Package 'MIND'

April 6, 2011

Type Package

Title Mathematically-based Integration of heterogeNeous Data

Version 1.0

Date 2011-03-31

Author Jozsef Bukszar <jbukszar@vcu.edu>

Maintainer Jozsef Bukszar <jbukszar@vcu.edu>

Description MIND is a package that implements a method that integrates multiple heterogeneous data sets into a novel data collection (of genetical data) based on a rigorous mathematical foundation.

License GPL (version 2 or later)

LazyLoad yes

R topics documented:

`MIND-package` *Mathematically-based Integration of heterogeNeous Data (MIND)*

Description

The goal is to find genetic units (SNPs, genes, chromosomal segments) related to a disease/disorder in the novel data collection (*NDC*) by utilizing information already available from 'existing' data sets (*EDS*s).

For each genetic unit the code `mind` computes the **cltdr**, the posterior probability that it has an effect in the NDC, based on the information in the NDC and the EDSs. The code `mind` also computes the **ltdr**, the posterior probability that a genetic unit has an effect in the NDC based on the information only in the NDC.

The method is generic in the sense that

1. the EDSs can be of any type whose genetic units can be ranked, e.g. gene expression, linkage data, GWAS, literature search etc, candidate genes. Ties are allowed, e.g. an EDS may provide only binary information (i.e., a genetic unit is implicated or not).
2. the novel data collection can be of any type with the mild restriction that a statistic value needs to be assigned to each genetic unit,
3. the NDC and the EDSs may be of different type.

Also the codes `informtest` and `informtest_binary` test EDS if it is informative to the novel data collection.

Details

Package:	MIND
Type:	Package
Version:	1.0
Date:	2011-03-31
License:	GPL (version 2 or later)
LazyLoad:	yes

The main code that integrates the data sets (`mind`)

As the EDSs can be of different type, their genetic units maybe different. For instance, if we have gene expression data, GWAS and linkage data as EDS, their genetic units are gene, SNP, and chromosomal segment, respectively. Therefore, first the EDSs need to be transformed into data sets based on *test units*, the genetic unit of the NDC. For instance, a gene-based EDS can be transformed into SNP-based EDS by assigning to each SNP the smallest EDS rank (or p-value) of the genes that contain the SNP.

As a next step, prepare the array (vector) *stats_nov* that has the observed statistic values of the test units in the NDC. The test unit whose NDC statistic value is the i-th component in stats_nov will be referred to as the *i-th test unit*, for i=1,2,... Prepare the matrix *lowbetts*, each row of which contains the-lower-the-better-type (e.g. p-values) information for the test units from an existing data set. It is very important that the i-th column in lowbetts has information about the i-th test unit. If there is no information about the i-th test unit in an EDS, then the corresponding entry in lowbetts should be NA. It is also very important that each row of lowbetts has the-lower-the-better-type information of an EDS, that is the lower value a test unit is assigned to, the more likely it is to be related to the disease. Examples of the-lower-the-better-type information are p-values, the negative of the absolute value of test statistics or their ranks, etc. For binary data sets, the corresponding row of lowbetts must contain exactly two different symbols, including NA. If it contains NA, then test units with NA will be considered less preferred based on the EDS. For instance, if the EDS is a list of candidate genes, then the entries in the corresponding row in lowbetts may be 1s for test units 'covered by' the candidate genes, and NAs elsewhere, or any number bigger than 1 elsewhere.

Also the estimated null and alternative, i.e. non-null, c.d.f and p.d.f in the NDC (*cdf1_fun, cdf0_fun, pdf1_fun, pdf0_fun*) as well as the estimated number of the alternative, i.e. non-null, test units in the NDC are needed. There are multiple methods published that estimate the aforementioned distributions. Their performance typically depends on the type of the NDC. For GWAS NDC we developed our own method, which is not part of the package.

For each test unit the code `mind` returns an estimate of the *cltdr*) and the *ltdr*).

The code `cumcltdr_plot`

Plots and returns the expected number of test units with effect when cltdr- or ltdr-based selection is used.

The plot can be used to assess the gain of using information from the existing data sets. For further details see code `cumcltdr_plot`.

Informativeness test (`informtest` and `informtest_binary`)

Tests if an 'existing' data set is informative to the novel data collection.

Uses the O statistic to test if the input 'existing' data set is informative to the novel data collection. In case the input 'existing' data set is informative to the novel data collection the boldface black curve lies above the thin colored curves. Note that for some scenarios dots may be plotted intstead of curves. The boldface black curve represents the O statistics calculated with the novel data collection and the 'existing' data set, whereas each thin colored curve represents the O statistics calculated with the novel data collection and a randomly permutated 'existing' data set.

Author(s)

Jozsef Bukszar

Maintainer: Jozsef Bukszar <jbukszar@vcu.edu>

References

Jozsef Bukszar, Amit N. Khachane, Karoling Aberg, Youfang Liu, Joseph L. McClay, Partick F. Sullivan, and Edwin J.C.G. van den Oord, *A rigorous method for integrating multiple heterogeneous*

databases in large scale genetic studies (submitted).

Examples

```
# Simulating data

numeds <- 300000      #  the number of test units in the EDS  (the test units in
                      #     the EDS that are not in the NDC are ignored through
                      #     the entire analysis)
m1eds <- 5000         #  the number of test units that are alternative (non-null)
                      #     in the EDS
m1star <- 3300        #  the number of test units in the EDS that are alternative
                      #     in the NDC
m1over <- 2500        #  the number of test units that are alternative in the EDS
                      #     and in the NDC
detnov <- 2.0         #  detectability (=effect size*sqrt(sample size)) of the
                      #     alternative test units in the NDC
psi <- 1.6            #  needed to simulate existing data set ranks
numnov <- 2*numeds    #  the number of test units in the NDC
m1nov <- 2*m1star     #  the number of alternative (non-null) test units in the NDC

stats_ndc <- rnorm(numnov)
stats_ndc[1:m1star] = stats_ndc[1:m1star] + detnov
stats_ndc[(numnov-m1nov+m1star+1):numnov] =
                                 stats_ndc[(numnov-m1nov+m1star+1):numnov] + detnov

aux <- rnorm(numeds)
aux[1:m1over] = aux[1:m1over] + psi
aux[(numeds-m1eds+m1over+1):numeds] = aux[(numeds-m1eds+m1over+1):numeds] + psi
                                      # WARNING! If you change the parameters,
                                      # make sure that numeds<numnov-m1nov+m1star+1
                                      # remains valid (important for correctly
                                      # simulate the NDC-EDS structure)

bux <- rank(-abs(aux))  # Existing data set prior (the lower the better type of data)

lowbett <- c(bux,rep(NA,numnov-numeds))   # NAs are put for test units about which
                                          #   we have no EDS information available

dim(lowbett) <- c(1,numnov)          # Needed to be set in a matrix form (each row
                                     #   corresponds to an EDS)


# Defining null and alternative, i.e. non-null, c.d.f and p.d.f in the NDC

cdf1_fun <- function(x) {
    det <- detnov
    absx <- abs(x)
    fval <- pnorm(absx-det) - pnorm(-absx-det)
    return(fval)
}

cdf0_fun <- function(x) {
    absx <- abs(x)
    fval <- pnorm(absx) - pnorm(-absx)
    return(fval)
```

```
}

pdf1_fun <- function(x) {
    det <- detnov
    absx <- abs(x)
    fval <- dnorm(absx-det) + dnorm(absx+det)
    return(fval)
}

pdf0_fun <- function(x) {
    absx <- abs(x)
    fval <- 2*dnorm(absx)
    return(fval)
}

# Data integration (it takes about eight seconds on a 3.33GHz,
#    3.25GB RAM dual core proc. computer  )

res <- mind(stats_ndc, lowbett, m1nov,
            cdf1_fun, cdf0_fun, pdf1_fun, pdf0_fun,noties_innonbinary=TRUE)
                                    # WARNING! noties_innonbinary should be FALSE if
                                    #    there are ties in any EDS.

cltdr <- res$cltdr
ltdr <- res$ltdr


rez<-cumcltdr_plot(res,NN=10000)    #  Plotting the cumulative cltdr/ltdr curves
```

`alkappa_estim` *Alkappa estimator*

Description

Estimates alkappa, a technical parameter.

Usage

```
alkappa_estim(psi_est, cc, mvec, numeds,
     inp_alkappa_est_index = -1, M1_techn = 100, plottingon = FALSE)
```

Arguments

`psi_est`	psi, a technical parameter
`cc`	rough cumulative contribution estimates on *mvec*
`mvec`	grid for the ranks in the existing data set (technical parameter).
`numeds`	the number of test units in the existing data set.
`inp_alkappa_est_index`	the number of the first elements in *mvec* and *cc* that will be taken into account by the estimator (if negative, then all elements will be taken into account).
`M1_techn`	technical parameter
`plottingon`	logical. Do we need a plot or not?

Value

`alkappa`	estimate of alkappa
`cumcontribs`	smoothed cumulative contribution estimates on the elements of *mvec*

`cltdr_calc` *Computing the compound true discovery rate (cltdr)*

Description

Computes the compound true discovery rates, cltdr-s, for test units.

Usage

```
cltdr_calc(f0, f1, numalt, contribs)
```

Arguments

`f0`	array of the null probability density function values on the novel data collection test statistics.
`f1`	array of the alternative probability density function values on the novel data collection test statistics.
`numalt`	the number of alternative test units in the novel data collection.
`contribs`	if matrix, then each of its row contains the contributions of an existing data set; if vector, then it contains the contributions of an existing data set; if NULL, the code computes the local true discovery rate (ltdr) of test units.

Details

If *contribs* is NULL, then the code computes the local true discovery rate (ltdr) of test units, which is the posterior probability that a test unit is alternative based on the novel data collection data only.

Value

`cltdr`	array of the compound true discovery rates (cltdr-s) of test units.
`beta`	combined prior odds.
`adjratio`	adjustment ratio needed to attain that cltdr-s sum up to *numalt*.

combing *Combining multiple grids for 'existing' data set ranks*

Description

Combes multiple grids for a ranked 'existing' data set

Usage

```
combing(mvecom, grids, mvec_finest)
```

Arguments

mvecom	the grid that need to be approximated.
grids	list of multiple grids for a ranked 'existing' data set
mvec_finest	the finest possible grid for the ranks in the existing data set.

contribution_estimator *Contribution estimator for non-binary 'existing' data sets*

Description

Estimates the test units' contribution to the novel data collection from a non-binary 'existing' data set.

Usage

```
contribution_estimator(stats_ndc, lowbett, dvec, predic,
                mvec_finest, m1ran,  M1_techn = 100, max_iter = 1)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
mvec_finest	the finest possible grid for the ranks in the existing data set.
m1ran	technical parameter.
M1_techn	technical parameter.
max_iter	technical parameter: the maximum number of iterations in one of the functions the code calls.

Value

contribs	array of the estimated contributions

contribution_estimators
Contribution estimator

Description

Estimates the test units' contribution to the novel data collection from a non-binary existing data set that contains no ties.

Usage

```
contribution_estimators(stats_ndc, lowbett, dvec, predic,
                            mvec_finest, m1ran, M1_techn, max_iter)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
mvec_finest	the finest possible grid for the ranks in the existing data set.
m1ran	technical parameter.
M1_techn	technical parameter.
max_iter	technical parameter: the maximum number of iterations in one of the functions the code calls.

Value

contribs	array of the estimated contributions

contribution_estimator_aux
Contribution estimator

Description

This is a low level version of the code contribution_estimator. Estimates the test units' contribution to the novel data collection from an existing data set.

Usage

```
contribution_estimator_aux(stats_ndc, lowbett_eds, dvec,
     predic, mvec_entire, mvecom, cc_grid, dd_grid, ee_grid,
             grid_mvecbeg, m1ran, M1_techn = 100, max_iter = 1)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett_eds	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
mvec_entire	the finest possible grid for the ranks in the existing data set.
mvecom	grid for the ranks in the existing data set (technical parameter).
cc_grid	grid for estimating the technical parameter alkappa.
dd_grid	grid for estimating the technical parameter psi.
ee_grid	grid for estimating the technical parameter M1_techn (not in use now).
grid_mvecbeg	grid for estimating the technical parameter psi.
m1ran	technical parameter.
M1_techn	technical parameter
max_iter	technical parameter: the maximum number of iterations in one of the functions the code calls.

Value

contribs	array of the estimated contributions

---

contribution_estimator_binary

Contribution estimator for binary existing data set

Description

Estimates the test units' contribution to the novel data collection from a *binary* existing data set.

Usage

```
contribution_estimator_binary(stats_ndc, lowbett_eds,
                                dvec, predic, plottingon = FALSE)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett_eds	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
plottingon	logical. Do we need a plot or not?

Value

contribs	array of the estimated contributions
contibvals	array of two values, the first value is the positive and the second one is the negative real number that appear among the contributions of the binary data set

---

contribution_estimator_few

Contribution estimator for non-binary 'existing' data sets with few categories

---

Description

Estimates the test units' contribution to the novel data collection from a non-binary 'existing' data set with few categories.

Usage

```
contribution_estimator_few(stats_ndc, lowbett,
                  dvec, predic, mvec_finest, M1_techn = 100)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
mvec_finest	the finest possible grid for the ranks in the existing data set.
M1_techn	technical parameter.

Value

contribs	array of the estimated contributions

---

contribution_estimator_fews

Contribution estimator for non-binary 'existing' data sets with few categories

---

Description

Estimates the test units' contribution to the novel data collection from a non-binary 'existing' data set using the simultaneous estimator of psi and alkappa for the curve-fitting.

Usage

```
contribution_estimator_fews(stats_ndc, lowbett,
           dvec, predic, mvec_finest, mvecom, M1_techn = 100)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	grid for distribution of the novel data collection statistics (technical parameter).
predic	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
mvec_finest	the finest possible grid for the ranks in the existing data set.
mvecom	grid for the ranks in the 'existing' data set for the estimator that estimates psi and alkappa simultaneously.
M1_techn	technical parameter.

Value

contribs	array of the estimated contributions

---

cumcltdr_plot *Cumulative cltdr and ltdr plot*

Description

Plots the estimated cumulative cltdr (black curve) and ltdr (red curve), that is the expected number of test units with effect when cltdr- or ltdr-based selection is used.

Also returns the cummulative cltdr-s/ltdr-s as well as the indices of test units selected by the cltdr- and the ltdr-based method.

Also computes the number of test units that are selected by both the cltdr-based AND the ltdr-based selection, if *NN* test units are selected by both methods.

Usage

```
cumcltdr_plot(res, NN)
```

Arguments

res	output object of the code mind.
NN	the number of test units the cumulative cltdr and ltdr will be calculated for.

Details

The cumulative cltdr/ltdr at *k* is defined as the sum of the largest *k* cltdr/ltdr-s. It equals the expected value of the number of alternative, i.e. non-null, test units (test units with effect) in the *k* test units with the largest cltdr/ltdr-s.

Therefore, the plot and the output can be used to assess the gain of using information from the existing data sets.

Value

`cum_cltdr`	array of the cummulative cltdr-s of test units
`cum_ltdr`	array of the cummulative ltdr-s of test units
`mutual_sel`	the number of mutally selected test units. More precisely, the number of test units that are in the set of test units with largest *NN* cltdr-s AND in the set of test units with the largest *NN* cltdr-s.
`ordc`	array of the indices of test units with the largest cltdr-s.
`ordl`	array of the indices of test units with the largest ltdr-s.

An index of a test unit is the corresponding index in the input *stats_nov* or *lowbetts* of code `mind`.

`cumcontribution_calc_noties` *Rough cumulative contribution estimator*

Description

Calculates the test units' 'roughly estimated' cumulative contributions to the novel data collection from a non-binary 'existing' data set.

Usage

```
cumcontribution_calc_noties(stats_ndc, lowbett_eds, dvec, mvec, F0F1dvec)
```

Arguments

`stats_ndc`	observed statistic values of the test units in the novel data collection.
`lowbett_eds`	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
`dvec`	array of cut-offs for the novel data collection statistics.
`mvec`	array of cut-offs for the existing data set ranks.
`F0F1dvec`	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively, and *dvec* is defined above.

Details

The k-th cumulative contribution is the sum of the k largest contributions. This function returns the rough estimate, i.e. the non-smoothed estimate, of the k-th cumulative contribution for some values of k. The values of k are determined by array *mvec*, defined above.

Value

`cumcontribs`	an array of roughly etimated cumulative contributions.
`statistics`	a matrix of the statistics that were used for estimating the rough cumulative contributions.

cumcontribution_empir_noties

Caculating the statistics used for estimating contribution and evaluating existing data sets

Description

Caculates the statistics used for estimating contribution and evaluating existing data set based. The statistics are calculated for a pair of data sets, the novel data collection and one existing data set, utilizing ultimately only ranked data in both data sets.

Usage

```
cumcontribution_empir_noties(stats_ndc, lowbett_eds, dvec, mvec)
```

Arguments

stats_ndc	observed statistic values of the test units in the novel data collection.
lowbett_eds	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
dvec	array of cut-offs for the novel data collection statistics.
mvec	array of cut-offs for the existing data set ranks.

Details

The statistics are calculated for multiple cut-offs for the ranks in the existing data set (mvec) and for multiple cut-offs for the test statistics in the novel data collection (dvec).

Value

A matrix of statistic values arranged in such a way that each column corresponds to a cut-off for the novel data collection statistics (dvec) and each row corresponds to a cut-off for the existing data set ranks (mvec).

cumcontribution_rough_noties

Rough cumulative contribution estimator

Description

Calculates the test units' 'roughly estimated' cumulative contributions to the novel data collection from a non-binary 'existing' data set.

Usage

```
cumcontribution_rough_noties(ostat, F0F1dvec)
```

Arguments

ostat	a matrix of statistics used for estimating contribution and evaluating existing data set based, it is the output of function cumcontribution_empir_noties.
F0F1dvec	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively, and dvec is array of cut-offs for the novel data collection statistics used to calculate *ostat*.

Details

The k-th cumulative contribution is the sum of the k largest contributions. This function returns the rough estimate, i.e. the non-smoothed estimate, of the k-th cumulative contribution for some values of k. The values of k are determined by vector mvec used to calulate the input matrix *ostat*.

Value

An array of roughly etimated cumulative contributions.

---

cumulative_gamma *Computing rank probabilities*

Description

Calculates very close approximates to Gamma_k for every k in input array *Rvec*.

Gamma_k = Sum_i=1^(k) gamma_i, where gamma_i is the probability that a statistic with rank i comes from an alternative distribution given that there are m1 alternative statistics with parameter delta and m0=numark-m1 null statistic.

Usage

```
cumulative_gamma(Rvec, del, m1, numark)
```

Arguments

Rvec	array of indices. For each index, say k, the sum of probabilities corresponding to ranks smaller than or equal to k will be computed.
del	the detectability in the underlying distribution.
m1	the number of alternative statistics.
numark	the number of statistics.

Value

Array with sums of the probabilities. Each element of the array corresponds to the element of the input array *Rvec* in the same position.

dvec_creator *Grid for novel data collection statistics*

Description

Creates a grid for novel data collection statistics.

Usage

```
dvec_creator(stats_nov, smallrat = 0.0005426192,
                              bigrat = 0.137374, lend = 120)
```

Arguments

stats_nov	observed statistic values of the test units in the novel data collection.
smallrat	technical parameter.
bigrat	technical parameter.
lend	the desired length of *dvec*.

Details

Creates a grid, dvec, for novel data collection statistics. For the first element of array dvec, dvec[1], we have that the proportion of the elements in stats_nov whose absolute value is greater than or equal to dvec[1] is smallrat, i.e. sum(abs(stats_nov)>=dvec[1]) = round(smallrat*length(stats_nov)).

Similarly, for the last element of array dvec, dvec[lend], we have that the proportion of the elements in stats_nov whose absolute value is greater than or equal to dvec[lend] is bigrat, i.e. sum(abs(stats_nov)>=dvec[lend]) = round(bigrat*length(stats_nov)).

The rest of the elements of dvec are equidistantly chosen between the first and the last one.

Value

A grid for novel data collection statistics.

exameds *Examining existing data sets*

Description

Examines an existing data set.

Usage

```
exameds(lowbett)
```

Arguments

lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.

exameds_noties *Examining existing data sets*

Description

Examines an existing data set.

Usage

```
exameds_noties(lowbett)
```

Arguments

lowbett	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.

Note

It differs from the code exameds by assuming that no non-binary data set contains any ties.

gamma_calc *Computing rank probabilities*

Description

Calculates Gamma_k = Sum_i=1^(k) gamma_i, where gamma_i is the probability that a statistic with rank i comes from an alternative distribution given that there are m1 alternative statistics with parameter delta and m0=numark-m1 null statistic.

Usage

```
gamma_calc(k, delta, m1, numark, tolerance = 1e-10)
```

Arguments

k	the number of the smallest (best) ranks the sum of whose probabilities will be computed.
delta	the detectability in the underlying distribution.
m1	the number of alternative statistics.
numark	the number of statistics.
tolerance	the desired accuracy.

Value

The sum of the probabilities.

`informtest` *Informativeness test*

Description

Tests if an 'existing' data set is informative to the novel data collection.

The input arrays *stats_nov* and *lowbett* must be synchornized, i.e. the i-th element of *stats_nov* and the i-th element of *lowbett* must represent the same test unit for every i=1,2,3... Array *lowbett* should contain NA-s for test units the 'existing' data set has no information for.

Usage

```
informtest(stats_nov, lowbett, dpercent = 0.01,
                        Mpercent = 0.1, N = 100, Mlength = 10)
```

Arguments

`stats_nov`	observed statistic values of the test units in the novel data collection.
`lowbett`	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
`dpercent`	the percent of the novel data collection test statistics used for the test.
`Mpercent`	the maximum percent of the 'existing' data set ranks used for the test.
`N`	the number of permutations used for the test.
`Mlength`	the number of percents of the 'existing' data set ranks used for the test.

Details

Uses the O statistic to test if the input 'existing' data set is informative to the novel data collection. In case the input 'existing' data set is informative to the novel data collection the boldface black curve lies above the thin colored curves. Note that for some scenarios dots may be plotted intstead of curves. The boldface black curve represents the O statistics calculated with the novel data collection and the 'existing' data set, whereas each thin colored curve represents the O statistics calculated with the novel data collection and a randomly permutated 'existing' data set.

For the permutation test, the category labels are permuted in the existing data set. The O statistics are calculated with using *dpercent* percent for the novel data collection statistics and the rank cut-offs in array *mvecom* for the 'existing' data set ranks. There are *Mlength* rank cut-offs in the 'unadjusted version' of array *mvecom* equidistantly chosen in such a way that the maximum element is *Mpercent* times the number of test units in the existing data set. Then *mvecom* is adjusted in such a way that for every rank cut-off in *mvecom* all ranks in a category are either smaller or bigger than the rank cut-off. The code returns an array of p-values of the permutation tests. Each p-value is calculated for an element of *mvecom* and *dpercent*.

Value

`p_values`	Array of p-values obtained by the permutation tests. Each element corresponds to a percent in *mvecom* (see details above).

informtest_aux *Auxiliary function for informativeness test*

Description

Auxiliary function for informativeness test `informtest`.

Usage

```
informtest_aux(sta, mvec, dcrit, lenm, recnumeds)
```

Arguments

`sta`	absolute value of observed statistic values of the test units in the novel data collection.
`mvec`	array of cut-offs for ranks in the 'existing' data set.
`dcrit`	critical value for the absolute values of the statistics in the novel data collection.
`lenm`	technical parameter.
`recnumeds`	the reciprocal of the number of test units in the 'existing' data set.

informtest_binary *Informativeness test for binary 'existing' data sets*

Description

Tests if a binary 'existing' data set is informative to the novel data collection.

The input arrays *stats_nov* and *lowbett* must be synchornized, i.e. the i-th element of *stats_nov* and the i-th element of *lowbett* must represent the same test unit for every i=1,2,3... Array *lowbett* may contain NA-s for test units the 'existing' data set has no information for, however, *lowbett* must contain exactly two different simbols (incl. NA).

Usage

```
informtest_binary(stats_nov, lowbett, dpercent = 0.01, N = 100)
```

Arguments

`stats_nov`	observed statistic values of the test units in the novel data collection.
`lowbett`	array that contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
`dpercent`	the percent of the novel data collection test statistics used for the test.
`N`	the number of permutations used for the test.

Details

Uses the O statistic to test if the input binary 'existing' data set is informative to the novel data collection. In case the input binary 'existing' data set is informative to the novel data collection the big black dot appears above the small colored dots. The big black dot represents the O statistic calculated with the novel data collection and the binary 'existing' data set, whereas each small colored dot represents the O statistics calculated with randomly permutated novel data collection and the 'existing' data set.

For the permutation test, the novel data collection statistics are permuted. The O statistics are calculated with using *dpercent* percent for the novel data collection statistics and the rank cut-off dictated by the binary 'existing' data sets. In particular, the rank cut-off seperates the ranks of the two categories in the binary 'existing' data set. The code returns the p-values of the permutation tests, that is the proportion of O statistics on the permuted data greater than the O statistic on the original data.

Value

`p_value`	The p-value obtained by the permutation tests.

`m1_estim` *m1 estimator*

Description

Estimates m1, a technical parameter.

Usage

```
m1_estim(numeds, ee, mvi, psi_est, alkappa_est, m1ran)
```

Arguments

`numeds`	the number of test units in the existing data set.
`ee`	rough cumulative contribution estimates on *mvi*.
`mvi`	grid for the ranks in the existing data set.
`psi_est`	psi estimate
`alkappa_est`	alkappa estimate
`m1ran`	the interval the m1 estimate is searched in.

Value

An m1 estimate.

mind *Integrating information from 'existing' data sets and the novel data collection*

Description

For each test unit (marker) the code returns an estimate of the posterior probability (**cltdr**) that the test unit has an effect in the novel data collection based on the information in the novel data collection and the 'existing' data sets.

The code also returns the posterior probability estimates (**ltdr**) computed without using the 'existing' data sets.

Usage

```
mind(stats_nov, lowbetts, numalt, cdf1_fun, cdf0_fun,
       pdf1_fun, pdf0_fun, noties_innonbinary = FALSE, cutfew = 50)
```

Arguments

stats_nov	observed statistic values of the test units in the novel data collection.
lowbetts	matrix each row of which contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
numalt	the (estimated) number of test units with effect in the novel data collection.
cdf1_fun	the (estimated) alternative cumulative distribution function in the novel data collection.
cdf0_fun	the (estimated) null cumulative distribution function in the novel data collection.
pdf1_fun	the (estimated) alternative probability density function in the novel data collection.
pdf0_fun	the (estimated) null probability density function in the novel data collection.
noties_innonbinary	logical. Set it TRUE, if there is no ties in any non-binary existing data sets. Shortens the running-time if it is TRUE.
cutfew	technical parameter. It is the cut-off value for the number of categories in an existing data set. It determines which smoother method should be used to estimate the cumulative contributions.

Details

The vectors stats_nov and the rows in lowbetts must be synchornized, i.e. the ith element of vector stats_nov and of each row in lowbetts must represent the same test unit.

For test units with no available information in certain 'existing' data sets, NA-s should be put in the corresponding columns and rows of lowbetts.

Value

cltdr	array of the compound local true discovery rate estimates of the test units
ltdr	array of the local true discovery rate estimates of the test units

Warning

The vectors stats_nov and the rows in lowbetts must be synchornized, i.e. the ith element of vector stats_nov and of each row in lowbetts must represent the same test unit.

For test units with no available information in certain 'existing' data sets, NA-s should be put in the corresponding columns and rows of lowbetts.

Author(s)

Jozsef Bukszar

References

For further details please visit the web site http://www.people.vcu.edu/~jbukszar/.

`mind_aux` *Integrating information from 'existing' data sets and the novel data collection*

Description

This is a low level version of the code `mind`. For each test unit the code returns an estimate of the posterior probability (**cltdr**) that the test unit has an effect in the novel data collection based on the information in the novel data collection and the 'existing' data sets. The code also returns the posterior probability estimates (**ltdr**) computed without using the 'existing' data sets.

Usage

```
mind_aux(stats_nov, lowbetts, numalt, f0, f1, dvec, predic,
                    cutfew = 50, M1_techn_rat = 0.1, max_iter = 1)
```

Arguments

`stats_nov`	observed statistic values of the test units in the novel data collection.
`lowbetts`	matrix each row of which contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
`numalt`	the (estimated) number of test units with effect in the novel data collection.
`f0`	the null pdf values on stats_nov, where the null pdf is the null probability distribution function of the statistic in the novel data collection
`f1`	the alternative pdf values on stats_nov, where the alternative pdf is the alternative (non-null) probability distribution function of the statistic in the novel data collection
`dvec`	grid for distribution of the novel data collection statistics (technical parameter)
`predic`	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
`cutfew`	technical parameter. It is the cut-off value for the number of categories in an existing data set. It determines which smoother method should be used to estimate the cumulative contributions.
`M1_techn_rat`	technical parameter
`max_iter`	technical parameter. It is the maximum number of iterations in one of the functions the code calls.

Value

A list with components **cltdr**, which is an array with compound local true discovery rate estimates of the test units, and **ltdr**, which is an array with local true discovery rate estimates of the test units.

Author(s)

Jozsef Bukszar

`mind_bux` *Integrating information from 'existing' data sets and the novel data collection assuming no ties in each non-binary 'existing' data set.*

Description

This is a low level version of the code `mind`. This version assumes that there is no tie in any non-binary 'existing' data set. For each test unit the code returns an estimate of the posterior probability (**cltdr**) that the test unit has an effect in the novel data collection based on the information in the novel data collection and the 'existing' data sets. The code also returns the posterior probability estimates (**ltdr**) computed without using the 'existing' data sets.

Usage

```
mind_bux(stats_nov, lowbetts, numalt, f0, f1, dvec, predic,
                    cutfew = 50, M1_techn_rat = 0.1, max_iter = 1)
```

Arguments

`stats_nov`	observed statistic values of the test units in the novel data collection.
`lowbetts`	matrix each row of which contains the-lower-the-better-type (e.g. p-values, -test stats) information for the test units from an existing data set.
`numalt`	the (estimated) number of test units with effect in the novel data collection.
`f0`	the null pdf values on stats_nov, where the null pdf is the null probability distribution function of the statistic in the novel data collection
`f1`	the alternative pdf values on stats_nov, where the alternative pdf is the alternative (non-null) probability distribution function of the statistic in the novel data collection
`dvec`	grid for distribution of the novel data collection statistics (technical parameter)
`predic`	F0_fun(dvec) - F1_fun(dvec), where F1_fun and F0_fun is the alternativ and the null c.d.f. in the novel data collection, respectively.
`cutfew`	technical parameter.
`M1_techn_rat`	technical parameter.
`max_iter`	technical parameter.

Value

A list with components **cltdr**, which is an array with compound local true discovery rate estimates of the test units, and **ltdr**, which is an array with local true discovery rate estimates of the test units.

Author(s)

Jozsef Bukszar

mvec_creator *Grids for 'existing' data set ranks*

Description

Creates multiple grids for a ranked 'existing' data set for multiple estimators.

Usage

```
mvec_creator(numeds, lenmvec = 30, lenmvecbeg = 20)
```

Arguments

numeds	the number of test units in the existing data set.
lenmvec	technical parameter.
lenmvecbeg	technical parameter.

Value

A list of multiple grids.

param_estim *Psi, alkappa and m1 estimator*

Description

Estimates the technical parameters, psi and alkappa. Used for smoothing the rough cumulative contribution estimates.

Usage

```
param_estim(ee, mvi, dd, mvecbeg, cc, mvec, numeds, grid_mvecbeg,
             m1ran, m1_start = 100, max_iter = 10, tolprec = 0.2,
                                smootheron = TRUE, plottingon = TRUE)
```

Arguments

ee	rough cumulative contribution estimates on *mvi*.
mvi	grid for the ranks in the existing data set.
dd	rough cumulative contribution estimates on *mvecbeg*.
mvecbeg	grid for the ranks in the existing data set for the psi estimator.
cc	rough cumulative contribution estimates on *mvec*.
mvec	grid for the ranks in the existing data set for the alkappa estimator.
numeds	the number of test units in the existing data set.
grid_mvecbeg	grid for *mvecbeg* used for psi estimate after smoothing *dd*.
m1ran	the interval the m1 estimate is searched in.

m1_start	initial number for the m1 estimator.
max_iter	the maximum number of iteration for the m1 estimator.
tolprec	tolerance for the precesion the m1 estimator is required to reach expressed in terms of percentage of m1.
smootheron	logical. Should *dd* be smoothed for the psi estimator or not?
plottingon	logical. Do we need a plot or not?

Value

psi	estimate of psi
alkappa	estimate of alkappa
m1	estimate of m1
cumcontribs	smoothed cumulative contribution estimates on the elements of *mvec*

---

param_estim_simult *Psi and alkappa estimator (simultaneous)*

Description

Estimates the technical parameters, psi and alkappa, simultaneously. Used for smoothing the rough cumulative contribution estimates.

Usage

```
param_estim_simult(cc, mvec, numeds, M1_techn = 100, psi_range = c(0.01, 5),
                                     PSI_tolerance = 0.01, plottingon = TRUE)
```

Arguments

cc	rough cumulative contribution estimates on *mvec*.
mvec	grid for the ranks in the existing data set for the estimator that estimates psi and alkappa simultaneously.
numeds	the number of test units in the existing data set.
M1_techn	technical parameter.
psi_range	the interval the psi estimate is searched in.
PSI_tolerance	tolerance for error of the psi estimator.
plottingon	logical. Do we need a plot or not?

Value

psi	estimate of psi
alkappa	estimate of alkappa
cumcontribs	smoothed cumulative contribution estimates on the elements of *mvec*

plotter_double *Plotting rough and smoothed cumulative contribution estimates*

Description

Plotting rough and smoothed cumulative contribution estimates for two grids of the ranks in the existing data set.

Usage

```
plotter_double(dd, mvecbeg, cc, mvec, cum_contribution_est)
```

Arguments

dd	rough cumulative contribution estimates on *mvecbeg*.
mvecbeg	one of the grids for the ranks in the existing data set.
cc	rough cumulative contribution estimates on *mvec*.
mvec	the second grid for the ranks in the existing data set.
cum_contribution_est	smoothed cumulative contribution estimates

Details

Creates two plots, each corresponds to one of the grids for the ranks in the existing data set.

Warning

Input array *cum_contribution_est* must contain the smoothed cumulative contribution estimates for both grid.

priorprob_calc *Computing prior probabilities*

Description

Computes prior probabilities of test units for an existing data set.

Usage

```
priorprob_calc(contribs, m1perm)
```

Arguments

contribs	array of contribution estimates of an existing data set.
m1perm	the number of alternative test units divided by the the number of test units in the novel data collection.

Value

Array of prior probabilities of test units for an existing data set.

psifind_withratiomatrix *Psi estimator*

Description

Estimates psi, a technical parameter.

Usage

```
psifind_withratiomatrix(yvec, xvec, numeds, M1_techn = 100,
                  psi_range = c(0.01, 5), PSI_tolerance = 0.01)
```

Arguments

yvec	cumulative contribution estimates on *xvec*
xvec	grid for the ranks in the existing data set.
numeds	the number of test units in the existing data set.
M1_techn	technical parameter.
psi_range	the interval the psi estimate is searched in.
PSI_tolerance	tolerance for error of the psi estimator.

Value

An estimate of psi.

psi_alkappa_estim *Psi and alkappa estimator*

Description

Estimates psi and alkappa (technical parameters). It is a wrapper function for psi_estim and alkappa_estim.

Usage

```
psi_alkappa_estim(dd, mvecbeg, cc, mvec, numeds, grid_mvecbeg,
             M1_techn = 100, smootheron = TRUE, plottingon = TRUE)
```

Arguments

dd	rough cumulative contribution estimates on *mvecbeg*.
mvecbeg	grid for the ranks in the existing data set for the psi estimator.
cc	rough cumulative contribution estimates on *mvec*.
mvec	grid for the ranks in the existing data set for the alkappa estimator.
numeds	the number of test units in the existing data set.
grid_mvecbeg	grid for *mvecbeg* used for psi estimate after smoothing *dd*.
M1_techn	technical parameter.
smootheron	logical. Should *dd* be smoothed for the psi estimator or not?
plottingon	logical. Do we need a plot or not?

Value

psi	estimate of psi
alkappa	estimate of alkappa
cumcontribs	smoothed cumulative contribution estimates on the elements of *mvec*

---

psi_estim *Psi estimator*

Description

Estimates psi, a technical parameter.

Usage

```
psi_estim(cc, mvecbeg, numeds, mgrid, M1_techn = 100, plottingon = FALSE,
                              smootheron = TRUE, smoother.method = "spline")
```

Arguments

cc	rough cumulative contribution estimates on *mvecbeg*.
mvecbeg	grid for the ranks in the existing data set.
numeds	the number of test units in the existing data set.
mgrid	grid for *mvecbeg* used for psi estimate after smoothing *cc*.
M1_techn	technical parameter.
plottingon	logical. Do we need a plot or not?
smootheron	logical. Should *cc* be smoothed for the psi estimator or not?
smoother.method	the name of the method used for smoothing *cc*. It can be either "spline" or "quadratic".

Value

An estimate of psi.

`quadratic_approx` *Quadratic approximation*

Description

Calculates quadratic approximation for the response y and predictor x by least square method.

Usage

```
quadratic_approx(x, y)
```

Arguments

`x`	array of predictors
`y`	array of responses

Details

Uses the model y ~ a*x^2 + b*x + c + eps, where eps is a normally distributed random variable with 0 expected value.

Value

`y`	array of response without noise
`comp2`	array of coefficients, a, b and c

Author(s)

Jozsef Bukszar

Index