# Package 'SurrogateRank'

January 8, 2026

**Type** Package

**Title** Rank-Based Test to Evaluate a Surrogate Marker

**Version** 2.2

**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> and Hughes A et al (2025) <doi:10.1002/sim.70241>. 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

**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** 2026-01-08 04:00:14 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` *Calculates the rank-based test statistic for Y and S and the difference, delta, accomodating paired data and allowing for a two-sided test*

## 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 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, 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`. |
| `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
)
```

---

## 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)
```

### 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 |

### 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 *Example data for the high-dimensional functions*

### Description

A simulated high-dimensional dataset for demonstrating the RISE methodology implemented in this package. 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")
```

### 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 (note that this is from simulated data and is used to demonstrate the method; this would be unknown in practice)

## Source

Simulated for package examples.

## Examples

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

---

`rise.evaluate` *Performs 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 = "less",
  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 "less". |
| 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
data("example.data.highdim")
yone <- example.data.highdim$y1
yzero <- example.data.highdim$y0
sone <- example.data.highdim$s1
szero <- example.data.highdim$s0

rise.evaluate.result <- rise.evaluate(yone, yzero, sone, szero, power.want.s = 0.8)
```

---

# rise.screen *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 = "less",
  paired = FALSE,
```

```
  return.all.screen = TRUE,
  return.all.weights = FALSE,
  weight.mode = "inverse.delta",
  normalise.weights = 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 "less". |
| 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.

## 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
data("example.data.highdim")
yone <- example.data.highdim$y1
yzero <- example.data.highdim$y0
sone <- example.data.highdim$s1
szero <- example.data.highdim$s0

rise.screen.result <- rise.screen(yone, yzero, sone, szero, power.want.s = 0.8)
```

---

# 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)
```

## 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 |

## 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` *Tests whether the surrogate is valid, extended to the paired, two sided test setting*

## 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. This extends the `test.surrogate()` function 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 = "less",
  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 "less".

`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` *Performs 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 = "less",
  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 = 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 "less".

`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
data("example.data.highdim")
yone <- example.data.highdim$y1
yzero <- example.data.highdim$y0
sone <- example.data.highdim$s1
szero <- example.data.highdim$s0

rise.result <- test.surrogate.rise(yone, yzero, sone, szero, power.want.s = 0.8)
```

# Index