Package 'matchBox'

April 10, 2015

Type Package

Title Utilities to compute, compare, and plot the agreement between ordered vectors of features (ie. distinct genomic experiments). The package includes Correspondence-At-the-TOP (CAT) analysis.

Version 1.8.0

Date 2012-08-23

Author Luigi Marchionni <marchion@jhu.edu>, Anuj Gupta <Anuj Gupta <agupta52@jhu.edu>

Maintainer Luigi Marchionni <marchion@jhu.edu>, Anuj Gupta <Anuj Gupta <agupta52@jhu.edu>

Depends R (>= 2.8.0)

Imports

Suggests

Description The matchBox package enables comparing ranked vectors of features, merging multiple datasets, removing redundant features, using CAT-plots and Venn diagrams, and computing statistical significance.

License Artistic-2.0

LazyLoad yes

biocViews Software, Annotation, Microarray, MultipleComparison, Visualization

R topics documented:

`matchBox-package` *A package to produce Correspondence-At-Top plots (CAT-plots) between ranked list of genes.*

Description

The matchBox package allows to annotate and compare ranked vectors (e.g. by differential expression) of genomic features (e.g. genes, or probe sets), using CAT curves. A CAT curve displays the overlap proportion between two ranked vectors of identifiers against the number of considered features. This techiques was used for comparing differential gene expression results obtained from different platforms in different laboratories (see Irizarry et al, Nat Methods (2005))

matchBox package features

Enables to filter data.frames containing feature identifiers and ranking statistics;

Enables to identify the common set of features across results from different genomic experiments;

Enables to merge multiple data.frames based on the commons set of features;

Computes the overlap proportion between any pair of ranked features;

Creates plots of the proportion of overlap along with conidence intervals;

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

`calcHypPI` *Probability intervals calculation for CAT curves using the hypergeometric distribution.*

Description

The `calcHypPI` function calculates probability intervals for a correspondence at the top (CAT) curve using the hypergeometric distribution. This function, based on the `qhyper` quantile function, produces a probability intervals matrix to be passed as argument to `plotCat` in order to add probability intervals shades when plotting CAT curves.

Usage

```
calcHypPI(data,  expectedProp = 0.1, prob = c(0.999999,0.999,0.99,0.95))
```

Arguments

`data`	The same data frame used to compute the CAT curves with the `computeCat` function. It contains a column of unique identifiers and at least two columns of ranking statistics.
`expectedProp`	A single numeric value between 0 and 1. This is the proportion of features expected to be corresponding at the top of the ranking. The "expectedProp" argument can be set to NULL if the number of features expected to be similarly ranked is unknown.
`prob`	A numeric vector specifying the probabiliy intervals for the CAT curves to be computed.

Details

The `calcHypPI` uses `qhyper` quantile function to compute the proportions of common features between two ordered vectors for specified quantiles of the hypergeometric distribution. Such proportions are used to add probability intervals to CAT curves computed using ranks (see `computeCat`). The `prob` argument is used to specify the desired probability intervals to be computed. By default this numeric vector is equal to `c(0.999999, 0.999, 0.99, 0.95)`.

To understand the way this function works we can use the analogy of repeated drawing of an increasing number of balls from an urn containing both white and black balls (see `qhyper`). According to this analogy the total number of balls in the urn corresponds to the total number of common features between two ordered vectors that are being compared (e.g. all the genes in common between two genomic studies).

The number of white balls corresponds to the top ranking features that are correctly ordered (successes), while the black balls represent the features that are not correctly ordered (failures).

Finally, according to this analogy, comparing the first top 10 features from each vector will correspond to a first draw of 10 balls from the urn, while comparing the top 20 features to a draw of 20 balls, and so on until all balls are drawn at once.

By default the `calcHypPI` function expects that the top 10% of the features of the two vectors are similarly ordered. This expectation can be modified by the `expectedProp` argument. When

expectedProp is set equal to NULL the number of white balls in the urn (i.e. the top ranking features in the correct order) corresponds to the number of balls that are drawn at each attempt (i.e. the increasing size of top features from each vector that are being compared).

Value

It returns a numeric matrix containing the probability intervals for CAT curves based on equal ranks. The column names of this matrix specifies the quantiles of the hypergeometric distribution used to compute the intervals. The values represent the proportions of overlap associated with the defined quantiles. The resulting matrix object is used to add the probability intervals shades when plotting CAT curves by passing it to the preComputedPI argument of the plotCat function.

Note

This function will take more and more time to run when more and more features are used. For this reason it is convenient to compute the probability intervals separately and store the probability intervals matrix for re-use when plotting the CAT curves.

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate, 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

See Also

See qhyper, plotCat, calcHypPI and computeCat.

Examples

```
###load data
data(matchBoxExpression)

###the column name for the identifiers
idCol <- "SYMBOL"

###the column name for the ranking statistics
byCol <- "t"
```

```
###use lapply to remove redundancy from all data.frames
###default method is "maxORmin"
newMatchBoxExpression <- lapply(matchBoxExpression, filterRedundant, idCol=idCol, byCol=byCol)

###select t-statistics and merge into a new data.frame using SYMBOL
mat <- mergeData(newMatchBoxExpression, idCol=idCol, byCol=byCol)

### compute probability intervals with default values
confInt <- calcHypPI(data=mat)

###structure of confInt
str(confInt)

### compute probability intervals with "expectedProp" set to NULL
confInt2 <- calcHypPI(data=mat, expectedProp=NULL)

###structure of confInt
str(confInt2)
```

`computeCat` *Computing overlap proportions among ordered vectors*

Description

`computeCat` computes the overlap proportions between pairs of ordered vectors of identifiers. The input to this function is a data.frame containing non-redundant identifiers and a number of ranking statistics organized by columns. This function enables comparing all possible pair combinations, or selecting one column as the reference ranking for the remaining. The output of this function can be used as the input to `plotCat`, which creates correspondence at the top curves, as used in Irizarry et al, Nat Methods (2005), for comparing differential gene expression across platforms and labs.

Usage

```
computeCat(data, size=nrow(data), idCol=1, ref,
method = c("equalRank", "equalStat"),
decreasing = TRUE)
```

Arguments

`data`	A data.frame produced by `mergeData`, containing a column of unique identifiers and at least two columns of ranking statistics (e.g. t-statistics, fold-change, Cox coefficients)
`size`	numeric. The number of top ranking statistics to be considered in the computation of the overlap proportions. If omitted all rows in `data` will be considered. If `size` is large computation time may be long.
`idCol`	numeric or character. The index (by default equal to one), or the name of the column containing the common identifiers (e.g. ENTREZID, SYMBOLS, ...).

ref	character. The column name corresponding to the ranking statistics to be used as the reference in all pairs of comparisons.
method	character. The method used to compute the overlap proportion between two ordered vectors of identifiers: either "equalRank" or "equalStat". The first method computed the overlap based on equal ranks, whereas the latter uses equal statistics.
decreasing	logical. This argument defines whether decreasing or increasing ordering should be used

Details

computeCat computes overlapping proportions between pairs of ordered vectors of identifiers. This function first finds all possible pairs of vector combinations, then it computes the corresponding overlapping proportions. If a column is selected as the reference, using the argument ref, only the combinations involving this column will be returned.

Briefly, for each CAT curve two vectors of identifiers are first ordered by the ranking statistics of choice, then the overlap between the two vectors is computed by considering more and more identifiers (vector size).

This function enables to compute overlapping proportions using two distinct methods: "equalRank" or "equalStat". With "equalRank" the overlap is obtained between vectors of the same size using equal ranks, which in turn can potentially correspond to ranking statistics of different magnitude (e.g. the vectors are of the same size, but might have different ranking statistics). With "equalStat" the overlap is obtained between vectors defined by using equal ranking statistics, which can potentially correspond to different rank, and hence to vectors of different size (e.g. the vectors are of different size, but have similar ranking statistics).

Value

A list of lists in which each element correspond to a CAT curve. If a specific reference column is provided through the ref argument, the number of list elements is equal to the number of combinations involving the reference group, otherwise all possible combinations are returned. When the "equalRank" method is used each list element contains only the overlapping proportion, while when the "equalStat" method is used the number of genes with equal statistics is stored along with the overlapping proportion. This output is used to produce CAT curves, using the plotCat function, as described in Irizarry et al, Nat Methods (2005).

Note

Given the combinatorial nature of the computation, a long computational time can be necessary if the input data contains many columns and many rows (number of features). In such a case consider limiting the number of rows used using the size argument.

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

See Also

See `mergeData` and `plotCat`.

Examples

```
###load data
data(matchBoxExpression)

###the column name for the identifiers
idCol <- "SYMBOL"

###the column name for the ranking statistics
byCol <- "t"

###use lapply to remove redundancy from all data.frames
###default method is "maxORmin"
newMatchBoxExpression <- lapply(matchBoxExpression, filterRedundant, idCol=idCol, byCol=byCol)

###select t-statistics and merge into a new data.frame using SYMBOL
mat <- mergeData(newMatchBoxExpression, idCol=idCol, byCol=byCol)

###Compute CAT for decreasing t-statistics: all genes
cpH2L <- computeCat(mat, idCol=1,decreasing=TRUE, method="equalRank")

###Compute CAT for increasing t-statistics:only the first 300 genes
cpL2H <- computeCat(mat, idCol=1, size=300, decreasing=FALSE, method="equalRank")

###Compute CAT for increasing t-statistics:only the first 300 genes
###use the second column as the reference
cpL2H.ref <- computeCat(mat, idCol=1, size=300, ref="dataSetA.t",
  decreasing=FALSE, method="equalRank")
```

`filterRedundant` *This functions removes redundant features from a data.frame*

Description

Prior computing proportion of overlap between ranked vector of features it is necessary to remove the redundant features. This can be accomplished using a number of methods implemeted in the `filterRedundant` function, as explained below.

Usage

```
filterRedundant(object,
    method=c("maxORmin", "geoMean", "mean", "median","random"),
    idCol=1, byCol=2, absolute=TRUE, decreasing=TRUE, trim=0, ...)
```

Arguments

`object`	a data.frame from which redundant features (rows) must be removed.
`method`	character. The method used for removing redundancy. Currently available methods are: `maxORmin`, `geoMean`, `random`, `mean`, `median`, (see Details below).
`idCol`	character or numeric. Name or index of the column containing redundant identifiers (e.g. ENTREZID, SYMBOLS, ...).
`byCol`	character or numeric. Name or index of the column containing the ranking statistics (used only with `maxORmin` method).
`absolute`	logical. Indicates whether the absolute statistics, as defined by `byCol`, should be used when reordering (used only with `maxORmin` method).
`decreasing`	logical. Indicates whether reodering should be decreasing or not (used only with `maxORmin` method).
`trim`	numeric. Indicates whether a trimmed mean should be computed (used only with `mean` method).
`...`	further arguments to be passed (not currently implemented).

Details

The `maxORmin` method removes redundant features by selecting the rows that correspond to the maximum or minimum value of a selected statistics. With this approach redundant features are first ranked in increasing or decreasing order, as defined by the `decreasing` argument, using the ranking statistics defined by `byCol`, either in their original or absolute scale, as defined by `absolute` argument. Subsequently data.frame rows corresponding to redundant identifiers are removed, after these have been identified in the column defined by the `idCol`, using the `duplicated` function.

The `mean`, `median`, `geoMean`, and `random` methods provide alternative ways for summarizing numerical values corresponding to redundant features, as defined by the `idCol` argument: `mean` takes the average, `median` the median, `geoMean` the geometric mean, `random` select a random value.

Value

A data.frame with fewer rows with respect to the input one, unique by the identifier specified by the `idCol` argument.

Note

`filterRedundant` is a utility function providing various methods to remove redundant rows from a data.frame. The choice of the method depends on the nature of the values, and the final goal.

Therefore caution should be used when taking the mean or the median across few values, or passing the arguments with the `minORmax` method (for instance it would make no sense at all to use a decreasing ordering if the ranking statistics is a p-value).

Author(s)

Luig Marchionni <marchion@jhu.edu>

See Also

See `duplicated`.

Examples

```
###load data
data(matchBoxExpression)

###check whether there are redundant identifiers
sapply(matchBoxExpression,nrow)

###the column name for the identifiers
idCol <- "SYMBOL"

###the column name for the ranking statistics
byCol <- "t"

###use lapply to remove redundancy from all data.frames
###default method is "maxORmin"
newMatchBoxExpression <- lapply(matchBoxExpression, filterRedundant, idCol=idCol, byCol=byCol)

###recheck number of rows
sapply(newMatchBoxExpression, nrow)
```

`matchBoxExpression` *Example data: ranking from three differential gene expression experiments*

Description

List of differentially expressed genes from three distinct experiments from which the identifiers and the ranking statistics to be used for computing overlap proportions will be retrieved. This type of object is the starting point of a CAT-plot analysis,

Usage

```
data(matchBoxExpression)
```

Format

This object is a list of data.frames containing at least two common columns, one for the identifiers and one for the ranking statistics. The common columns must have the same column names. In the provided example the following columns are present in each data.frame:

`SYMBOL`: Gene symbol column;

`GENENAME`: Gene name column;

`ENTREZID`: ENTREZ Gene identifier column;

`logFC`: Log2 fold-change column;

`AveExpr`: Average expression (A-value) column;

`t`: moderated t-statistics column;

`P.Value`: P-value column;

`adj.P.Val`: adjusted P-value column;

`B`: B-statitics (log-odds) column;

The statistics were computed using the `topTable` function from `limma`.

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

mergeData *Merging data.frames based on common identifiers*

Description

This utility function is used for merging specific columns from a set of distinct data.frames based on a specific set of identifiers. For instance this utility function can be used to retrieve from multiple data.frames the ranking statistics and the identifiers that will be used for computing the correspondence at the top curves.

Usage

```
mergeData(listOfDataFrames, idCol=1, byCol=2)
```

Arguments

listOfDataFrames	list. This object is a list of distinct data.frames to be merged based on common identifiers. The data.frames to be merged must contain at least two common columns, one for the identifiers (as specified by idCol), and one for the ranking statistics (as specified by byCol). Redundant features are not allowed, and should be previously removed using filterRedundant.
idCol	character or numeris. Name or index of the column containing the common identifiers (e.g. ENTREZID, SYMBOLS, ...).
byCol	character or numeric . Name of index the column containing the ranking statistics.

Details

This function first identifies the common set of features across all the data.frames contained in the listOfDataFrames object. Subsequently, for this common set of features, it returns a single data.frame containing the ranking statistics values of choice collected from each data.frame.

Value

A data.frame containing the identifiers and the ranking statistics common to all data.frames in listOfDataFrames to be used for computing the correspondence at the top (see Irizarry et al, Nat Methods (2005))

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

See Also

See `filterRedundant`.

Examples

```
###load data
data(matchBoxExpression)

###the column name for the identifiers
idCol <- "SYMBOL"

###the column name for the ranking statistics
byCol <- "t"

###use lapply to remove redundancy from all data.frames
###default method is "maxORmin"
newMatchBoxExpression <- lapply(matchBoxExpression, filterRedundant, idCol=idCol, byCol=byCol)

###select t-statistics and merge into a new data.frame using SYMBOL
mat <- mergeData(listOfDataFrames = newMatchBoxExpression, idCol = idCol,
byCol = byCol)

###structure of mat
str(mat)
```

`plotCat` *Plotting correspondence at the top curves*

Description

This function plots corresponding at the top (CAT) curves using overlap proportions computed by `computeCat`. A number of arguments can be used for a pretty display, and for annotating the plot, and adding the legend

Usage

```
plotCat(catData, whichToPlot = 1:length(catData),
     preComputedPI, size=500, main="CAT-plot",
     minYlim=0, maxYlim=1, col, pch, lty, cex=1, lwd=1,
     spacePts=10, cexPts=1, legend=TRUE, legendText,
     where="center", legCex=1,
   plotLayout=layout(matrix(1:2, ncol = 2, byrow = TRUE), widths = c(0.7, 0.3)), ...)
```

Arguments

catData	The ouput list obtained from computeCat, containing the overlapping proportions among pairs of ordered vectors. Names in catData are used for annotating the legend if legendText is not provided (see below).
whichToPlot	numeric vector. Indexes corresponding to the elements of catData to be selected for displaying in the plot.
preComputedPI	numeric matrix. Probability intervals computed using the calcHypPI function. It is used to add grey shades to the plot corresponding to overlapping proportion probabilities based on the hypegeomtric distribution. If missing no PI will be added to the plot.
size	numeric. The number of top ranking features to be displayed in the plot.
main	character. The title of the plot, if not provided, main default is "CAT-plot".
minYlim	numeric. The lower numeric value of the y axis, to be displayed in the plot.
maxYlim	numeric. The upper numeric value of the y axis, to be displayed in the plot.
col	character or numeric. Vector specifying colors for CAT curves plotting. col default uses rainbow function to generate a color vector for all CAT curves in catData. When provided by the user, it will be recycled if needed.
pch	graphical parameter. pch specifies point types for annotating the CAT curves. If not provided, pch is created by default, and recycled if needed. See par for details.
lty	graphical parameter. The type of line for the plot. If not provided generated by default, recycled id needed. See par if needed.
cex	numeric. Standard graphical parameter useful for controlling axes and title annotation size. See par.
lwd	numeric. Standard graphical parameter useful for controlling line size. See par.
spacePts	numeric. Specifies the interval to be used for adding point labels on the CAT curves (evenly spaced along the x axis dimention).
cexPts	numeric. Graphical parameter useful for controlling points size used for annotating CAT-plot lines.
legend	logical. Wheter a legend should be added to the plot.
legendText	character. A vector used for legend creation. legendText default correspond to catData names.
where	character. The position of the plot where the legend will be created; where default is center, see legend.

legCex	numeric. Graphical parameter setting the font size for the legend text.
plotLayout	A layout matrix to arrange the plot and the legend. For further details see layout.
...	Other graphical parameters, currently passed only to legend (e.g. the number of columns to be used in the legend, or the legend background).

Details

This function uses outputs from computeCat and calcHypPI to plot the CAT curves and add grey shades corresponding to probability intervals. The default plot uses a pre-specified layout with separate areas for the plot and the legend. If not specified by the user, different points, colors and line types are used for the different CAT curves. If the CAT curves where computed using equal ranks (e.g. "equalRank" was passed to the method argument of the computeCat function), the user has the option of adding probability intervals to the plot. Such intervals must be pre-computed using the calcHypPI function.

Value

Produces an annotated CAT plot.

Note

In order to make the "best looking" plot for your needs you must play around with graphical parameters

Author(s)

Luigi Marchionni <marchion@jhu.edu>

References

Irizarry, R. A.; Warren, D.; Spencer, F.; Kim, I. F.; Biswal, S.; Frank, B. C.; Gabrielson, E.; Garcia, J. G. N.; Geoghegan, J.; Germino, G.; Griffin, C.; Hilmer, S. C.; Hoffman, E.; Jedlicka, A. E.; Kawasaki, E.; Martinez-Murillo, F.; Morsberger, L.; Lee, H.; Petersen, D.; Quackenbush, J.; Scott, A.; Wilson, M.; Yang, Y.; Ye, S. Q. and Yu, W. Multiple-laboratory comparison of microarray platforms. Nat Methods, 2005, 2, 345-350

Ross, A. E.; Marchionni, L.; Vuica-Ross, M.; Cheadle, C.; Fan, J.; Berman, D. M.; and Schaeffer E. M. Gene Expression Pathways of High Grade Localized Prostate Cancer. Prostate, 2011, 71, 1568-1578

Benassi, B.; Flavin, R.; Marchionni, L.; Zanata, S.; Pan, Y.; Chowdhury, D.; Marani, M.; Strano, S.; Muti, P.; and Blandino, G. c-Myc is activated via USP2a-mediated modulation of microRNAs in prostate cancer. Cancer Discovery, 2012, March, 2, 236-247

See Also

See computeCat, calcHypPI, rainbow, par, legend, and layout.

Examples

```
###load data
data(matchBoxExpression)

###the column name for the identifiers and  the ranking statistics
idCol <- "SYMBOL"
byCol <- "t"

####filter the redundant features using SYMBOL and t-statistics
matchBoxExpression <- lapply(matchBoxExpression, filterRedundant, idCol=idCol, byCol=byCol)

###select and merge into a matrix
mat <- mergeData(matchBoxExpression, idCol=idCol, byCol=byCol)

###COMPUTE CAT
cpH2L <- computeCat(mat, idCol=1, size=round(nrow(mat)/1),
decreasing=TRUE, method="equalRank")

###CATplot without probability intervals
par(mar=c(3,3,2,1))
plotCat(cpH2L, main="CAT-plot, decreasing t-statistics",
cex=1, lwd=2, cexPts=1.5, spacePts=15,
legend=TRUE, where="center",
legCex=1, ncol=1)

###compute probability intervals
confInt <- calcHypPI(data=mat)

###CATplot with probability intervals
par(mar=c(3,3,2,1))
plotCat(cpH2L, main="CAT-plot, decreasing t-statistics, probability intevals",
cex=1, lwd=2, cexPts=1.5, spacePts=15,
legend=TRUE, where="center",
legCex=1, ncol=1)
```

Index