = 5
n = 20
theta = seq(-10, 10, by=0.1)
```

```
power = sgpower(true=theta, null.lo=-1, null.hi=1, std.err=sigma/sqrt(n),
 interval.type='confidence', interval.level=0.05)$power.alt
plot(theta, power, type='l', ylab='power')
```

sgpvalue *Second-Generation p-Values*

Description

This function computes the second-generation *p*-value (SGPV) and its associated delta gaps, as introduced in Blume et al. (2018).

Usage

```
sgpvalue(
  est.lo,
  est.hi,
  null.lo,
  null.hi,
  inf.correction = 1e-05,
  warnings = TRUE
)
```

Arguments

`est.lo` A numeric vector of lower bounds of interval estimates. Values may be finite or `-Inf` or `Inf`. Must be of same length as `est.hi`.

`est.hi` A numeric vector of upper bounds of interval estimates. Values may be finite or `-Inf` or `Inf`. Must be of same length as `est.lo`.

`null.lo` A numeric vector of lower bounds of null intervals. Values may be finite or `-Inf` or `Inf`. Must be of same length as `null.hi`.

`null.hi` A numeric vector of upper bounds of null intervals. Values may be finite or `-Inf` or `Inf`. Must be of same length as `null.lo`.

`inf.correction` A small scalar to denote a positive but infinitesimally small SGPV. Default is 1e-5. SGPVs that are infinitesimally close to 1 are assigned `1-inf.correction`. This option can only be invoked when one of the intervals has infinite length.

`warnings` Warnings toggle. Warnings are on by default.

Details

Values of `NA` or `NaN` for `est.lo`, `est.hi`, `null.lo`, or `null.lo` will yield a warning and result in a SGPV of `NA` or `NaN`.

When `null.hi` and `null.lo` are of length 1, the same null interval is used for every interval estimate of [`est.lo`, `est.hi`]. If `null.hi` is not of length 1, its length must match that of `est.hi`.

When possible, one should compute the second-generation *p*-value on a scale that is symmetric about the null hypothesis. For example, if the parameter of interest is an odds ratio, computations are typically done on the log scale. This keeps the magnitude of positive and negative delta-gaps comparable. Also, recall that the delta-gaps magnitude is not comparable across different null intervals.

Value

A list containing the following components:

`p.delta` Vector of second-generation p-values

`delta.gap` Vector of delta-gaps. Reported as `NA` when the corresponding second-generation p-value is not zero.

References

Blume JD, Greevy RA Jr., Welty VF, Smith JR, Dupont WD (2019). An Introduction to Second-generation *p*-values. *The American Statistician*. 73:sup1, 157-167, DOI: https://doi.org/10.1080/00031305.2018.1537893

Blume JD, D'Agostino McGowan L, Dupont WD, Greevy RA Jr. (2018). Second-generation *p*-values: Improved rigor, reproducibility, & transparency in statistical analyses. *PLoS ONE* 13(3): e0188299. https://doi.org/10.1371/journal.pone.0188299

See Also

`fdrisk`,`sgpower`,`plotsgpv`

Examples

```
## Simple example for three estimated log odds ratios but the same null interval
lb <- c(log(1.05), log(1.3), log(0.97))
ub <- c(log(1.8), log(1.8), log(1.02))
sgpv <- sgpvalue(est.lo = lb, est.hi = ub, null.lo = log(1/1.1), null.hi = log(1.1))
sgpv$p.delta

sgpv$delta.gap

## Works with infinte interval bounds
sgpvalue(est.lo = log(1.3), est.hi = Inf, null.lo = -Inf, null.hi = log(1.1))


sgpvalue(est.lo = log(1.05), est.hi = Inf, null.lo = -Inf, null.hi = log(1.1))



## Example t-test with simulated data
set.seed(1776)
x1 <- rnorm(15,mean=0,sd=2) ; x2 <- rnorm(15,mean=3,sd=2)
ci <- t.test(x1,x2)$conf.int[1:2]
sgpvalue(est.lo = ci[1], est.hi = ci[2], null.lo = -1, null.hi = 1)
```

```
set.seed(2019)
x1 <- rnorm(15,mean=0,sd=2) ; x2 <- rnorm(15,mean=3,sd=2)
ci <- t.test(x1,x2)$conf.int[1:2]
sgpvalue(est.lo = ci[1], est.hi = ci[2], null.lo = -1, null.hi = 1)

## Simulated two-group dichotomous data for different parameters
set.seed(1492)
n1 <- n2 <- 30
x1 <- rbinom(1,size=n1,p=0.15) ; x2 <- rbinom(1,size=n2,p=0.50)

# On the difference in proportions
ci.p  <- prop.test(c(x1,x2),n=c(n1,n2))$conf.int[1:2]
sgpvalue(est.lo = ci.p[1], est.hi = ci.p[2], null.lo = -0.2, null.hi = 0.2)

# On the log odds ratio scale
a <- x1 ; b <- x2 ; c <- n1-x1 ; d <- n2-x2
ci.or <- log(a*d/(b*c)) + c(-1,1)*1.96*sqrt(1/a+1/b+1/c+1/d) # Delta-method SE for log odds ratio
sgpvalue(est.lo = ci.or[1], est.hi = ci.or[2], null.lo = log(1/1.5), null.hi = log(1.5))
```

Index