Package 'SurrogateRank'

June 18, 2026

Type Package

Title Rank-Based Test to Evaluate a Surrogate Marker

Version 3.0

Description Uses a novel rank-based nonparametric approach to evaluate a surrogate marker in a small sample size setting. Details are described in Parast et al (2024) <doi:10.1093/biomtc/ujad035>, in Hughes A et al (2025) <doi:10.1002/sim.70241>, and in Hughes A et al (2026) <doi:10.48550/arXiv.2605.03819>. A tutorial for this package can be found at <https://www.laylaparast.com/surrogaterank> and a Shiny App implementing the package can be found at <https://parastlab.shinyapps.io/SurrogateRankApp/>.

License GPL

Encoding UTF-8

Imports stats, dplyr, ggplot2, pbmcapply, cowplot, tidyr, ComplexUpset, ggVennDiagram, tibble, glue, scales, MASS

Suggests roxygen2

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

Author Layla Parast [aut, cre], Arthur Hughes [aut]

Maintainer Layla Parast <parast@austin.utexas.edu>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-06-18 09:50:09 UTC

Contents

`delta.calculate` *Calculates the rank-based test statistic for Y and S and the difference, delta*

Description

Calculates the rank-based test statistic for Y and the rank-based test statistic for S and the difference, delta, along with corresponding standard error estimates

Usage

```
delta.calculate(full.data = NULL, yone = NULL, yzero = NULL, sone = NULL, szero = NULL)
```

Arguments

`full.data`	either full.data or yone, yzero, sone, szero must be supplied; if full data is supplied it must be in the following format: one observation per row, Y is in the first column, S is in the second column, treatment group (0 or 1) is in the third column.
`yone`	primary outcome, Y, in group 1
`yzero`	primary outcome, Y, in group 0
`sone`	surrogate marker, S, in group 1
`szero`	surrogate marker, S, in group 0

Value

u.y	rank-based test statistic for Y
u.s	rank-based test statistic for S
delta	difference, u.y-u.s
sd.u.y	standard error estimate of u.y
sd.u.s	standard error estimate of u.s
sd.delta	standard error estimate of delta

Author(s)

Layla Parast

Examples

```
data(example.data)
delta.calculate(yone = example.data$y1, yzero = example.data$y0, sone = example.data$s1,
szero = example.data$s0)
```

---

delta.calculate.extension

Calculate Delta: Difference in Rank-based Statistics for Two Outcomes

---

Description

This function calculates the difference in treatment effects on a univariate marker and on a continuous primary response. This extends the `delta.calculate()` function from the `SurrogateRank` package to the case where samples may be paired instead of independent, and where a two sided test is desired.

Usage

```
delta.calculate.extension(
  yone,
  yzero,
  sone,
  szero,
  alpha = 0.05,
  paired = FALSE
)
```

Arguments

yone	numeric vector of primary response values in the treated group.
yzero	numeric vector of primary response values in the untreated group.
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of untreated samples and p the number of candidates. Sample ordering must match exactly yzero.
alpha	significance level of test, default is 0.05
paired	logical flag giving if the data is independent or paired. If FALSE (default), samples are assumed independent. If TRUE, samples are assumed to be from a paired design. The pairs are specified by matching the rows of yone and sone to the rows of yzero and szero.

Details

This function estimates the difference (delta) between two rank-based statistics (e.g., Wilcoxon statistics or paired ranks) for a primary outcome and a surrogate, under either an independent or paired design.

Value

A list with the following elements:

- u.y: Rank-based test statistic for the primary outcome
- u.s: Rank-based test statistic for the surrogate
- delta.estimate: Estimated difference between outcome and surrogate statistics
- sd.u.y: Standard deviation of the outcome statistic
- sd.u.s: Standard deviation of the surrogate statistic
- sd.delta: Standard error of the delta estimate

Author(s)

Arthur Hughes, Layla Parast

Examples

```
# Load data
data("example.data")
yone <- example.data$y1
yzero <- example.data$y0
sone <- example.data$s1
szero <- example.data$s0
delta.calculate.extension.result <- delta.calculate.extension(
  yone, yzero, sone, szero,
  paired = TRUE
)
```

`delta.reml.meta` *Function to perform meta-analysis of summary statistics and hypothesis testing for a single marker*

Description

Function to perform meta-analysis of summary statistics and hypothesis testing for a single marker

Usage

```
delta.reml.meta(
  delta = NULL,
  sd.delta = NULL,
  epsilon = NULL,
  alpha = 0.05,
  alternative = "two.sided",
  tol = 1e-10,
  verbose = FALSE,
  test = "knha",
  meta.analysis.method = "RE"
)
```

Arguments

`delta`	numeric vector of delta values per study
`sd.delta`	numeric vector of standard error of delta values per study
`epsilon`	numeric non-inferiority margin for testing cross-study validity
`alpha`	numeric significance level of test. Note : using the two-one-sided test (`alternative = "two.sided"`) produces a (1-2`alpha`)*100% confidence interval.
`alternative`	character giving the alternative hypothesis type for testing the summary effect. One of `c("less","two.sided")`, where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".
`tol`	numeric convergence tolerance for finding a root of the score equation
`verbose`	logical flag indicating whether messages should be printed, defaults to `FALSE`
`test`	character giving the type of test to be performed. The default is `knha`, corresponding to variance estimation using the more conservative Hartung-Knapp estimator and performes tests with the t-distribution, whereas setting this argument to `z` estimates the variance with the conventional estimator and uses a normal approximation for testing.
`meta.analysis.method`	character giving the meta-analysis method to be used. The default is `RE`, corresponding to random-effects meta-analysis, whereas setting this argument to `FE` uses fixed-effects meta-analysis.

Value

a list with elements

- `n.studies` : numeric, number of studies considered
- `tau2` : numeric, estimated tau-squared (between-study heterogeneity)
- `mu.delta` : numeric, estimated mean of distribution of delta
- `se.delta` : numeric, standard error of delta summary estimate
- `ci.delta.upper` : numeric, upper confidence interval for mean of delta. Note : if using the non-inferiority test (i.e. `alternative = "less"`), these bounds correspond to a (1-`alpha`)*100% confidence interval, whereas the two-one-sided test (i.e. `alternative = "two.sided"`) corresponds to a (1-2`alpha`)*100% interval.
- `ci.delta.lower` : numeric, lower confidence interval for mean of delta
- `p.lower` : numeric, if `alternative` is `"two.sided"`, gives the p-value corresponding to testing the null hypothesis that `delta` is less than `-epsilon`. Value is `NULL` if `alternative` is `"less"`.
- `p.upper` : numeric, if `alternative` is `"two.sided"`, gives the p-value corresponding to testing the null hypothesis that `delta` is less than `epsilon`. Value is `NULL` if `alternative` is `"less"`.
- `p` : numeric, consensus p-value for hypothesis test for either the two-one-sided test or the non-inferiorty test.
- `Q` : numeric, Cochran's Q-statistic for heterogeneity between studies
- `I2` : numeric, Higgins-Thompson I-squared statistic representing the total percentage of variation attributable to between-study heterogeneity
- `weights.tau` : numeric vector of raw study weights for the summary measure
- `weights.tau.relative` : numeric vector of relative study weights for the summary measure, such that each weight is a percentage adding to 100%
- `weights.tau.sum` : numeric, sum of `weights.tau`

Author(s)

Arthur Hughes

`est.power` *Estimated power to detect a valid surrogate*

Description

Calculates the estimated power to detect a valid surrogate given a total sample size and specified alternative

Usage

```
est.power(n.total, rho = 0.8, u.y.alt, delta.alt, power.want.s = 0.7, alpha = 0.05)
```

Arguments

n.total	total sample size in study
rho	rank correlation between Y and S in group 0, default is 0.8
u.y.alt	specified alternative for u.y
delta.alt	specified alternative for u.s
power.want.s	desired power for u.s, default is 0.7
alpha	significance level, default is 0.05

Value

estimated power

Author(s)

Layla Parast

Examples

```
est.power(n.total = 50, rho = 0.8, u.y.alt=0.9, delta.alt = 0.1)
```

---

example.data *Example data*

Description

Example data use to illustrate the functions

Usage

```
data("example.data")
```

Format

A list with 4 elements representing 25 observations from a treatment group (group 1) and 25 observations from a control group (group 0):

- y1 the primary outcome,Y, in group 1
- y0 the primary outcome, Y, in group 0
- s1 the surrogate marker, S, in group 1
- s0 the surrogate marker, S, in group 0

Examples

```
data(example.data)
```

example.data.highdim *High-dimensional surrogate candidate example dataset*

Description

A simulated high-dimensional dataset for demonstrating the RISE methodology implemented in **SurrogateRank**. The data contains primary response and 1000 surrogate candidates from 25 treated individuals and 25 untreated individuals, where 10% of the surrogate candidates are "valid".

Usage

```
data("example.data.highdim", package = "SurrogateRank")
```

Format

A list containing :

y1 primary response in treated

y0 primary response in untreated

s1 1000 surrogate candidates in treated

s0 1000 surrogate candidates in untreated

hyp for each surrogate, `null` `false` if the surrogate is valid

Source

Simulated for package examples.

Examples

```
data("example.data.highdim", package = "SurrogateRank")
head(example.data.highdim)
```

example.data.highdim.multistudy *High-dimensional, multi-study surrogate candidate example dataset*

Description

A simulated high-dimensional, multi-study dataset for demonstrating the RISE-meta methodology implemented in **SurrogateRank**, generated with the generate.example.data.highdim.multistudy() function. The data contains treatment effect measures on the primary endpoint and on 500 surrogate candidates, where the first 50 of these candidates are "valid" surrogates.

Usage

```
data("example.data.highdim.multistudy", package = "SurrogateRank")
```

Format

A list with the following components:

uy Numeric vector of length `M` containing treatment effects on the primary endpoint across trials.

us Numeric matrix of dimension `M` times `J` containing treatment effects on each of the `J` candidate markers.

hyp Vector of length `J` containing the truth of surrogate validity. `null false` corresponds to valid surrogates, whereas `null true` corresponds to invalid surrogates.

epsilon Value of `epsilon` used to define surrogate validity.

Source

Simulated for package examples.

Examples

```
data("example.data.highdim.multistudy", package = "SurrogateRank")
head(example.data.highdim.multistudy)
```

---

example.data.highdim.multistudy.ipd

High-dimensional multi-study individual participant surrogate candidate example dataset

---

Description

A simulated high-dimensional dataset for demonstrating the RISE-Meta methodology implemented in **SurrogateRank**. The data contains primary response and 100 surrogate candidates from 25 treated individuals and 25 untreated individuals across 5 different studies, where 10% of the surrogate candidates are "valid".

Usage

```
data("example.data.highdim.multistudy.ipd", package = "SurrogateRank")
```

Format

A list containing :

y1 primary response in treated

y0 primary response in untreated

s1 1000 surrogate candidates in treated

s0 1000 surrogate candidates in untreated

study1 study names for treated

study0 study names for untreated

hyp for each surrogate, `null false` if the surrogate is valid

Source

Simulated for package examples.

Examples

```
data("example.data.highdim.multistudy.ipd", package = "SurrogateRank")
head(example.data.highdim.multistudy.ipd)
```

---

generate.example.data.highdim

Generate individual participant data for high-dimensional surrogate candidates and response

---

Description

Generates individual participant data for high-dimensional surrogate candidates using one of two data generating processes, as described in Hughes A et al (2025) doi:10.1002/sim.70241.

Usage

```
generate.example.data.highdim(
  n1,
  n0,
  p,
  prop_valid,
  valid_sigma = 1,
  corr = 0,
  mode = "simple",
  y0_mean = 0,
  y0_sd = 1,
  y1_mean = 3,
  y1_sd = 1,
```

```
  s0_mean = 0,
  s0_sd = 1,
  s1_mean = 0,
  s1_sd = 1,
  seed = 12345
)
```

Arguments

n1	positive numeric giving the sample size in the treated group
n0	positive numeric giving the sample size in the untreated group
p	positive numeric giving the number of markers to generate
prop_valid	numeric between 0 and 1 (inclusive) giving the proportion of surrogate candidates to generate as valid.
valid_sigma	non-negative numeric giving the standard deviation for valid candidates
corr	non-negative numeric giving the correlation between the surrogate candidates
mode	character taking values in c("simple", "complex"). If "simple", generates all variables with (multivariate) normal distributions. Else, uses a more complex exponential distribution.
y0_mean	numeric giving the mean of the primary endpoint in the untreated group
y0_sd	non-negative numeric giving the standard deviation of the primary endpoint in the untreated group
y1_mean	numeric giving the mean of the primary endpoint in the treated group
y1_sd	non-negative numeric giving the standard deviation of the primary endpoint in the treated group
s0_mean	numeric giving the mean of the surrogate candidates in the untreated group
s0_sd	non-negative numeric giving the standard deviation of the surrogate candidates in the untreated group
s1_mean	numeric giving the mean of the surrogate candidates in the treated group
s1_sd	non-negative numeric giving the standard deviation of the surrogate candidates in the treated group
seed	numeric giving a seed for reproducibility

Value

A list with the following components:

y1 vector containing primary endpoint values in treated group

y0 vector containing primary endpoint values in untreated group

s1 n1 times p matrix containing surrogate candidate values in treated group

s0 n0 times p matrix containing surrogate candidate values in untreated group

hyp character vector giving the truth behind the null hypothesis for each surrogate candidate

Examples

```
res <- generate.example.data.highdim(n1 = 25, n0 = 25, p = 500, prop_valid = 1)
dim(res$s1)       # 25 x 500
```

`generate.example.data.highdim.multistudy`

Generate high-dimensional multi-study surrogate marker trial-level effects

---

Description

Generates simulated trial-level treatment effects for multiple surrogate markers across multiple studies, including both valid and invalid surrogates. This function implements a hierarchical random-effects model: true trial-level effects are drawn from marker-specific means with between-trial heterogeneity, and observed trial-level effects include additional within-study sampling error.

Usage

```
generate.example.data.highdim.multistudy(
  epsilon = 0.2,
  M = 5,
  sample_sizes = c(25, 50, 100, 150, 250),
  J = 500,
  prop_valid = 0.1,
  u_tau_min = 0.01,
  u_tau_max = 0.1,
  u_nu_min = 0.01,
  u_nu_max = 0.1,
  prop_invalid_under = 0.5,
  invalid_at_boundary = FALSE,
  invalid_mean_discrete = NULL,
  valid_mean_discrete = NULL,
  seed = 12345
)
```

Arguments

`epsilon`	Numeric in (0,1). Defines the region of validity for the surrogate marker means. Markers with mean discrepancy within `[-epsilon, epsilon]` are valid; others are invalid.
`M`	Integer. Number of trials (studies) to simulate. Must be > 1.
`sample_sizes`	Numeric vector of length `M`. Sample size for each trial. Used to compute within-study variances.
`J`	Integer. Total number of markers to simulate (valid + invalid).
`prop_valid`	Numeric, between 0 and 1. Proportion of markers that are valid.

`u_tau_min` Numeric >= 0. Lower bound of marker-specific between-trial heterogeneity variance (τ_j^2).

`u_tau_max` Numeric >= u_tau_min. Upper bound of marker-specific between-trial heterogeneity variance (τ_j^2).

`u_nu_min` Numeric > 0. Lower bound of marker-specific variance component (ν_j) used to scale within-study sampling error.

`u_nu_max` Numeric >= u_nu_min. Upper bound of marker-specific variance component (ν_j) used to scale within-study sampling error.

`prop_invalid_under`
Numeric, between 0 and 1. Probability that an invalid marker underestimates the treatment effect on Y.

`invalid_at_boundary`
default `FALSE`, meaning invalid surrogates are generated uniformly across the entire invalid region. If `TRUE`, generates invalid surrogates at the boundary values defined by epsilon. This is the worst-case scenario for invalid surrogates and thus is useful for checking the calibration of the method.

`invalid_mean_discrete`
vector of discrete numeric values to sample true means of valid surrogates at. These values must be greater or equal in absolute value than epsilon.

`valid_mean_discrete`
vector of discrete numeric values to sample true means of valid surrogates at. These values must be smaller in absolute value than epsilon.

`seed` numeric giving a seed for reproducibility

Details

The function first draws marker-level parameters: $\mu_{\delta,j}$ from the validity or invalidity region, τ_j^2 from a uniform distribution, and ν_j from a uniform distribution. Then, for each trial, true trial-level effects are drawn as $\delta_{m,j}^{true} \sim N(\mu_{\delta,j}, \tau_j^2)$, and observed effects include independent within-study sampling error $\hat{\delta}_{m,j} \sim N(\delta_{m,j}^{true}, \nu_j/n_m)$.

Value

A list with the following components:

delta M x J matrix of observed trial-level discrepancies ($\hat{\delta}_{m,j}$) including sampling error.

sd.delta M x J matrix of within-study standard deviations ($\sigma_{m,j}$).

n Numeric vector of sample sizes for each trial.

hyp Character vector of length J, "null true" for valid markers and "null false" for invalid markers.

mu.true Numeric vector of true marker-level mean discrepancies ($\mu_{\delta,j}$).

tau2.true Numeric vector of marker-specific between-trial heterogeneity variances (τ_j^2).

Examples

```
res <- generate.example.data.highdim.multistudy(
  epsilon = 0.2,
  M = 5,
  sample_sizes = c(25, 50, 100, 150, 250),
  J = 500,
  prop_valid = 0.1
)
dim(res$delta)       # 5 x 500
head(res$mu.true)
```

`generate.example.data.highdim.multistudy.ipd` *Generate multi-study individual participant data for high-dimensional surrogate candidates and response*

Description

Generates individual participant data for high-dimensional surrogate candidates using one of two data generating processes, as described in Hughes A et al (2025) doi:10.1002/sim.70241.

Usage

```
generate.example.data.highdim.multistudy.ipd(
  M,
  n1,
  n0,
  p,
  prop_valid,
  valid_sigma = 1,
  corr = 0,
  mode = "simple",
  y0_mean = 0,
  y0_sd = 1,
  y1_mean = 3,
  y1_sd = 1,
  s0_mean = 0,
  s0_sd = 1,
  s1_mean = 0,
  s1_sd = 1,
  seed = 12345
)
```

Arguments

`M`	number of studies

n1	positive numeric giving the sample size in the treated groups
n0	positive numeric giving the sample size in the untreated groups
p	positive numeric giving the number of markers to generate
prop_valid	numeric between 0 and 1 (inclusive) giving the proportion of surrogate candidates to generate as valid.
valid_sigma	non-negative numeric giving the standard deviation for valid candidates
corr	non-negative numeric giving the correlation between the surrogate candidates
mode	character taking values in c("simple", "complex"). If "simple", generates all variables with (multivariate) normal distributions. Else, uses a more complex exponential distribution.
y0_mean	numeric giving the mean of the primary endpoint in the untreated group
y0_sd	non-negative numeric giving the standard deviation of the primary endpoint in the untreated group
y1_mean	numeric giving the mean of the primary endpoint in the treated group
y1_sd	non-negative numeric giving the standard deviation of the primary endpoint in the treated group
s0_mean	numeric giving the mean of the surrogate candidates in the untreated group
s0_sd	non-negative numeric giving the standard deviation of the surrogate candidates in the untreated group
s1_mean	numeric giving the mean of the surrogate candidates in the treated group
s1_sd	non-negative numeric giving the standard deviation of the surrogate candidates in the treated group
seed	numeric giving a seed for reproducibility

Value

A list with the following components:

y1 vector containing primary endpoint values in treated group

y0 vector containing primary endpoint values in untreated group

s1 n1 times p matrix containing surrogate candidate values in treated group

s0 n0 times p matrix containing surrogate candidate values in untreated group

study1 study names for treated samples

study0 study names for untreated samples

hyp character vector giving the truth behind the null hypothesis for each surrogate candidate

Examples

```
res <- generate.example.data.highdim.multistudy.ipd(
M = 5,
n1 = 25,
n0 = 25,
p = 500,
prop_valid = 1
)
dim(res$s1)       # (5 studies x 25 individuals = 125) x 500
```

`rise.evaluate` *Function to perform the evaluation stage of RISE : Two-Stage Rank-Based Identification of High-Dimensional Surrogate Markers*

Description

A set of high-dimensional surrogate candidates are evaluated jointly. Strength of surrogacy is assessed through a rank-based measure of the similarity in treatment effects on a candidate surrogate and the primary response.

Usage

```
rise.evaluate(
  yone,
  yzero,
  sone,
  szero,
  alpha = 0.05,
  power.want.s = NULL,
  epsilon = NULL,
  u.y.hyp = NULL,
  p.correction = "BH",
  n.cores = 1,
  alternative = "two.sided",
  paired = FALSE,
  return.all.evaluate = TRUE,
  return.plot.evaluate = TRUE,
  evaluate.weights = TRUE,
  screening.weights = NULL,
  markers = NULL
)
```

Arguments

`yone`	numeric vector of primary response values in the treated group.
`yzero`	numeric vector of primary response values in the untreated group.
`sone`	matrix or dataframe of surrogate candidates in the treated group with dimension `n1 x p` where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly `yone`.
`szero`	matrix or dataframe of surrogate candidates in the untreated group with dimension `n0 x p` where n0 is the number of untreated samples and p the number of candidates. Sample ordering must match exactly `yzero`.
`alpha`	significance level for determining surrogate candidates. Default is `0.05`.
`power.want.s`	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. Either this or `epsilon` argument must be specified.

epsilon — numeric in (0,1) - non-inferiority margin for determining surrogate validity. Either this or power.want.s argument must be specified.

u.y.hyp — hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.

p.correction — character. Method for p-value adjustment (see p.adjust() function). Defaults to the Benjamini-Hochberg method ("BH").

n.cores — numeric giving the number of cores to commit to parallel computation in order to improve computational time through the pbmcapply() function. Defaults to 1.

alternative — character giving the alternative hypothesis type. One of c("less","two.sided"), where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".

paired — logical flag giving if the data is independent or paired. If FALSE (default), samples are assumed independent. If TRUE, samples are assumed to be from a paired design. The pairs are specified by matching the rows of yone and sone to the rows of yzero and szero.

return.all.evaluate — logical flag. If TRUE (default), a dataframe will be returned giving the evaluation of each individual marker passed to the evaluation stage.

return.plot.evaluate — logical flag. If TRUE (default), a ggplot2 object will be returned allowing the user to visualise the association between the composite surrogate on the individual-scale.

evaluate.weights — logical flag. If TRUE (default), the composite surrogate is constructed with weights such that surrogates which are predicted to be stronger receive more weight.

screening.weights — dataframe with columns marker and weight giving the weight in for the evaluation. Typically this is taken directly from the screening stage as the output from the rise.screen() function. Must be given if evaluate.weights is TRUE.

markers — a vector of marker names (column names of szero and sone) to evaluate. If not given, will default to evaluating all markers in the dataframes.

Value

a list with

- individual.metrics if return.all.evaluate=TRUE, a dataframe of evaluation results for each significant marker.
- gamma.s a list with elements gamma.s.one and gamma.s.zero, giving the combined surrogate marker in the treated and untreated groups, respectively.
- gamma.s.evaluate : a dataframe giving the evaluation of gamma.s
- gamma.s.plot : a ggplot2 plot showing gamma.s against the primary response on the rank-scale.

Author(s)

Arthur Hughes

Examples

```
# Load high-dimensional example data
```

`rise.evaluate.meta` *Function to perform the evaluation stage of RISE-meta : Meta-Analysis of High-Dimensional Surrogate Markers*

Description

Function to perform the evaluation stage of RISE-meta : Meta-Analysis of High-Dimensional Surrogate Markers

Usage

```
rise.evaluate.meta(
  yone,
  yzero,
  sone,
  szero,
  studyone,
  studyzero,
  alpha = 0.05,
  power.want.s.study = NULL,
  epsilon.study = NULL,
  epsilon.meta.mode = "user",
  epsilon.meta = NULL,
  u.y.hyp = NULL,
  p.correction = "BH",
  n.cores = 1,
  alternative = "two.sided",
  test = "knha",
  paired.all = FALSE,
  paired.studies = NULL,
  evaluate.weights = TRUE,
  screening.weights = NULL,
  weight.mode = "diff.epsilon",
  markers = NULL,
  return.all.evaluate = FALSE,
  return.forest.plot = TRUE,
  return.fit.plot = TRUE,
  show.pooled.effect = TRUE,
  meta.analysis.method = "RE"
)
```

Arguments

yone	numeric vector of primary response values in the treated participants
yzero	numeric vector of primary response values in the untreated participants
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yzero.
studyone	character vector of length n1 indicating the study corresponding to each treated sample. Ordering much match yone.
studyzero	character vector of length n0 indicating the study corresponding to each untreated sample. Ordering much match yzero.
alpha	significance level for determining valid surrogates. Default is 0.05.
power.want.s.study	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. If return.all.evaluate = TRUE, either this or epsilon.study argument must be specified.
epsilon.study	numeric in (0,1) - non-inferiority margin for determining surrogate validity in the within-study screening phase. If return.all.evaluate = TRUE, either this or power.want.s.study argument must be specified.
epsilon.meta.mode	character string specifying the mode to choose the value of the acceptable margin defined by epsilon. By default, this is set to "user", where the value of epsilon is fixed by the user, defined by the value of the argument epsilon.meta. The alternative is to set this as "mean.power", which corresponds to taking the mean value of epsilon across studies such that the power to detect departures from the null within each study is defined by the power.want.s.study argument.
epsilon.meta	numeric in (0,1) - non-inferiority margin for determining surrogate validity in the meta-analysis stage. Must be specified.
u.y.hyp	hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.
p.correction	character. Method for p-value adjustment (see p.adjust() function). Defaults to the Benjamini-Hochberg method ("BH").
n.cores	numeric giving the number of cores to commit to parallel computation in order to improve computational time through the pbmcapply() function. Defaults to 1.
alternative	character giving the alternative hypothesis type. One of c("less","two.sided"), where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".
test	character giving the type of test to be performed. The default is knha, corresponding to variance estimation using the more conservative Hartung-Knapp estimator and performes tests with the t-distribution, whereas setting this argument to z estimates the variance with the conventional estimator and uses a normal approximation for testing.

`paired.all` logical flag giving if the data is independent or paired. If `FALSE` (default), samples are assumed independent. If `TRUE`, all samples are assumed to be from a paired design. The pairs are specified by matching the rows of `yone` and `sone` to the rows of `yzero` and `szero`, and all studies must be paired. If only some studies are paired and others have independent samples, one may specify the `paired.studies` argument instead.

`paired.studies` character vector specifying the names of the studies in `studyone` or `studyzero` which are paired, in the case where some have paired designs and others do not. By default, this is `NULL`, indicating that study designs are all specified by the `paired.all` argument.

`evaluate.weights` logical flag. If `TRUE` (default), the composite surrogate is constructed with weights such that surrogates which are predicted to be stronger receive more weight.

`screening.weights` dataframe with columns `marker` and `weight` giving the weight in for the evaluation. Typically this is taken directly from the screening stage as the output from the `rise.screen.meta()` function. Must be given if `evaluate.weights` is `TRUE`.

`weight.mode` character giving the type of weighting to return to be used in case `return.all.evaluate = TRUE`. See `rise.screen.meta()` for detail.

`markers` a vector of marker names (column names of szero and sone) to evaluate. If not given, will default to evaluating all markers in the dataframes.

`return.all.evaluate` logical flag. If `TRUE`, a dataframe will be returned giving the meta-analysis evaluation of each individual marker passed to the evaluation stage. Defaults to `FALSE` for computational time.

`return.forest.plot` logical flag. If `TRUE` (default), a forest plot of the effect sizes for the combined signature across studies, with its meta-analysis summary measure, will be included in the output.

`return.fit.plot` logical flag. If `TRUE` (default), a plot of the effects on the primary response versus the effects on the combined surrogate signature for each study will be included in the output.

`show.pooled.effect` logical flag. If `TRUE` (default), the forest plot will show the pooled effect estimate. Otherwise, it will just show the individual trial estimates.

`meta.analysis.method` character giving the meta-analysis method to be used. The default is `RE`, corresponding to random-effects meta-analysis, whereas setting this argument to `FE` uses fixed-effects meta-analysis.

Value

a list with elements

- individual.metrics : if return.all.evaluate=TRUE, a list containing dataframes individual.metrics.study (per-study results for individual markers) and individual.metrics.meta (meta-analysis results for individual markers).
- evaluation.metrics.study : study-level results for the combined marker, gamma.
- evaluation.metrics.meta : meta-analysis results for the combined marker, gamma.
- gamma.s : a list with elements gamma.s.one and gamma.s.zero, giving the values of the combined surrogate marker gamma in the treated and untreated groups, respectively.
- gamma.s.plot : if return.forest.plot and/or return.fit.plot are TRUE, returns evaluation plots as a list

Author(s)

Arthur Hughes

Examples

```
data("example.data.highdim.multistudy.ipd")
yone <- example.data.highdim.multistudy.ipd$y1
yzero <- example.data.highdim.multistudy.ipd$y0
sone <- example.data.highdim.multistudy.ipd$s1
szero <- example.data.highdim.multistudy.ipd$s0
studyone <- example.data.highdim.multistudy.ipd$study1
studyzero <- example.data.highdim.multistudy.ipd$study0
rise.meta.screen.result <- rise.screen.meta(
yone, yzero,
sone, szero,
studyone, studyzero,
epsilon.study = 0.2, epsilon.meta = 0.2
)
markers = rise.meta.screen.result[["significant.markers"]]
screening.weights = rise.meta.screen.result[["screening.weights"]]
rise.meta.evaluate.result <- rise.evaluate.meta(
yone, yzero,
sone, szero,
studyone, studyzero,
epsilon.meta = 0.2,
markers = markers,
screening.weights = screening.weights,
epsilon.study = 0.2
)
```

---

rise.screen *Function to perform the screening stage of RISE : Two-Stage Rank-Based Identification of High-Dimensional Surrogate Markers*

Description

A set of high-dimensional surrogate candidates are screened one-by-one to identify strong candidates. Strength of surrogacy is assessed through a rank-based measure of the similarity in treatment effects on a candidate surrogate and the primary response. P-values corresponding to hypothesis testing on this measure are corrected for the high number of statistical tests performed.

Usage

```
rise.screen(
  yone,
  yzero,
  sone,
  szero,
  alpha = 0.05,
  power.want.s = NULL,
  epsilon = NULL,
  u.y.hyp = NULL,
  p.correction = "BH",
  n.cores = 1,
  alternative = "two.sided",
  paired = FALSE,
  return.all.screen = TRUE,
  return.all.weights = FALSE,
  weight.mode = "inverse.delta",
  normalise.weights = TRUE,
  verbose = T
)
```

Arguments

yone	numeric vector of primary response values in the treated group.
yzero	numeric vector of primary response values in the untreated group.
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of untreated samples and p the number of candidates. Sample ordering must match exactly yzero.
alpha	significance level for determining surrogate candidates. Default is 0.05.
power.want.s	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. Either this or epsilon argument must be specified.
epsilon	numeric in (0,1) - non-inferiority margin for determining surrogate validity. Either this or power.want.s argument must be specified.
u.y.hyp	hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.

`p.correction` character. Method for p-value adjustment (see `p.adjust()` function). Defaults to the Benjamini-Hochberg method (`"BH"`).

`n.cores` numeric giving the number of cores to commit to parallel computation in order to improve computational time through the `pbmcapply()` function. Defaults to `1`.

`alternative` character giving the alternative hypothesis type. One of `c("less","two.sided")`, where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".

`paired` logical flag giving if the data is independent or paired. If `FALSE` (default), samples are assumed independent. If `TRUE`, samples are assumed to be from a paired design. The pairs are specified by matching the rows of `yone` and `sone` to the rows of `yzero` and `szero`.

`return.all.screen` logical flag. If `TRUE` (default), a dataframe will be returned giving the screening results for all candidates. Else, only the significant candidates will be returned.

`return.all.weights` logical flag. If `FALSE` (default), a dataframe will be returned giving weights for significant markers screened. If `TRUE`, weights for all markers will be returned. Note that, if normalised weights are required, these will only be returned for significant markers, and raw weights will be returned in a second column.

`weight.mode` character giving the type of weighting to return. One of `c("inverse.delta","diff.epsilon", or "none")`. The default is `"inverse.delta"`, which means the weights are determined by taking the inverse of the absolute values of delta. If delta is exactly 0, this is uncomputable and the weight defaults to the inverse of the next closest absolute delta value. If delta is very close to 0, these estimates can be unstable and extreme. The `"diff.epsilon"` option seeks to aid this by calculating weights as the proportion of the interval between 0 and epsilon cut by the absolute value of delta, therefore giving delta = 0 a weight of 1 and delta = epsilon a weight of 0. When `"none"`, the weights are set to 1 for every marker.

`normalise.weights` logical flag. If `TRUE` (default), the weights are normalised by the sum of all the weights such that the maximum weight is 1, which can help with interpretability.

`verbose` logical flag. If `TRUE`, prints warning messages.

Value

a list with elements

- `screening.metrics` : dataframe of screening results (for each candidate marker - number of observations n, u.y, u.s, delta, CI, sd, epsilon, p-values).
- `significant.markers`: character vector of markers with `p_adjusted < alpha`
- `screening.weights`: dataframe giving marker names and the inverse absolute value of the associated deltas.

Author(s)

Arthur Hughes

Examples

```
# Load high-dimensional example data
```

`rise.screen.meta` *Function to perform the screening stage of RISE-meta : Meta-Analysis of High-Dimensional Surrogate Markers*

Description

The RISE screening algorithm is applied to each study using a rank-based measure of treatment effect similarity. In the second stage, these effect estimates are combined using a random-effects meta-analysis and the retained markers are those for which there is strong evidence of surrogacy across many studies.

Usage

```
rise.screen.meta(
  yone,
  yzero,
  sone,
  szero,
  studyone,
  studyzero,
  alpha = 0.05,
  power.want.s.study = NULL,
  epsilon.study = NULL,
  epsilon.meta.mode = "user",
  epsilon.meta = NULL,
  u.y.hyp = NULL,
  p.correction = "BH",
  n.cores = 1,
  alternative = "two.sided",
  test = "knha",
  paired.all = FALSE,
  paired.studies = NULL,
  return.all.screen = TRUE,
  return.all.weights = FALSE,
  weight.mode = "diff.epsilon",
  return.screen.plot = TRUE,
  screen.plot.topN = 15,
  screen.plot.point.estimate = FALSE,
  normalise.weights = TRUE,
  return.forest.plot = TRUE,
  return.fit.plot = TRUE,
  show.pooled.effect = TRUE,
  return.study.similarity.plot = TRUE,
```

```
  return.evaluate.results = TRUE,
  meta.analysis.method = "RE"
)
```

Arguments

yone	numeric vector of primary response values in the treated participants
yzero	numeric vector of primary response values in the untreated participants
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yzero.
studyone	character vector of length n1 indicating the study corresponding to each treated sample. Ordering much match yone.
studyzero	character vector of length n0 indicating the study corresponding to each untreated sample. Ordering much match yzero.
alpha	significance level for determining surrogate candidates in both stages. Default is 0.05.
power.want.s.study	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. Either this or epsilon.study argument must be specified.
epsilon.study	numeric in (0,1) - non-inferiority margin for determining surrogate validity in the within-study screening phase. Either this or power.want.s.study argument must be specified.
epsilon.meta.mode	character string specifying the mode to choose the value of the acceptable margin defined by epsilon. By default, this is set to "user", where the value of epsilon is fixed by the user, defined by the value of the argument epsilon.meta. The alternative is to set this as "mean.power", which corresponds to taking the mean value of epsilon across studies such that the power to detect departures from the null within each study is defined by the power.want.s.study argument.
epsilon.meta	numeric in (0,1) - fixed non-inferiority margin for determining surrogate validity in the meta-analysis stage.
u.y.hyp	hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.
p.correction	character. Method for p-value adjustment (see p.adjust() function). Defaults to the Benjamini-Hochberg method ("BH").
n.cores	numeric giving the number of cores to commit to parallel computation in order to improve computational time through the pbmcapply() function. Defaults to 1.
alternative	character giving the alternative hypothesis type. One of c("less","two.sided"), where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".

`test` character giving the type of test to be performed. The default is `knha`, corresponding to variance estimation using the more conservative Hartung-Knapp estimator and performes tests with the t-distribution, whereas setting this argument to `z` estimates the variance with the conventional estimator and uses a normal approximation for testing.

`paired.all` logical flag giving if the data is independent or paired. If `FALSE` (default), samples are assumed independent. If `TRUE`, all samples are assumed to be from a paired design. The pairs are specified by matching the rows of `yone` and `sone` to the rows of `yzero` and `szero`, and all studies must be paired. If only some studies are paired and others have independent samples, one may specify the `paired.studies` argument instead.

`paired.studies` character vector specifying the names of the studies in `studyone` or `studyzero` which are paired, in the case where some have paired designs and others do not. By default, this is `NULL`, indicating that study designs are all specified by the `paired.all` argument.

`return.all.screen` logical flag. If `TRUE` (default), a dataframe will be returned giving the screening results for all candidates. Else, only the significant candidates will be returned.

`return.all.weights` logical flag. If `FALSE` (default), a dataframe will be returned giving weights for significant markers screened. If `TRUE`, weights for all markers will be returned. Note that, if normalised weights are required, these will only be returned for significant markers, and raw weights will be returned in a second column.

`weight.mode` character giving the type of weighting to return. One of `c("diff.epsilon", "inverse.delta", or "none")`. The default is `"diff.epsilon"`, which calculates weights as the proportion of the interval between 0 and epsilon.study cut by the absolute value of delta, therefore giving delta = 0 a weight of 1 and delta = epsilon.study a weight of 0. Another option is `"inverse.delta"` where the weights are determined by taking the inverse of the absolute values of delta. When `"none"`, the weights are set to 1 for every marker.

`return.screen.plot` logical flag. If `TRUE` (default), returns a forest plot of the top predictors, sorted by p-value, from the screening stage. The number of predictors to display is given by the `screen.plot.topN` argument, which has default value 15.

`screen.plot.topN` number of predictors to display in the screening results figure, default value is 15.

`screen.plot.point.estimate` logical flag. If `FALSE` (default), uses the `screen.plot.topN` argument to determine how many markers to display on the screen plot. Otherwise, plots all the markers with a point estimate within the equivalence region.

`normalise.weights` logical flag. If `TRUE` (default), the weights are normalised by the sum of all the weights such that the maximum weight is 1, which can help with interpretability.

`return.forest.plot` logical flag. If `TRUE` (default), a forest plot of the effect sizes for the combined signature across studies, with its meta-analysis summary measure and prediction interval, will be included in the output.

`return.fit.plot`
: logical flag. If `TRUE` (default), a plot of the effects on the primary response versus the effects on the combined surrogate signature for each study will be included in the output.

`show.pooled.effect`
: logical flag. If `TRUE` (default), the forest plot will show the pooled effect estimate. Otherwise, it will just show the individual trial estimates.

`return.study.similarity.plot`
: logical flag. If `TRUE` (default), will return two plots showing the similarity between study-wise marker signatures (i.e., the application of RISE to each study individually, with p-value correction within-study).

`return.evaluate.results`
: logical flag. If `TRUE` (default), returns results for combined marker gamma, evaluated on the same data. Can be useful to set this as `FALSE` to save computational time if this is not of interest.

`meta.analysis.method`
: character giving the meta-analysis method to be used. The default is `RE`, corresponding to random-effects meta-analysis, whereas setting this argument to `FE` uses fixed-effects meta-analysis.

Value

a list with elements

- `screening.metrics.study` : dataframe of per-study results from RISE screening. For each candidate marker - study name, study sample size, estimate of delta, standard error of delta.
- `screening.metrics.meta` : dataframe of meta-analysis screening results. For each candidate marker - number of studies `n.studies`, estimate of mean delta value `mu.delta`, its standard error `se.delta`, confidence interval and prediction interval, estimate of tau-squared `tau2`, Cochran's Q-statistic and Higgins-Thompson I-Squared, unadjusted and adjusted meta-analysis p-values, and standardised weights. Note : if using the non-inferiority test (i.e. `alternative = "less"`), the intervals have width (1-`alpha`)*100%, whereas the two-one-sided test (i.e. `alternative = "two.sided"`) corresponds to a (1-2`alpha`)*100% width.
- `significant.markers`: character vector of markers with meta-analysis p-values < `alpha`
- `screening.weights`: dataframe giving marker names and the standardised meta-analysis weights
- `evaluation.metrics.study` : dataframe of per-study results for the combined marker gamma, evaluated on the same data
- `evaluation.metrics.meta` : dataframe of meta-analysis results for the combined marker gamma, evaluated on the same data
- `gamma.s.plot`: if `return.forest.plot`, `return.fit.plot`, and/or `return.study.similarity.plot` are `TRUE`, returns fitted evaluation plots on training data as a list.

Author(s)

Arthur Hughes

Examples

```
data("example.data.highdim.multistudy.ipd")
yone <- example.data.highdim.multistudy.ipd$y1
yzero <- example.data.highdim.multistudy.ipd$y0
sone <- example.data.highdim.multistudy.ipd$s1
szero <- example.data.highdim.multistudy.ipd$s0
studyone <- example.data.highdim.multistudy.ipd$study1
studyzero <- example.data.highdim.multistudy.ipd$study0
rise.meta.screen.result <- rise.screen.meta(
yone, yzero,
sone, szero,
studyone, studyzero,
epsilon.study = 0.2, epsilon.meta = 0.2
)
```

---

test.surrogate *Tests whether the surrogate is valid*

Description

Calculates the rank-based test statistic for Y and the rank-based test statistic for S and the difference, delta, along with corresponding standard error estimates, then tests whether the surrogate is valid

Usage

```
test.surrogate(full.data = NULL, yone = NULL, yzero = NULL, sone = NULL,
szero = NULL, epsilon = NULL, power.want.s = 0.7, u.y.hyp = NULL, alpha = 0.05)
```

Arguments

full.data	either full.data or yone, yzero, sone, szero must be supplied; if full data is supplied it must be in the following format: one observation per row, Y is in the first column, S is in the second column, treatment group (0 or 1) is in the third column.
yone	primary outcome, Y, in group 1
yzero	primary outcome, Y, in group 0
sone	surrogate marker, S, in group 1
szero	surrogate marker, S, in group 0
epsilon	threshold to use for delta, default calculates epsilon as a function of desired power for S
power.want.s	desired power for S, default is 0.7
u.y.hyp	hypothesized value of u.y used in the calculation of epsilon, default uses estimated valued of u.y
alpha	significance level, default is 0.05

Value

u.y	rank-based test statistic for Y
u.s	rank-based test statistic for S
delta	difference, u.y-u.s
sd.u.y	standard error estimate of u.y
sd.u.s	standard error estimate of u.s
sd.delta	standard error estimate of delta
ci.delta	1-sided confidence interval for delta
epsilon.used	the epsilon value used for the test
is.surrogate	logical, TRUE if test indicates S is a good surrogate, FALSE otherwise

Author(s)

Layla Parast

Examples

```
data(example.data)
test.surrogate(yone = example.data$y1, yzero = example.data$y0, sone = example.data$s1,
szero = example.data$s0)
```

---

test.surrogate.extension

Function to test for trial-level surrogacy of a single marker extended to the paired, two sided test setting

---

Description

This function tests for surrogacy of a univariate marker with respect to a continuous primary response. This extends the `test.surrogate()` function from the `SurrogateRank` package to the case where samples may be paired instead of independent, and where a two sided test is desired.

Usage

```
test.surrogate.extension(
  yone,
  yzero,
  sone,
  szero,
  alpha = 0.05,
  power.want.s = NULL,
  epsilon = NULL,
  u.y.hyp = NULL,
  alternative = "two.sided",
  paired = FALSE
)
```

Arguments

yone	numeric vector of primary response values in the treated group.
yzero	numeric vector of primary response values in the untreated group.
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of untreated samples and p the number of candidates. Sample ordering must match exactly yzero.
alpha	significance level for determining surrogate candidates. Default is 0.05.
power.want.s	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. Either this or epsilon argument must be specified.
epsilon	numeric in (0,1) - non-inferiority margin for determining surrogate validity. Either this or power.want.s argument must be specified.
u.y.hyp	hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.
alternative	character giving the alternative hypothesis type. One of c("less","two.sided"), where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".
paired	logical flag giving if the data is independent or paired. If FALSE (default), samples are assumed independent. If TRUE, samples are assumed to be from a paired design. The pairs are specified by matching the rows of yone and sone to the rows of yzero and szero.

Value

A list containing:

- u.y: Estimated rank-based treatment effect on the outcome.
- u.s: Estimated rank-based treatment effect on the surrogate.
- delta.estimate: Estimated difference in treatment effects: u.y - u.s.
- sd.u.y: Standard deviation of u.y.
- sd.u.s: Standard deviation of u.s.
- sd.delta: Standard deviation of delta.estimate.
- ci.delta: One-sided confidence interval upper bound for delta.estimate.
- p.delta: p-value for validity of trial-level surrogacy.
- epsilon.used: Non-inferiority threshold used in the test.
- is.surrogate: TRUE if the surrogate passes the test, else FALSE.

Author(s)

Arthur Hughes, Layla Parast

Examples

```
# Load data
data("example.data")
yone <- example.data$y1
yzero <- example.data$y0
sone <- example.data$s1
szero <- example.data$s0
test.surrogate.extension.result <- test.surrogate.extension(
  yone, yzero, sone, szero,
  power.want.s = 0.8, paired = TRUE, alternative = "two.sided"
)
```

`test.surrogate.rise` *Function to perform RISE : Two-Stage Rank-Based Identification of High-Dimensional Surrogate Markers*

Description

RISE (Rank-Based Identification of High-Dimensional Surrogate Markers) is a two-stage method to identify and evaluate high-dimensional surrogate candidates of a continuous response.

In the first stage (called screening), the high-dimensional candidates are screened one-by-one to identify strong candidates. Strength of surrogacy is assessed through a rank-based measure of the similarity in treatment effects on a candidate surrogate and the primary response. P-values corresponding to hypothesis testing on this measure are corrected for the high number of statistical tests performed.

In the second stage (called evaluation), candidates with an adjusted p-value below a given significance level are evaluated by combining them into a single synthetic marker. The surrogacy of this marker is then assessed with the univariate test as described before.

To avoid overfitting, the two stages are performed on separate data.

Usage

```
test.surrogate.rise(
  yone,
  yzero,
  sone,
  szero,
  alpha = 0.05,
  power.want.s = NULL,
  epsilon = NULL,
  u.y.hyp = NULL,
  p.correction = "BH",
  n.cores = 1,
  alternative = "two.sided",
  paired = FALSE,
  screen.proportion = 0.66,
```

```
  return.all.screen = TRUE,
  return.all.evaluate = TRUE,
  return.plot.evaluate = TRUE,
  evaluate.weights = TRUE,
  return.all.weights = FALSE,
  weight.mode = "inverse.delta",
  normalise.weights = TRUE
)
```

Arguments

yone	numeric vector of primary response values in the treated group.
yzero	numeric vector of primary response values in the untreated group.
sone	matrix or dataframe of surrogate candidates in the treated group with dimension n1 x p where n1 is the number of treated samples and p the number of candidates. Sample ordering must match exactly yone.
szero	matrix or dataframe of surrogate candidates in the untreated group with dimension n0 x p where n0 is the number of untreated samples and p the number of candidates. Sample ordering must match exactly yzero.
alpha	significance level for determining surrogate candidates. Default is 0.05.
power.want.s	numeric in (0,1) - power desired for a test of treatment effect based on the surrogate candidate. Either this or epsilon argument must be specified.
epsilon	numeric in (0,1) - non-inferiority margin for determining surrogate validity. Either this or power.want.s argument must be specified.
u.y.hyp	hypothesised value of the treatment effect on the primary response on the probability scale. If not given, it will be estimated based on the observations.
p.correction	character. Method for p-value adjustment (see p.adjust() function). Defaults to the Benjamini-Hochberg method ("BH").
n.cores	numeric giving the number of cores to commit to parallel computation in order to improve computational time through the pbmcapply() function. Defaults to 1.
alternative	character giving the alternative hypothesis type. One of c("less","two.sided"), where "less" corresponds to a non-inferiority test and "two.sided" corresponds to a two one-sided test procedure. Default is "two.sided".
paired	logical flag giving if the data is independent or paired. If FALSE (default), samples are assumed independent. If TRUE, samples are assumed to be from a paired design. The pairs are specified by matching the rows of yone and sone to the rows of yzero and szero.
screen.proportion	numeric in (0,1) - proportion of data to be used for the screening stage. The default is 2/3. If 1 is given, screening and evaluation will be performed on the same data.
return.all.screen	logical flag. If TRUE (default), a dataframe will be returned giving the screening results for all candidates. Else, only the significant candidates will be returned.

return.all.evaluate
: logical flag. If TRUE (default), a dataframe will be returned giving the evaluation of each individual marker passed to the evaluation stage.

return.plot.evaluate
: logical flag. If TRUE (default), a ggplot2 object will be returned allowing the user to visualise the association between the composite surrogate on the individual-scale.

evaluate.weights
: logical flag. If TRUE (default), the composite surrogate is constructed with weights such that surrogates which are predicted to be stronger receive more weight.

return.all.weights
: logical flag. If FALSE (default), a dataframe will be returned giving weights for significant markers screened. If TRUE, weights for all markers will be returned. Note that, if normalised weights are required, these will only be returned for significant markers, and raw weights will be returned in a second column.

weight.mode
: character giving the type of weighting to return. One of c("inverse.delta","diff.epsilon", or "none"). The default is "inverse.delta", which means the weights are determined by taking the inverse of the absolute values of delta. If delta is exactly 0, this is uncomputable and the weight defaults to the inverse of the next closest absolute delta value. If delta is very close to 0, these estimates can be unstable and extreme. The "diff.epsilon" option seeks to aid this by calculating weights as the proportion of the interval between 0 and epsilon cut by the absolute value of delta, therefore giving delta = 0 a weight of 1 and delta = epsilon a weight of 0. When "none", the weights are set to 1 for every marker.

normalise.weights
: logical flag. If TRUE (default), the weights are normalised by the sum of all the weights such that the maximum weight is 1, which can help with interpretability.

Value

a list with

- screening.results: a list with
  - screening.metrics : dataframe of screening results (for each candidate marker - number of observations n, u.y, u.s, delta, CI, sd, epsilon, p-values)
  - significant_markers: character vector of markers with p_adjusted < alpha.
- evaluate.results: a list with
  - individual.metrics if return.all.evaluate=TRUE, a dataframe of evaluation results for each significant marker.
  - gamma.s a list with elements gamma.s.one and gamma.s.zero, giving the combined surrogate marker in the treated and untreated groups, respectively.
  - gamma.s.evaluate : a dataframe giving the evaluation of gamma.s
  - gamma.s.plot : a ggplot2 plot showing gamma.s against the primary response on the rank-scale.

Author(s)

Arthur Hughes

Examples

```
# Load high-dimensional example data
```

Index