Package: informativeSCI (via r-universe)

October 25, 2026

Type Package

Title Informative Simultaneous Confidence Intervals

Version 1.0.4

Maintainer Liane Kluge <liane@uni-bremen.de>

Description Calculation of informative simultaneous confidence intervals for graphical described multiple test procedures and given information weights. Bretz et al. (2009) <doi:10.1002/sim.3495> and Brannath et al. (2024) <doi:10.48550/arXiv.2402.13719>. Furthermore, exploration of the behavior of the informative bounds in dependence of the information weights. Comparisons with compatible bounds are possible. Strassburger and Bretz (2008) <doi:10.1002/sim.3338>.

License MIT + file LICENSE

URL https://github.com/LianeKluge/informativeSCI

BugReports https://github.com/LianeKluge/informativeSCI/issues

Depends gMCP (>= 0.8-17)

Imports mvtnorm (>= 1.2-4)

Encoding UTF-8

RoxygenNote 7.3.2

SystemRequirements Java (>= 5.0)

NeedsCompilation no

Author Liane Kluge [aut, cre], Werner Brannath [ctb], Martin Scharpenberg [ctb]

Config/pak/sysreqs default-jdk

Repository https://cran.r-universe.dev

Date/Publication 2025-05-26 15:36:03 UTC

RemoteUrl https://github.com/cran/informativeSCI

RemoteRef HEAD

RemoteSha 660874cf93cff02646fd113598fe78800330be53

Contents

informativeSCI-package 2
explore_q 3
inExactSCI 7
informSCI 10
notInExactSCI 13
sigmaManyToOne 16

Index **17**

`informativeSCI-package`

informativeSCI: Informative Simultaneous Confidence Intervals

---

Description

Calculation of informative simultaneous confidence intervals for graphical described multiple test procedures and given information weights. Bretz et al. (2009) doi:10.1002/sim.3495 and Brannath et al. (2024) doi:10.48550/arXiv.2402.13719. Furthermore, exploration of the behavior of the informative bounds in dependence of the information weights. Comparisons with compatible bounds are possible. Strassburger and Bretz (2008) doi:10.1002/sim.3338.

Details

The main function of the package is the informSCI-function for calculating informative lower simultaneous confidence bounds for a given graphical test procedure and given information weights. The explore_q-function can help to find the right information weights for the informSCI-algorithm. The inExactSCI- and notInExactSCI-functions can help to determine how accurate a (numerical) approximation of the true informative lower SCI-bounds is.

Author(s)

Maintainer: Liane Kluge <liane@uni-bremen.de>

Other contributors:

- Werner Brannath [contributor]
- Martin Scharpenberg [contributor]

References

F. Bretz, W. Maurer, W. Brannath, M. Posch: A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine 28.4 (2009), pp. 586-604.

K. Strassburger, F. Bretz: Compatible simultaneous lower confidence bounds for the Holm procedure and other Bonferroni based closed tests. Statistics in Medicine 27.4 (2008), pp. 4914-4927. S. Schmidt, W. Brannath: Informative Simultaneous Confidence Intervals in Hierarchical Testing. Methods of Information in Medicine 53.4 (2014), pp. 278–283.

See Also

Useful links:

- https://github.com/LianeKluge/informativeSCI
- Report bugs at https://github.com/LianeKluge/informativeSCI/issues

---

`explore_q` *Exploration of the Information Weights*

Description

The function calculates various statistical quantities giving some information about the behavior of informative lower SCI-bounds (informSCI) and its induced test for a given graphical test procedure with m hypotheses. The simulation is done for different information weights of the hypotheses. These statistical quantities are intended to be used for determining information weights that represent the best possible trade-off between the number of rejections and the expected size of the informative lower informative SCI-bounds. The statistical quantities can also be calculated for the graphical test and the related compatible lower SCI-bounds, which allows a comparison between the two strategies.

Usage

```
explore_q(
  gMCP = NULL,
  g = NULL,
  weights = NULL,
  trueParam,
  sigma = NULL,
  qFixed = matrix(0, 0, 2),
  mu_0 = 0,
  alpha = 0.05,
  addHyp = matrix(0, 0, 3),
  allRej = NULL,
  atLeastOneRej = NULL,
  qGrid = NULL,
  qInterval = c(0, 1),
  qStepSize = 1/10,
  numSim = 1000,
  sampleSizes = NULL,
  sampleSizeControl = NULL,
  varObs = NULL,
  exploreGraph = TRUE,
  eps = 1/10^5,
  timesSmallerEps = 3,
  maxIterSCI = 1000,
  maxIterBisec = 1000,
  tolBisec = 1/10^3
)
```

Arguments

gMCP	An object of class graphMCP indicating the underlying graphical test.
g	Numeric square matrix of transition weights for the graphical test with m rows and m columns. The i-th row of the entered matrix defines the arrows starting from the i-th hypothesis. Each entry has to be between 0 and 1 and each row must sum to a number less than or equal to 1. The diagonal elements must be zero. Entering g and weights can be used as the input as an alternative to specifying gMCP.
weights	Numeric vector of weights of dimension m. It defines the initial proportion of significance level which is assigned to each null hypothesis. Entering g and weights can be used as the input as an alternative to specifying gMCP.
trueParam	A numeric vector of dimension m defining the assumed true parameters $\vartheta_i, 1 \leq i \leq m$.
sigma	A covariance matrix of dimension $m \times m$. sigma indicates the covariance matrix of the point estimators for the parameter of interest. Can be missing in the case of a many-to-one comparison. Then, sampleSizes, sampleSizeControl and varObs must be specified.
qFixed	A numeric matrix with l rows and 2 columns, where l is an integer between 0 and m. The matrix describes the fixed information weights of the simulation. The first column indicates the indices of the hypothesis for which the information weight should be fixed during the simulation (i.e. the entries of the first column must be natural numbers between 1 and m). The second column contains the fixed values of their respective fixed information weights (i.e. the entries of the second column must be between 0 and 1 (inclusive)). It is permissible for all information weights to be fixed (i.e. qFixed has m rows) or none to be fixed (i.e. qFixed has 0 rows).
mu_0	A numeric vector of dimension 1 or m defining the bounds of the null hypotheses of the underlying graphical test. If mu_0 has dimension 1, the same value is used for each null hypothesis.
alpha	A numeric defining the overall significance level for the graphical test (i.e. SCIs will have coverage probability of at least $1-$alpha. The parameter must be strictly between 0 and 1.
addHyp	A numeric matrix with k rows and 3 columns (k can be 0) The matrix indicates for which (further) shifted hypotheses the rejection probability is to be calculated. Every row describes one hypothesis. The first entry is a natural number greater than m identifying the hypothesis. The second entry of each row is the index of the corresponding parameter of interest. The third entry is the right border of the hypothesis.
allRej	A list of vectors. Each vector in the list contains the indices of subfamilies of the family of all hypotheses, including the addHyp. The indices of the null hypotheses of the underlying graph range from 1 to m. The indices for addHyp are given by the first column of addHyp. For each such family, the probability of rejecting all hypotheses at the same time is calculated.
atLeastOneRej	A list of vectors. Each vector in the list contains the indices of subfamilies of the family of all hypotheses, including the addHyp. The indices of the null

hypotheses of the underlying graph range from 1 to m. The indices for addHyp are given by the first column of addHyp. For each such family, the probability of rejecting at least one hypothesis is calculated.

qGrid — A numeric vector indicating the values of the non-fixed information weights for the simulation. The entries must be between 0 and 1 (inclusive).

qInterval — A numeric vector of dimension 2 specifying the minimum and maximum values allowed for the varying information weights. qInterval and qStepsize can be used as the input as an alternative to specifying qGrid. If all are entered, qGrid is used and qInterval and qStepSize are ignored.

qStepSize — A positive numeric defining the step size for the varying information weights. qInterval and qStepsize can be used as the input as an alternative to specifying qGrid.

numSim — A natural number indicating how many simulations are to be performed.

sampleSizes — A numeric vector indicating the sample size of each non-control group, in the many-to-one case. Not required if sigma is entered.

sampleSizeControl — A numeric indicating the sample size of the control group, in the many-to-one case. Not required if sigma is entered.

varObs — A positive numeric indicating the variance of the individual observations, in the many-to-one case. Not required if sigma is entered.

exploreGraph — A boolean indicating whether the simulation should be also done for the underlying graphical test and the corresponding compatible lower SCI-bounds.

eps — A numeric for the informSCI-algorithm indicating the desired strict upper bound on the Chebyshev distance between two successive calculated approximations (the Chebyshev distance is induced by the maximum norm).

timesSmallerEps — A positive integer for the informSCI-algorithm indicating how many times the Chebyshev distance of two successive calculated approximations should be less than eps in succession. Here we use the convention $-\infty - (-\infty) := 0$.

maxIterSCI — Maximum number of iterations for determining the lower informative SCI-bounds.

maxIterBisec — Maximum number of iterations of the bisection method which is used during the informSCI-algorithm for finding roots.

tolBisec — A non-negative numeric indicating the error tolerance of the bisection method which is used for finding roots in the informSCI-algorithm.

Details

It is assumed that there are m parameters of interest $\vartheta_1, \ldots, \vartheta_m$. For each parameter there is a null hypothesis defined as $H_i^{\mu_{0i}} : \vartheta_i \leq \mu_{0i}$. The bounds μ_0 correspond to mu_0. The underlying graphical test (specified by gMCP or g and weights) is based on these hypotheses.

The function simulates estimations of point estimators for the parameter of interest $\vartheta_1, \ldots, \vartheta_m$. The estimators follow a multivariate normal distribution with mean trueParam and covariance matrix sigma. The function repeatedly calls the informSCI-function.

The algorithm only optimizes for a single parameter, which is used for all non-fixed information weights. The parameter is chosen from a grid specified by `qInterval` and `qStepsize`. The constructed grid contains all values which are between `qInterval[1]` and `qInterval[2]` and can be written as `qInterval[1]`$+k\cdot$`qStepsize` where k is a natural number. Alternatively, the parameter is chosen directly from `qGrid`.

Value

The function returns a list containing several statistical quantities to use for the informative lower SCI-bounds to find the best possible trade-off between the number of rejections and the expected size of the informative lower SCI-bounds. In the case that `exploreGraph=TRUE`, the returned list also contains the same quantities for the (original) graphical test and related compatible bounds. This allows a comparison.

- `rejecHyp`: A matrix containing for several hypotheses the empirical rejection probability by the informative confidence bounds. The first m rows correspond to the hypotheses of the graph. The other rows correspond to the hypotheses specified by `addHyp`. Each row indicates the rejection probability for different values of the information weights.
- `meanISCI`: A matrix containing in its columns the empirical mean of the lower informative confidence bounds for different information weights. Only the lower bounds which are greater than `-Inf` are used for the empirical mean.
- `impISCI`: A matrix containing in its columns the empirical average distance between the lower informative confidence bounds and `mu_0` for different information weights. Only the lower bounds which are greater than `-Inf` are used for the empirical average distance.
- `biasISCI`: A matrix containing in its columns the empirical average distance between the lower informative confidence bounds and the true parameters `trueParam` for different information weights. Only the lower bounds which are greater than `-Inf` are used for the empirical average distance.
- `numISCIfinite`: A matrix containing in its columns how many times the lower informative confidence bounds were each greater than `-Inf` for different information weights.
- `rejecAllHyp`: A matrix containing in its columns for each family from `allRej` the empirical probability of rejecting all of the hypotheses from the family with the induced test at the same time for different information weights.
- `rejecAtLeastHyp`: A matrix containing in its columns for each family from `atLeastOneRej` the empirical probability of rejecting at least one of the hypotheses from the family with the induced test for different information weights.

If `exploreGraph=TRUE`:

- `rejecHypGraph`: A vector containing for each of the null hypotheses of the graph and of the additional hypotheses (specified by `addHyp`) its empirical rejection probability by the original graph.
- `meanCSCI`: A vector containing, for each parameter $\vartheta_i, 1 \leq i \leq m$ the empirical mean of the lower compatible confidence bounds. Only the lower bounds which are greater than `-Inf` are used for the empirical mean.
- `impCSCI`: A vector containing, for each parameter, the empirical average distance between the lower compatible confidence bounds and `mu_0`. Only the lower bounds which are greater than `-Inf` are used.

- biasCSCI: A vector containing, for each parameter, the empirical average distance between the lower compatible confidence bounds and the true parameters trueParam. Only the lower bounds which are greater than -Inf are used.
- numCSCIfinite: A vector containing, for each parameter, how many times the compatible lower confidence bounds were each greater than -Inf.
- rejecAllHypCSCI: A vector containing, for each family from allRej, the empirical probability of rejecting all of the hypotheses from the family with the (original) graphical test.
- rejecAtLeastHypCSCI: A vector containing, for each family from atLeastOneRej, the empirical probability of rejecting at least one of the hypotheses from the family with the (original) graphical test.

References

S. Schmidt, W. Brannath: Informative simultaneous confidence intervals for the fallback procedure. Biometrical Journal 57.4 (2015), pp. 712–719.

See Also

informSCI gMCP simConfint

Examples

```
explore_q(gMCP=BonferroniHolm(3), trueParam=c(1.5,1,0.2),
sigma=diag(3)*0.2, qFixed=matrix(c(2,3,0.3,0.3),2,2), mu_0=c(-0.5,0,0),
addHyp=matrix(c(4,1,0),1,3),allRej =list(c(1,2), c(4,2)),
atLeastOneRej=list(c(2,3)),numSim=100)
explore_q(g=matrix(c(0,0,1,0),2,2), weights=c(1,0), trueParam=c(0.5,2),
mu_0=c(-1,0), alpha=0.025, qGrid=c(1/10*c(1:10),c(0.97,0.98,0.99)),
numSim=100, sampleSizes=c(89,95), sampleSizeControl=77, varObs=10)
```

---

inExactSCI *Checking Precision of Approximations*

Description

The functions checks whether information about the precision of an approximation for the informative lower SCI-bounds can be collected.

Usage

```
inExactSCI(
  L,
  randomShifts = 0,
  shifts = NULL,
  tolTrueSCI,
  gMCP = NULL,
```

```
  g = NULL,
  weights = NULL,
  q,
  estimates = NULL,
  Z = NULL,
  pValues = NULL,
  SE = NULL,
  I = NULL,
  mu_0,
  alpha,
  checkInput = TRUE
)
```

Arguments

L	An m-dimensional non-negative vector whose entries are the lower bounds of an approximation of the informative SCI.
randomShifts	A positive integer indicating how many random directions of length tolTrueSCI should be generated.
shifts	A matrix with m columns and any number of rows. Each entry must be non-negative. Each row is a direction in the m-dimensional real space. Each row must have at least one positive entry. randomShifts should be a positive integer or shifts should contain at least one row.
tolTrueSCI	The randomly generated shift-vectors and the row vectors in shifts are rescaled to have length tolTrueSCI.
gMCP	An object of class graphMCP indicating the underlying graphical test.
g	Numeric square matrix of transition weights for the graphical test with m rows and m columns. The i-th row of the entered matrix defines the arrows starting from the i-th hypothesis. Each entry has to be between 0 and 1 and each row must sum to a number less than or equal to 1. The diagonal elements must be zero. Entering g and weights can be used as the input as an alternative to specifying gMCP.
weights	Numeric vector of weights of dimension m. It defines the initial proportion of significance level which is assigned to each null hypothesis. Entering g and weights can be used as the input as an alternative to specifying gMCP.
q	A numeric vector of dimension 1 or m defining the information weights for each hypothesis. The entries have to be between 0 and 1 (inclusive). If q has dimension 1, the same information weight is used for each hypothesis.
estimates	An m-dimensional numeric vector of unbiased point estimates for the parameters of interest. Each estimator is assumed to be (asymptotically) normal.
Z	An m-dimensional numeric vector of z-scores for testing the null hypotheses. Z can be entered as an alternative to estimates.
pValues	An m-dimensional numeric vector of p-values from (asymptotic) z-tests for testing the null hypotheses. pValues can be entered as an alternative to estimates and Z.

SE	A positive numeric vector of dimension 1 or m indicating the standard errors of the point estimators. If SE has dimension 1, the same value is used for each estimator.
I	A positive numeric vector indicating the information of the estimators. It can be entered as an alternative to the vector SE. The relationship $SE = 1/I^{1/2}$ is assumed.
mu_0	A numeric vector of dimension 1 or m defining the bounds of the null hypotheses of the underlying graphical test. If mu_0 has dimension 1, the same value is used for each null hypothesis.
alpha	A numeric defining the overall significance level for the graphical test (i.e. SCIs will have coverage probability of at least $1-$alpha. The parameter must be strictly between 0 and 1.
checkInput	A boolean specifying whether the entered values should be checked.

Details

The function checks if it can be determined whether L can be shifted by a randomly generated rescaled direction or by a rescaled direction in the shift matrix such that it lies in the true SCI. If this is possible, the approximation is precise. (The random directions are generated in such a way that all entries are positive.)

Let m be the dimension of L. m also describes the number of hypotheses of interest.

Value

Returns TRUE if we can determine that the approximation is indeed precise. Returns FALSE if we cannot determine that the approximation is precise. (The approximation may still be precise.)

See Also

informSCI explore_q

Examples

```
g <- matrix(c(0,0,1,0),2,2)
weights <- c(1,0)
q <- c(0.0068,1)
mu_0 <- c(-1,0)
pValues <- c(0.0002,0.01)
SE <- c(0.31,1.11)
alpha <- 0.025
L <- informSCI(g=g, weights=weights, q=q, mu_0=mu_0, pValues=pValues, SE=SE,
alpha=alpha, eps=1/10^5, tolBisec=1/10^5)$L
# When the randomShifts- or shift-parameter in the informSCI-function is
# specified, the inExactSCI-function is called by the informSCI-function.
# It is also possible to analyse the accuracy of a calculated L (or an
# approximation of the lower informative SCI-bounds) by directly using
# the inExactSCI-function:
inExactSCI(L=L, randomShifts=100, tolTrueSCI=1/10^5, g=g, weights=weights,
q=q, pValues=pValues, SE=SE, mu_0=mu_0, alpha=alpha)
```

informSCI *Calculation of Lower Informative SCI-bounds*

Description

The function calculates informative lower SCI-bounds for a given graph of m hypotheses and given information weights. m is a natural number.

Usage

```
informSCI(
  gMCP = NULL,
  g = NULL,
  weights = NULL,
  q,
  mu_0 = 0,
  estimates = NULL,
  Z = NULL,
  pValues = NULL,
  SE = NULL,
  I = NULL,
  alpha = 0.05,
  eps = 1/10^5,
  timesSmallerEps = 3,
  randomShifts = 0,
  shifts = NULL,
 tolTrueSCI = sqrt(ifelse(!is.null(gMCP), length(getWeights(gMCP)), length(weights))) *
    eps,
  maxIter = 1000,
  maxIterBisec = 1000,
  tolBisec = 1/10^5,
  calculateCSCI = TRUE,
  checkInput = TRUE
)
```

Arguments

gMCP	An object of class graphMCP indicating the underlying graphical test.
g	Numeric square matrix of transition weights for the graphical test with m rows and m columns. The i-th row of the entered matrix defines the arrows starting from the i-th hypothesis. Each entry has to be between 0 and 1 and each row must sum to a number less than or equal to 1. The diagonal elements must be zero. Entering g and weights can be used as the input as an alternative to specifying gMCP.
weights	Numeric vector of weights of dimension m. It defines the initial proportion of significance level which is assigned to each null hypothesis. Entering g and weights can be used as the input as an alternative to specifying gMCP.

q — A numeric vector of dimension 1 or m defining the information weights for each hypothesis. The entries have to be between 0 and 1 (inclusive). If q has dimension 1, the same information weight is used for each hypothesis.

mu_0 — A numeric vector of dimension 1 or m defining the bounds of the null hypotheses of the underlying graphical test. If mu_0 has dimension 1, the same value is used for each null hypothesis.

estimates — An m-dimensional numeric vector of unbiased point estimates for the parameters of interest. Each estimator is assumed to be (asymptotically) normal.

Z — An m-dimensional numeric vector of z-scores for testing the null hypotheses. Z can be entered as an alternative to estimates.

pValues — An m-dimensional numeric vector of p-values from (asymptotic) z-tests for testing the null hypotheses. pValues can be entered as an alternative to estimates and Z.

SE — A positive numeric vector of dimension 1 or m indicating the standard errors of the point estimators. If SE has dimension 1, the same value is used for each estimator.

I — A positive numeric vector indicating the information of the estimators. It can be entered as an alternative to the vector SE. The relationship SE$= 1/\mathtt{I}^{1/2}$ is assumed.

alpha — A numeric defining the overall significance level for the graphical test (i.e. SCIs will have coverage probability of at least $1-$alpha. The parameter must be strictly between 0 and 1.

eps — A numeric indicating the desired strict upper bound on the Chebyshev distance between two successive calculated approximations (the Chebyshev distance is induced by the maximum norm).

timesSmallerEps — A positive integer indicating how many times the Chebyshev distance of two successive calculated approximations should be less than eps in succession. Here we use the convention $|-\infty-(-\infty)| := 0$.

randomShifts — A positive integer indicating how many random directions of length tolTrueSCI should be generated. At the end of the algorithm the parameter is passed to the inExactSCI- and the notInExactSCI-functions to determine whether the approximation L of the true lower SCI-bounds is precise or imprecise.

shifts — A matrix with m columns and any number of rows. Each entry must be non-negative. Each row is a direction in the m-dimensional real space. Each row must have at least one positive entry. At the end of the algorithm the parameter is passed to the inExactSCI- and the notInExactSCI-functions to determine whether the approximation L of the true lower SCI-bounds is precise or imprecise. randomShifts must be a positive integer or shifts must contain at least one row. It is recommended to choose randomShifts> 0 or enter a shifts-matrix with at least one row. Entering both parameters is also permissible.

tolTrueSCI — The randomly generated shift-vectors and the row vectors in shifts are rescaled to have length tolTrueSCI. It is recommended to choose tolTrueSCI greater than $\sqrt{m}\cdot$eps and greater than tolBisec.

maxIter — Maximum number of iterations for determining the lower informative SCI bounds.

maxIterBisec	Maximum number of iterations of the bisection method which is used during the algorithm for finding roots.
tolBisec	A non-negative numeric indicating the error tolerance of the bisection method which is used for finding roots.
calculateCSCI	A boolean indicating whether compatible bounds should also be calculated.
checkInput	A boolean specifying whether the entered values should be checked.

Details

It is assumed that there are m parameters of interest $\vartheta_1, \ldots, \vartheta_m$. For each parameter there is a null hypothesis defined as $H_i^{\mu_{0i}} : \vartheta_i \leq \mu_{0i}$. The bounds μ_0 correspond to mu_0. The parameter gMCP or the parameters g and weights define a graphical multiple test procedure for the hypotheses.

The algorithm further assumes that for each hypothesis there is an unbiased point estimator which is (asymptotically) normal. The informSCI-algorithm is based on the p-values from the corresponding (asymptotic) z-tests.

The algorithm terminates when the Chebyshev distance of two successive calculated approximations is less than eps timesSmallerEps-times in succession or if the maximum number of iterations is reached.

The function also tries to find information about the precision of the final calculated approximation of the true lower informative SCI-bounds by calling the inExactSCI- and the notInExactSCI-functions.

For further details see the given references.

Value

The function returns a list containing the calculated lower informative SCI-bounds as well as compatible lower SCI-bounds (if calculateCSCI==TRUE) to allow a comparison. Additionally, the returned list contains elements which can give some information about the precision of the calculated lower informative SCI-bounds compared to the true informative SCI-bounds.

- L: A numeric vector of dimension m of the lower informative SCI-bounds
- rejecHyp: A boolean vector of dimension m indicating the rejected hypotheses of the multiple test induced by the informative SCI-bounds
- diffApprox: A non-negative numeric indicating the Chebyshev distance between the calculated last two approximations of the lower informative SCI-bounds.
- timesApprSmallerEps: A natural number between 0 and timesSmallerEps indicating how many times the Chebyshev distance of two successive calculated approximations in succession was less than eps when the algorithm terminated.
- numIter: A natural number indicating the number of iterations required by the algorithm.
- accuracyL: A string containing information about the collected information about the precision of the calculated lower informative SCI-bounds compared with the true lower SCI-bounds.

If calculateCSCI=TRUE:

- informSCIcompatible: A boolean vector of dimension m indicating whether each informative bound is compatible to the test decision about its corresponding null hypothesis by the underlying graphical test.
- cSCI: A numeric vector of dimension m of compatible lower SCI-bounds from simConfint.
- rejecHypGraph: A boolean vector of dimension m indicating the rejected hypotheses of the underlying graphical test.

References

F. Bretz, W. Maurer, W. Brannath, M. Posch: A graphical approach to sequentially rejective multiple test procedures. Statistics in Medicine 28.4 (2009), pp. 586-604.

K. Strassburger, F. Bretz: Compatible simultaneous lower confidence bounds for the Holm procedure and other Bonferroni based closed tests. Statistics in Medicine 27.4 (2008), pp. 4914-4927.

S. Schmidt, W. Brannath: Informative Simultaneous Confidence Intervals in Hierarchical Testing. Methods of Information in Medicine 53.4 (2014), pp. 278–283.

See Also

gMCP simConfint explore_q

Examples

```
informSCI(gMCP=BonferroniHolm(3), q=0.3, mu_0=-0.5 ,estimates=c(0,2,-1),
SE=0.1467)
Z <- (c(0,2,-1)-(-0.5))/0.1467
informSCI(gMCP=BonferroniHolm(3), q=0.3, mu_0=-0.5, Z=Z, I=1/(0.1467^2),
randomShifts=100)
informSCI(g=matrix(c(0,0,1,0),2,2), weights=c(1,0), q=c(0.0068,1),
mu_0=c(-1,0), pValues=c(0.0002,0.01), SE=c(0.31,1.11), alpha=0.025,
shifts=rbind(c(1,0),c(0,1),c(1,1)))
informSCI(g=matrix(c(0,0,1,0),2,2), weights=c(1,0), q=c(0.0068,1),
mu_0=c(-1,0), pValues=c(0.0002,0.01), I=1/c(0.31,1.11)^2, alpha=0.025,
shifts=rbind(c(1,0),c(0,1),c(1,1)), calculateCSCI = FALSE)
```

---

notInExactSCI *Checking Precision of Approximations*

Description

The function checks whether information about the precision of an approximation for the informative lower SCI-bounds can be collected.

Usage

```
notInExactSCI(
  L,
  randomShifts = 0,
  shifts = NULL,
  tolTrueSCI,
  gMCP = NULL,
  g = NULL,
  weights = NULL,
  q,
  estimates = NULL,
  Z = NULL,
  pValues = NULL,
  SE = NULL,
  I = NULL,
  mu_0,
  alpha,
  checkInput = TRUE
)
```

Arguments

L	An m-dimensional non-negative vector whose entries are the lower bounds of an approximation of the informative SCI.
randomShifts	A positive integer indicating how many random directions of length tolTrueSCI should be generated.
shifts	A matrix with m columns and any number of rows. Each entry must be non-negative. Each row is a direction in the m-dimensional real space. Each row must have at least one positive entry. randomShifts should be a positive integer or shifts should contain at least one row.
tolTrueSCI	The randomly generated shift-vectors and the row vectors in shifts are rescaled to have length tolTrueSCI.
gMCP	An object of class graphMCP indicating the underlying graphical test.
g	Numeric square matrix of transition weights for the graphical test with m rows and m columns. The i-th row of the entered matrix defines the arrows starting from the i-th hypothesis. Each entry has to be between 0 and 1 and each row must sum to a number less than or equal to 1. The diagonal elements must be zero. Entering g and weights can be used as the input as an alternative to specifying gMCP.
weights	Numeric vector of weights of dimension m. It defines the initial proportion of significance level which is assigned to each null hypothesis. Entering g and weights can be used as the input as an alternative to specifying gMCP.
q	A numeric vector of dimension 1 or m defining the information weights for each hypothesis. The entries have to be between 0 and 1 (inclusive). If q has dimension 1, the same information weight is used for each hypothesis.

estimates	An m-dimensional numeric vector of unbiased point estimates for the parameters of interest. Each estimator is assumed to be (asymptotically) normal.
Z	An m-dimensional numeric vector of z-scores for testing the null hypotheses. Z can be entered as an alternative to estimates.
pValues	An m-dimensional numeric vector of p-values from (asymptotic) z-tests for testing the null hypotheses. pValues can be entered as an alternative to estimates and Z.
SE	A positive numeric vector of dimension 1 or m indicating the standard errors of the point estimators. If SE has dimension 1, the same value is used for each estimator.
I	A positive numeric vector indicating the information of the estimators. It can be entered as an alternative to the vector SE. The relationship SE$= 1/\mathrm{I}^{1/2}$ is assumed.
mu_0	A numeric vector of dimension 1 or m defining the bounds of the null hypotheses of the underlying graphical test. If mu_0 has dimension 1, the same value is used for each null hypothesis.
alpha	A numeric defining the overall significance level for the graphical test (i.e. SCIs will have coverage probability of at least $1-$alpha. The parameter must be strictly between 0 and 1.
checkInput	A boolean specifying whether the entered values should be checked.

Details

The function checks if it can be determined whether L can be shifted by a rescaled randomly generated direction or by a rescaled direction in the shift matrix such that it describes valid lower informative SCI bounds. If this is possible, the approximation L is imprecise. (The random directions are generated in such a way that all entries are positive.)

Value

Returns TRUE if we can determine that the approximation is imprecise. Returns FALSE if we cannot determine that the approximation is imprecise. (The approximation may still be imprecise.) Note that inExactSCI and notInExactSCI could both return FALSE.

See Also

informSCI explore_q

Examples

```
g <- matrix(c(0,0,1,0),2,2)
weights <- c(1,0)
q <- c(0.0068,1)
mu_0 <- c(-1,0)
pValues <- c(0.0002,0.01)
SE <- c(0.31,1.11)
alpha <- 0.025
L <- informSCI(g=g, weights=weights, q=q, mu_0=mu_0, pValues=pValues, SE=SE,
```

```
alpha=alpha, eps=1/10, tolBisec=1/10)$L
# When the randomShifts- or shift-parameter in the informSCI-function is
# specified, the notInExactSCI-function is called by the informSCI-function.
# It is also possible to analyse the accuracy of a calculated L (or an
# approximation of the lower informative SCI-bounds) by directly using
# the notInExactSCI-function:
notInExactSCI(L=L, randomShifts=100, tolTrueSCI=1/10^5, g=g, weights=weights,
q=q, pValues=pValues, SE=SE, mu_0=mu_0, alpha=alpha)
```

`sigmaManyToOne` *Calculation of the Covariance Matrix for a Many-to-one-Comparison*

Description

The function calculates the covariance matrix for many-to-one-comparisons. The covariance matrix is calculated for the point estimators, each defined by the difference between the empirical mean of one of the experimental groups and the empirical mean of the control group.

Usage

```
sigmaManyToOne(sampleSizes, sampleSizeControl, varObs, checkInput = TRUE)
```

Arguments

`sampleSizes`	A numeric vector indicating the sample size of each non-control group.
`sampleSizeControl`	A numeric indicating the sample size of the control group.
`varObs`	A positive numeric indicating the variance of the individual observations.
`checkInput`	A boolean specifying whether the entered values should be checked.

Value

Returns covariance matrix.

Examples

```
sigmaManyToOne(sampleSizes=c(89,95), sampleSizeControl=77,
 varObs=10)
```

Index

explore_q, *2*, 3, *9*, *13*, *15*

gMCP, *7*, *13*
graphMCP, *4*, *8*, *10*, *14*

inExactSCI, *2*, 7, *11, 12*
informativeSCI
        *(*informativeSCI-package*)*, 2
informativeSCI-package, 2
informSCI, *2, 3*, *5*, *7*, *9*, 10, *15*

notInExactSCI, *2*, *11, 12*, 13

sigmaManyToOne, 16
simConfint, *7*, *13*