# SAGx

April 19, 2010

## `clin2mim` *Output a script file to WinMIM, linking clinical data and gene expression*

### Description

Given a clinical variable, it produces a script file for WinMIM by calculating means and covariances and for the N most highly correlated probes (in absolute value). Here N is an input parameter, but a recommended value 10. WinMIM can find a relevant graphical model for the dependencies between the probes and the clinical variable.

### Usage

```
clin2mim(variable="FEV1.ACTUAL",data=dbs,clindat=clinical,probes=probes,N=10,out
```

### Arguments

| | |
|---|---|
| `variable` | Clinical variable to be examined |
| `data` | The input data set, with subject id in first column. |
| `clindat` | The input clinical data, with subject id in first column |
| `probes` | The name of the probes in the order of *data* |
| `N` | The number of highly correlated probes to be studied |
| `out` | The MIM script file |

### Value

The correlation matrix

### Note

David Edwards' program WinMIM can be found on StatLib (`http://lib.stat.cmu.edu/graphmod/`). In MIM issue `input mimscript.txt` and the calculations to find a model will start. When finished go to the Graphics menu and click on `Independence Graph`. The resulting graph can be exported both to WMF and LaTeX.

### Author(s)

Per Broberg

### References

Edwards, David (1995) *Introduction to Graphical Modelling*. Springer-Verlag
Lautitzen, Steffen (1996) *Graphical Models*. Oxford University Press
Whittaker, Joe (1990) *Graphical Models in Multivariate Analysis*. Wiley

## `cluster.q` *Clustering Goodness measured by Q2*

### Description

Calculates a goodness of clustering measure based prediction sum squares.

### Usage

```
cluster.q(data,cluster)
```

### Arguments

| | |
|---|---|
| `data` | The data matrix |
| `cluster` | a vector descibing the cluster memberships |

### Value

The clustering mean Q2

### Author(s)

Per Broberg

### References

Eriksson, L., Johansson, E., Kettaneh-Wold, N. and Wold, S. (1999) *Introduction to Multi- and Megavariate Data Analysis using Projection Methods (PCA \& PLS)*, Umetrics

## `estimatep0` *Estimate proportion unchanged genes*

### Description

The function uses the vector of p-values to estimate p0.

### Usage

```
estimatep0(ps = pp, B = 500, range = seq(0,0.95, by = 0.05))
```

### Arguments

| | |
|---|---|
| `ps` | the vector of p-values, e.g. from firstpass |
| `B` | the number of Bootstrap samples |
| `range` | the values considered |

### Value

the value of p0, the proportion unchanged genes

### Author(s)

Per Broberg

### References

Storey, J. A Direct Approach to the False Discovery Rate, Technical Report Stanford (2001)

## `fetchSignal` *Fetch data from the GATC database*

### Description

Fetch FILENAME, PROBESET, SIGNAL and ABS_CALL from the GATC database

### Usage

```
fetchSignal(experiment="AZ33 ALI", channel, chip="HG_U95Av2")
```

### Arguments

| | |
|---|---|
| `experiment` | The name of the experiment corresponding to an individual chip |
| `channel` | The channel to the database |
| `chip` | the chip type |

### Value

dataframe with columns

### Author(s)

Ported to R by Per Broberg. Original Oracle code by Petter Hallgren, with input from Petra Johansson.

### Examples

```
## Not run:
# Do not run example 1. Fetch Probeset, Signal, ABS_CALL and CHIP for one sample.
library(RODBC)
(channel<-odbcConnect("DSN",uid="USERID",pwd="PASSWORD"))
ali.data <-fetchSignal(experiment="AZ33 ALI", channel, chip="hg_u95a")
 colnames(ali.data)
#[1] "FILENAME" "PROBESET" "SIGNAL" "ABS_CALL" "CHIP"

# Do not run example 2
t1 <- paste("select q1.name as name from experiment q1, physical_chip q2, chip_design q3"
t2 <- paste("where q1.physical_chip_id=q2.id and q3.id=q2.design_id and ")
t3 <- paste("upper(q1.name) like '
Ids <- sqlQuery(channel,paste(t1,t2,t3) )
# fetch Signal from GATC corresponding to the U95A chip for all samples in experiment. #
tmp <- apply(Ids,1,toupper)
probes <- data.frame(fetchSignal(experiment=tmp[1],channel, chip="hg_u95a")[,"PROBESET"])
test <- matrix(nrow=nrow(as.data.frame(probes)),ncol=nrow(Ids))
for(i in 1:nrow(as.data.frame(tmp))){
   test[,i] <- fetchSignal(experiment=tmp[i],channel, chip="hg_u95a")[,"SIGNAL"]
}
codes <- data.frame(apply(Ids,1,code<-function(x) substr(x,1,5)))
colnames(test) <- as.character(t(codes))
test <- test[,order(colnames(test))]

## End(Not run)
```

## `firstpass` *First pass description of GeneChip data*

### Description

Does a first-pass analysis for a comparative experiment. This includes the calculation of means and confidence intervals for the groups, and finally a Kruskal-Wallis p-value for the null hypothesis of no difference

### Usage

```
firstpass(data = D, probes = probes , g, log = FALSE, present = NULL, labels = N
```

### Arguments

`data` A data frame with one array in each column

`probes` a vector containing the names of the probes in the same order as rows in D

`g` A vector with the groups for the arrays, eg. TREATMENT and CONTROL

`present` A dataframe with the Present calls, 3 = P, 2 = M, 1 = A.

`log` if TRUE then data are log transformed through t(x) = log(1+x) and geometric means are calculated

`labels` a vector of labels given the group means

`output.data` if T the raw data are included in the output

## Details

A speed-up for Wilcoxon based on Kronecker products was put in place with SAGx v.1.4.5. Ties are currently not taken into account in Wilcoxon.

## Value

A dataframe with the coumns PROBES, followed by group means and sd's, lower confidence intervals and then, upper confidence interval (confidence level 95%), and followed a Kruskal-Wallis p-value, and finally the input data,. If present names a dataframe holding the present calls the proportion present is calculated. Furthermore, if there are two groups the difference in group means is added.

## Examples

```
## Not run:
# not run
 g <- c(rep(1,4),rep(2,4)); labs <- c("Mean Diet","Mean Control"); probes <- paste("Probe
 firstpass(data = utmat[1:2,], probes = probes[1:2], g, log = FALSE, labels = labs)
#  Probesets         Mean Diet      Mean Control             LCL.1             LCL.2
#1   Probe 1 -12.3444460036497 -11.7495704973055 -12.9047961446666 -12.2832657957485 -11.
#2   Probe 2 -7.99773926405627 -8.02799133391929 -8.47704512876227 -8.19487551919835 -7.5
#          Difference  Subject 1  Subject 2  Subject 3  Subject 4  Subject 5  Subject 6
#1 -0.594875506344176 -12.345150 -11.805071 -12.776232 -12.451332 -11.595748 -12.320430 -
#2 0.0302520698630131  -7.660097  -8.157944  -8.404433  -7.768484  -7.979951  -8.017327

## End(Not run)
```

---

# `fom` *Clustering Figure of Merit*

---

## Description

Goodness of clustering measure based on prediction error.

## Usage

```
fom(data,cluster)
```

## Arguments

| | |
|---|---|
| `data` | The data matrix |
| `cluster` | a vector descibing the cluster memberships |

## Details

The criterion in the Reference is not correct in the article (i.e. does not follow from the premises), but has been corrected here.

## Value

The Figure of Merit measure of the current clustering

### Author(s)

Per Broberg

### References

Yeung, K.Y., Haynor, D.R. and Ruzzo, W.L. (2001) Validating clustering for gene expression data. *Bioinformatics* Vol. 17, pp. 309-318

---

## `fp.fn` *Calculation of fp and fn based on a vector of p-values*

---

### Description

Based on a vector of p-values the proportion false positive (fp) and the proportion false negative are calculated for each entry, assuming that one to be the last to be called significant. The sum of fp and fn is also calculated (errors). Furthermore, an estimate of the proportion unchanged together with the number of the entry with minimum errors.

### Usage

```
fp.fn(ps = pvals, B = 100)
```

### Arguments

| | |
|---|---|
| `ps` | a vector of p-values |
| `B` | the number of bootstrap loops done by the function estimatep0 called by fp.fn |

### Value

A list with components

| | |
|---|---|
| `p0` | the estimated proportion unchamged |
| `fp` | the estimated proportion false positives |
| `fn` | the estimated proportion false negatives |
| `N` | the number of the p-value (significance level) that gives minimum fp + fn |

### Author(s)

Per Broberg

## Fstat *Calculation of F statistic by gene given a linear model*

### Description

Calculates F statistic.

### Usage

```
Fstat(indata =  M, formula1 = ~as.factor(g), formula0 = "mean", design1 = NULL,
```

### Arguments

| | |
|---|---|
| `indata` | The data matrix |
| `formula1` | a formula descibing the alternative linear model |
| `formula0` | a formula describing the nullmodel. Use linear models syntax, except for one-way ANOVA ("mean") |
| `design1` | the alternaive design matrix. If not NULL it overrides the formula argument |
| `design0` | the null design matrix. If not NULL it overrides the formula argument |
| `B` | the number of bootstrap replicates |

### Value

A list with the components

| | |
|---|---|
| `Fstat` | the value of the F statistic |
| `fnum` | the numerator degrees of freedom |
| `fdenom` | the denominator degrees og freedom |
| `design1` | the alternative design matrix |
| `design0` | the null design matrix |
| `SS1` | the sum of squares in the denominator of the F-statistic |
| `SS0` | the sum of squares in the numerator of the F-statistic |
| `pvalue` | the p-value for testing the alternative vs the null model |

### Author(s)

Per Broberg

### Examples

```
## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
## Page 9: Plant Weight Data.
 ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
 trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
 group <- gl(2,10,20, labels=c("Ctl","Trt"))
 weight <- c(ctl, trt)
 anova(lm.D9 <- lm(weight ~ group))
# Analysis of Variance Table
# Response: weight
#           Df Sum Sq Mean Sq F value Pr(>F)
#group       1 0.6882  0.6882  1.4191  0.249
#Residuals 18 8.7292  0.4850

 Fstat(indata = rbind(weight,weight),formula1=~group) # Fstat will need at least two gene
#$Fstat
#  weight   weight
#1.419101 1.419101

#$fnum
#[1] 18

#$fdenom
#[1] 1

#$design1
#   (Intercept) groupTrt
#1            1        0
#2            1        0
#3            1        0
#4            1        0
#5            1        0
#6            1        0
#7            1        0
#8            1        0
#9            1        0
#10           1        0
#11           1        1
#12           1        1
#13           1        1
#14           1        1
#15           1        1
#16           1        1
#17           1        1
#18           1        1
#19           1        1
#20           1        1
#attr(,"assign")
#[1] 0 1

# $design0
# NULL

# $SS1
# weight  weight
#8.72925 8.72925

#$SS0
#  weight   weight
#0.688205 0.688205
```

## gap *GAP statistic clustering figure of merit*

### Description

Calculates a goodness of clustering measure based on the average dispersion compared to a reference distribution.

### Usage

```
gap(data = swiss,class = g, B = 500, cluster.func = myclus)
```

### Arguments

| | |
|---|---|
| `data` | The data matrix, with samples (observations) in rows and genes (variables)in columns |
| `class` | a vector descibing the cluster memberships of the rows of data |
| `B` | the number of bootstrap samples |
| `cluster.func` | a function taking the arguments `data` and `k` (number of clusters) and outputs cluster assignments as list elements `cluster` ( accessed by `object$cluster` ). |

### Value

The GAP statistic and the standard deviation

### Author(s)

Per Broberg

### References

Tishirani, R., Walther, G. and Hastie, T. (2000) Estimating the number of clusters in a dataset via the Gap statistic. *Technical Report* Stanford

### Examples

```
library("MASS")
data(swiss)
cl <- myclus(data = swiss, k = 3)
gap(swiss,cl$cluster)
```

## `GSEA.mean.t` *Gene Set Enrichment Analysis using output from samroc*

### Description

Based on a list of gene sets, e.g. pathways, in terms Affymtrix identifiers, these sets are ranked with respect to regulation as measured by an effect in a linear model using the SAM statistic. Typical applications include two-group comparisons or simple linear regression to clinical variable or gene expression of a given gene.

### Usage

```
GSEA.mean.t(samroc = samroc.res, probeset = probeset,
pway = kegg, type = c("original","absolute", "maxmean"), two.side = FALSE, cutof
```

### Arguments

`samroc` an object of class samroc.result

`probeset` the Affymetrix identifiers

`pway` a list of pathways or gene sets

`type` if "absolute" value of the absolute value of the samroc test statistic is used. If "original" no transformation. "maxmean" not available.

`two.side` if TRUE a two-sided test is performed. Currently only two-sided test when type = "original" and else one-sided

`cutoff` Gene sets with the number of members not falling within the interval given by *cutoff* are excluded

`restand` if TRUE a 'restandardization' following Efron and Tibshirani (2006) is performed

### Details

Restandardization based on Efron and Tibshirani (2006) introduced. For normal approximation of the gene set statistic both the mean of the statstic, or the variance (and likewise for the Wilcoxon statistic), are obtained from the permutation distribution included in the samroc.result object. Note that this will account for the dependency between genes.

### Value

A matrix with columns normal approximation p-values, mean statistic, median statistic, and if type = "original", also Wilcoxon signed ranks statistic based p-value.

### Author(s)

Per Broberg

### References

Tian, Lu and Greenberg, Steven A. and Kong, Sek Won and Altschuler, Josiah and Kohane, Isaac S. and Park, Peter J. (2005) Discovering statistically significant pathways in expression profiling studies, *PNAS* Vol. 102, nr. 38, pp. 13544-13549

Bradley Efron and Robert Tibshirani (2006) On testing of the significance of sets of genes, Technical report, Stanford

---

## `JT.test` *Jonckheere-Terpstra trend test*

---

### Description

The test is testing for a monotone trend in terms of the class parameter. The number of times that an individual of a higher class has a higher gene expression forms a basis for the inference.

### Usage

```
trendA <- JT.test(data, class, labs = c("NS", "HS", "COPD0", "COPD1", "COPD2"),
```

## Arguments

`data` A matrix with genes in rows and subjects in columns

`class` the column labels, if not an ordered fctor it will be redefined to be one.

`labs` the labels of the categories coded by class

## Details

Assumes that groups are given in increasing order, if the class variable is not an ordered factor, it will be redefined to be one. The p-value is calculated through a normal approximation.

The implementation owes to suggestions posted to R list.

The definition of predictive strength appears in Flandre and O'Quigley.

## Value

an object of class JT-test, which extends the class htest, and includes the following slots

`statistic` the observed JT statistic

`parameter` the null hypothesis parameter, if other value than 0.

`p.value` the p-value for the two-sided test of no trend.

`method` Jonckheere-Terpstra

`alternative` The relations between the levels: decreasing, increasing or two-sided

`data.name` the name of the input data

`median1 ... mediann`
the medians for the n groups

`trend` the rank correlation with category

`S1` Predictive strength

## Author(s)

Per Broberg, acknowledging input from Christopher Andrews at SUNY Buffalo

## References

Lehmann, EH (1975) *Nonparametrics: Statistical Methods Based on Ranks* p. 233. Holden Day
Flandre, Philippe and O'Quigley, John, *Predictive strength of Jonckheere's test for trend: an application to genotypic scores in HIV infection*, Statistics in Medicine, 2007, 26, 24, 4441-4454

## Examples

```
# Enter the data as a vector
A <- as.matrix(c(99,114,116,127,146,111, 125,143,148,157,133,139, 149, 160, 184))
# create the class labels
g <- c(rep(1,5),rep(2,5),rep(3,5))
# The groups have the medians
tapply(A, g, median)
# JT.test indicates that this trend is significant at the 5% level
JT.test(data = A, class = g, labs = c("GRP 1", "GRP 2", "GRP 3"), alternative = "two-side
```

## list.experiments *Display all experiment names and id's*

### Description

Display all experiment names and id's in the GATC database

### Usage

```
list.experiments(channel, chip = "HG_U95Av2")
```

### Arguments

| | |
|---|---|
| `channel` | the ODBC channel set up through RODBC |
| `chip` | the chip type |

### Details

The GATC database has caused some problems by switching between upper and lower case in an erratic manner. To solve this all names are changed to upper case in the identification of experiments. Thus the function will not distinguish between the experiments 'A' and 'a', but with any sensible naming strategy, the restriction is without consequence

### Value

dataframe with column EXPERIMENT

### Examples

```
# Not run
## Not run: library(Rodbc)
channel <- odbcConnect(DBN, USRID, PWD)
ut <- list.experiments(channel, chip = "hu6800")
colnames(ut)
#[1] "EXPERIMENT"

## End(Not run)
```

## list.intersection.p *p-value for intersection of two gene lists.*

### Description

Calculates a p-value for observing a number of probe sets common to two lists drawn from the same chip.

### Usage

```
list.intersection.p(N = 14000, N1 = 100, N2 = 200, common = 30)
```

### Arguments

| | |
|---|---|
| `N` | The selectable number of probe sets |
| `N1` | the number of probe sets on the first list. |
| `N2` | the number of probe sets on the second list |
| `common` | the number of probe sets in common to the two lists. |

### Value

the p-value giving the probability of observing by chance at least as many in common as was actually observed.

### Author(s)

Per Broberg

## `mat2TeX` *Ouput matrix to LaTeX*

### Description

The function outputs a matrix to a LaTeX table

### Usage

```
mat2TeX(mat, digits = 4, rowNameTitle = "", file = "",
 roundNum = NULL, rowNameAlign = "l", matAlign = "r",
 prtHead = TRUE, prtEnd = TRUE, extraTitle = NULL,
 rowNameCols = 1, append = FALSE)
```

### Arguments

| | |
|---|---|
| `mat` | a matrix |
| `digits` | number of digits |
| `rowNameTitle` | title above row names |
| `file` | output file |
| `roundNum` | |
| `rowNameAlign` | alignment of row names |
| `matAlign` | |
| `prtHead` | |
| `prtEnd` | |
| `extraTitle` | |
| `rowNameCols` | |
| `append` | |

### Author(s)

Juerg Kindermann; code found on R list

## myclus *A clustering function*

### Description

Uses a hierarchical clustering to initiate a kmeans clustering.

### Usage

```
myclus(data = swiss, k = 3)
```

### Arguments

| | |
|---|---|
| `data` | The data matrix |
| `k` | the number of clusters |

### Value

a list from function kmeans

### Author(s)

From Ripley and Venables

### References

Venables, W.N. and Ripley, B.D (2000) *Modern Applied Statistics with S-PLUS*, Springer

### Examples

```
library(MASS)
data(swiss)
cl <- myclus(data = swiss, k = 3)
gap(swiss,cl$cluster)
```

## normalise *Normalise arrays*

### Description

Normalises arrays against a calculated average array, and calibrated linearly in a cube-root scatter plot.

### Usage

```
normalise(x,linear=TRUE)
```

### Arguments

| | |
|---|---|
| `x` | The data matrix |
| `linear` | if linear=TRUE then the matrix elements are raised to the power of 3. |

### Value

normalised version of indata

### Author(s)

Per Broberg

### References

Tusher, V.G., Tibshirani, R., and Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *PNAS* Vol. 98, no.9, pp. 5116-5121

---

## `one.probeset.per.gene` *Select the best probeset per gene*

---

### Description

This function takes a vector of probeset identifiers, a vector of gene identifiers and a vector of present rates, and outputs the probeset id per gene that corresponds to the highest present rate.

### Usage

```
one.probeset.per.gene(probeset = probeset, present = present, symbol = symbol)
```

### Arguments

| | |
|---|---|
| `probeset` | a vector of probeset id's |
| `present` | a vector of present rates |
| `symbol` | a vector of gene symbols |

### Details

It is assumed that missing gene symbol is coded as "". Note also that other measurements than present rate may be useful as selection criterion, such some variation measure. The function only assumes that high values are desirable.

### Value

A vector of probeset id's.

### Note

Experimental function. Feedback appreciated.

### Author(s)

Per Broberg

outlier *Identify outliers in the multivariate distribution*

## Description

A PCA model is fitted to data and two statistics as measures of extremity are calculated. These are the Hotelling t-square and DMODX, the first is a measure of how far away from the centre of the projection subspace the projection of the observation is. The second one measures how remote from the projection the actual observation is. SVD is done directly on the data matrix. The number of significant dimensions is defined as the number of eigenvalues greater than 1. Typically arrays are in different columns.

## Usage

```
outlier(M)
```

## Arguments

| | |
|---|---|
| M | matrix |

## Value

Dataframe with columns Hotelling and DMODX

## Author(s)

Per Broberg

## References

Jackson, J.E. (1991) *A User's Guide to Principal Components*. Wiley

## Examples

```
## Not run:
# not run
ut<-outlier(M)
#[1] "The number of significant dimensions is 19"
colnames(ut)
#[1] "Hotelling" "DMODX"

## End(Not run)
```

## `p0.mom` *Estimate proportion unchanged genes*

### Description

The function uses the vector of p-values to estimate p0.

### Usage

```
p0.mom(ps = pvalues)
```

### Arguments

| | |
|---|---|
| `ps` | the vector of p-values, e.g. from firstpass |

### Value

the value of p0, the proportion unchanged genes as a list with components

| | |
|---|---|
| `mgf` | estimate from the mgf method |
| `PRE` | estimate from the PRE method |
| `experimental1` | |
| `experimental2` | |

### Author(s)

Per Broberg

### References

Broberg, P. A new estimate of the proportion unchanged genes, 2005, *Genome Biology* 5:p10
Broberg, P. A comparative review of estimates of the proportion unchanged genes and the false discovery rate, submitted (2004)

## `pava.fdr` *Estimate of the FDR and the proportion unchanged genes*

### Description

Estimates tail area and local false discovery rate using isotonic regression

### Usage

```
pava.fdr(ps = pvalues, p0 = NULL)
```

### Arguments

| | |
|---|---|
| `ps` | the vector of p-values, e.g. from firstpass |
| `p0` | an estimate of the proportion unchanged genes |

### Details

If p0 = NULL the PRE estimate of p0 is calculated.

### Value

a list with components

| | |
|---|---|
| `pava.fdr` | estimate of the FDR |
| `p0` | estimate of p0 |
| `pava.local.fdr` | estimate of the local fdr |

### Author(s)

Per Broberg

### References

Broberg, P : A comparative review of estimates of the proportion unchanged genes and the false discovery rate, *BMC Bioinformatics* 2005, 5(1):199
Aubert J, Bar-Hen A, Daudin J-J, Robin S: Determination of the differentially expressed genes in microarray experiments using local FDR. *BMC Bioinformatics* 2004, 6(1):125

---

## `pava` *Pooling of Adjacent Violators*

---

### Description

The PAVA algorithm

### Usage

```
pava(x, wt = rep(1, length(x)))
```

### Arguments

| | |
|---|---|
| `x` | A numeric sequence |
| `wt` | observation weights; 1 by default. |

### Details

The algorithm will turn a non-increasing into a non-decreasing one. pava is an internal function used to force monotonicity, e.g. of p1 in function Zfreq

### Value

A non-decreasing sequence

### Author(s)

R.F. Raubertas, code from S list

## Examples

```
pava(c(1,2,4,3,5))
# [1] 1.0 2.0 3.5 3.5 5.0
```

---

R2BASE *Produces a BASE file*

---

## Description

The function produces a BASE file for import to Gene Data Viewer.

## Usage

```
R2BASE(context.data = clingen, sample.ids = AZID, expression.data = dats,
annotation = annots, out = "u:/temp/temp.base")
```

## Arguments

`context.data` e.g. a clinical database

`sample.ids` Sample Ids, that names the columns of the expression data.

`expression.data` a matrix with the gene expression data, samples correspond to columns and probesets to rows. It is assumed that probeset identifiers are found in the first column.

`annotation` annotations of the probesets, i.e. the rows in the expression.data. It is assumed that probeset identifiers are found in the first column.

`out` the output file including path

## Value

The file produced complies with an old BASE format. However, none of these formats are documented , as far as I know. So, essentially this function defines a data format that can be read by e.g. Gene Data Viewer.

## Author(s)

Per Broberg

## R2mim *Output a script file to WinMIM*

### Description

Given a candidate probe, it produces a script file for WinMIM by calculating means and covariances and for the N most highly correlated probes (in absolute value). Here N is an input parameter, but a recommended value 10. WinMIM can find a relevant graphical model for the dependencies between the probes.

### Usage

```
R2mim(probe="12345_at", N=10, data=inm, out="u:/study/copd/mimscr.txt")
```

### Arguments

| | |
|---|---|
| `probe` | The name of the candidate probe |
| `N` | The number of highly correlated probes to be studied |
| `data` | The input data set |
| `out` | The MIM script file |

### Value

The correlation matrix

### Note

David Edwards' program WinMIM can be found on StatLib (`http://lib.stat.cmu.edu/graphmod/`). In MIM issue `input mimscr.txt` and the calculations to find a model will start. When finished go to the Graphics menu and click on `Independence Graph`. The resulting graph can be exported both to WMF and LaTeX.

### Author(s)

Per Broberg

### References

Edwards, David (1995) *Introduction to Graphical Modelling*. Springer-Verlag
Lauritzen, Steffen (1996) *Graphical Models*. Oxford University Press
Whittaker, Joe (1990) *Graphical Models in Multivariate Analysis*. Wiley

## `rank.genes` *Rank genes with respect to multiple criteria*

### Description

It is assumed that genes come in rows and the criteria in columns. Furthermore, high values should be good. After ranking the genes with respect to each criterion, the function does a PCA on the ranks, uses the firsta PC to obtain the final ranks. In principle it could happen that genes are ranked in the opposite direction to the one intended, but that should be evident from a quick glance at the results.

### Usage

```
rank.genes(data = indats)
```

### Arguments

| | |
|---|---|
| `data` | A matrix with the criteria in columns. |

### Value

The total ranks of the genes.

### Author(s)

Per Broberg

## `rank.trend` *Trend analysis based on ranks*

### Description

Ranks are used to score genes with respect to degree of agreement to a given trend or pattern, Lehmann (1974) p.294.

### Usage

```
rank.trend(data = x, pattern = c(1:ncol(data)), har = FALSE)
```

### Arguments

| | |
|---|---|
| `data` | A data frame with one array in each column |
| `pattern` | A permutation of the integers 1:ncol(data) |
| `har` | logical parameter indicating whether or not a score based on Hardy's theorem shall be calculated. |

### Details

The rank scores gives a higher weight to a deviation from trend in more distant obseveations than a deviation between neighbouring observations. The p-values are calculated through a normal approximation.

### Value

A list with the components

| | |
|---|---|
| `score` | the rank score for each gene |
| `hardy` | if har = TRUE the hardy score, NULL otherwise |
| `pvals` | the p-values for the null hypothesis of no trend |

### Author(s)

Per Broberg

### References

Lehmann, E.L. (1975) Nonparametrics: Statistical Methods Based on Ranks, Holden-Day

### Examples

```
# not run
D <- c(123, 334, 578, 762, 755, 890)
rank.trend(data = t(as.matrix(D)), har = TRUE)
#     Trend score Hardy score p-value for no trend
# [1,]           2          90           0.01750284
```

## `rsd.test` *Compare two groups with respect to their RSD (CV)*

### Description

A by row comparison of the Relative Standard Deviation (RSD), asa Coefficient of Variation (CV), is done using a bootstrap

### Usage

```
rsd.test(data1 = x, data2= y, B = NULL)
```

### Arguments

| | |
|---|---|
| `data1` | A matrix with the samples for group 1 in columns. |
| `data2` | A matrix with the samples for group 2 in columns. |
| `B` | the number of bootstrap iterations. If NULL no bootstrap is performed. |

### Value

A list with the components

| | |
|---|---|
| `cv1` | A vector of the RSD's for sample 1 |
| `cv2` | A vector of the RSD's for sample 2 |
| `t.stat` | the test statistic |
| `p.vals` | A vector of p-values for the comparison between $cv1$ and $cv2$ |

### Author(s)

Per Broberg

### References

Broberg P, Estimation of Relative Standard Deviation,(1999) in *Drug Development and Industrial Pharmacy*, Vol 25 no 1 37-43

## `samrocNboot` *Calculate ROC curve based SAM statistic*

### Description

A c-code version of samrocN. Calculation of the regularised t-statistic which minimises the false positive and false negative rates.

### Usage

```
samrocNboot(data=M,formula=~as.factor(g), contrast=c(0,1), N = c(50, 100, 200, 3
smooth=FALSE, w = 1, measure = "euclid", probeset = NULL)
```

### Arguments

| | |
|---|---|
| `data` | The data matrix |
| `formula` | a linear model formula |
| `contrast` | the contrast to be estimnated |
| `N` | the size of top lists under consideration |
| `B` | the number of bootstrap iterations |
| `perc` | the largest eligible percentile of SE to be used as fudge factor |
| `smooth` | if TRUE, the std will be estimated as a smooth function of expression level |
| `w` | the relative weight of false positives |
| `measure` | the goodness criterion |
| `probeset` | probeset ids;if NULL then "probeset 1", "probeset 2", ... are used. |

### Details

The test statistic is based on the one in Tusher et al (2001):

$$\frac{d = diff}{s_0 + s}$$

where $diff$ is a the estimate of a constrast, $s_0$ is the regularizing constant and $s$ the standard error. At the heart of the method lies an estimate of the false negative and false positive rates. The test is calibrated so that these are minimised. For calculation of $p$-values a bootstrap procedure is invoked. Further details are given in Broberg (2003).

The p-values are calculated through permuting the rows of the design matrix. NB This is not adequate for all linear models.

samrocNboot uses C-code to speed up the bootstrap loop.

### Value

An object of class samroc.result.

### Author(s)

Per Broberg and Freja Vamborg

### References

Tusher, V.G., Tibshirani, R., and Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *PNAS* Vol. 98, no.9, pp. 5116-5121

Broberg, P. (2002) Ranking genes with respect to differential expression , http://genomebiology.com/2002/3/9/preprint/0007

Broberg. P: Statistical methods for ranking differentially expressed genes. Genome Biology 2003, 4:R41 http://genomebiology.com/2003/4/6/R41

### Examples

```
library(multtest)
#Loading required package: genefilter
#Loading required package: survival
#Loading required package: splines
#Loading required package: reposTools
data(golub)
 # This makes the expression data from Golub et al available
 # in the matrix golub, and the sample labels in the vector golub.cl
set.seed(849867)
samroc.res <- samrocNboot(data = golub, formula = ~as.factor(golub.cl))
# The proportion of unchanged genes is estimated at
samroc.res@p0
# The fudge factor equals
 samroc.res@s0
# A histogram of p-values
 hist(samroc.res@pvalues)
 # many genes appear changed
```

## samrocN *Calculate ROC curve based SAM statistic*

### Description

Calculation of the regularised t-statistic which minimises the false positive and false negative rates.

### Usage

```
samrocN(data=M,formula=~as.factor(g), contrast=c(0,1), N = c(50, 100, 200, 300),
 smooth = FALSE, w = 1, measure = "euclid", p0 = NULL, probeset = NULL)
```

## Arguments

| | |
|---|---|
| `data` | The data matrix, or ExpressionSet |
| `formula` | a linear model formula |
| `contrast` | the contrast to be estimnated |
| `N` | the size of top lists under consideration |
| `B` | the number of bootstrap iterations |
| `perc` | the largest eligible percentile of SE to be used as fudge factor |
| `smooth` | if TRUE, the std will be estimated as a smooth function of expression level |
| `w` | the relative weight of false positives |
| `measure` | the goodness criterion |
| `p0` | the proportion unchanged probesets; if NULL p0 will be estimated |
| `probeset` | probeset ids;if NULL then "probeset 1", "probeset 2", ... are used. |

## Details

The test statistic is based on the one in Tusher et al (2001):

$$\frac{d = diff}{s_0 + s}$$

where $diff$ is a the estimate of a constrast, $s_0$ is the regularizing constant and $s$ the standard error. At the heart of the method lies an estimate of the false negative and false positive rates. The test is calibrated so that these are minimised. For calculation of $p$-values a bootstrap procedure is invoked. Further details are given in Broberg (2003). Note that the definition of p-values follows that in Davison and Hinkley (1997), in order to avoid p-values that equal zero.

The p-values are calculated through permuting the residuals obtained from the null model, assuming that this corresponds to the full model except for the parameter being tested, coresponding to the contrast coefficient not equal to zero. This means that factors not tested are kept fixed. NB This may be adequate for testing a factor with two levels or a regression coefficient (correlation), but it is not adequate for all linear models.

## Value

An object of class samroc.result.

## Author(s)

Per Broberg

## References

Tusher, V.G., Tibshirani, R., and Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *PNAS* Vol. 98, no.9, pp. 5116-5121

Broberg, P. (2002) Ranking genes with respect to differential expression , `http://genomebiology.com/2002/3/9/preprint/0007`

Broberg. P: Statistical methods for ranking differentially expressed genes. Genome Biology 2003, 4:R41 `http://genomebiology.com/2003/4/6/R41`

Davison A.C. and Hinkley D.V. (1997) Bootstrap Methods and Their Application. Cambridge University Press

## samroc.result-class *Class "samroc.result" for results of the function samrocN*

### Description

The class samroc.result is the output of a call to `samrocN` and the input of various other functions.

### Slots

`d`: Object of class "numeric". Observed test statistic.

`diff`: Object of class "numeric". Estimate of effect, e.g. difference between group means.

`se`: Object of class "numeric". Standard error of `diff`.

`d0`: Object of class "matrix". Permutation test statistics.

`p0`: Object of class "numeric". The estimated proportion unaffceted genes.

`s0`: Object of class "numeric". The fudge factor.

`pvalues`: Object of class "numeric". The p-values.

`N.list`: Object of class "integer". The optimal top list size among the sizes suggested.

`errors`: Object of class "numeric". The sum of false postives and false negatives given a list that includes the current gene.

`formula`: Obeject of class "formula". The linear model formula used.

`contrast`: Object of class "numeric". The contrast estimated.

`annotation`: Object of class "character". Annotation or comments regarding the analysis. By default the date.

`N.sample`: Object of class "integer". The number of samples.

`B`: Object of class "integer". The number of premutations.

`call`: Object of class "character". The call to the function.

`id`: Object of class "character". The probeset ids.

`error.df`: Object of class "integer". The error degrees of freedom.

`design`: Object of class "matrix". The design matrix.

### Methods

**show** (samroc.result): Summarizes the test result.

**plot** (samroc.result): Plots the density of the observed test statistic and that of the corresponding null distribution

### Author(s)

Per Broberg

### See Also

`samrocN`

## `union.of.pways` *Create the union of two pathway lists*

### Description

This function takes two lists where each component is a vector of probe sets ids and create a new such list that contains all probe sets and pathways from the two lists.

### Usage

```
union.of.pways(x,y)
```

### Arguments

| | |
|---|---|
| `x` | the first list |
| `y` | the second list |

### Details

The function *merge.list* in package *RCurl* forms a basis for this function which adds the ability to add new probe sets to existing pathways.

### Value

A list which is the union of the two input lists.

### Note

Experimental function. Feedback appreciated.

### Author(s)

Per Broberg

### Examples

```
X = list(a=c(1,2),c=c(1,2));Y = list(a=c(3,4),d=c(12,2))
union.of.pways(X,Y)
```

## Xprep *Fitting of a linear model*

### Description

The function fits a linear model to a microarray data matrix.

### Usage

```
Xprep(indata=M, formula=~as.factor(g), contrast=c(0,1), design=NULL)
```

### Arguments

| | |
|---|---|
| `indata` | The data matrix |
| `formula` | a linear model formula in the lm format |
| `contrast` | a vector defining the contrast of interest |
| `design` | the design matrix |

### Value

a list with the entries

| | |
|---|---|
| `Mbar` | estimate of the contrast |
| `Vest` | the error variance |
| `k` | inverse of the scale factor turning Vest into a standard error |
| `f` | the degrees of freedom of Vest |
| `design` | the design matrix |

### Author(s)

Per Broberg

## Xprep.resid *Calculation of input of residuals from linear model*

### Description

The function fits a linear model to a microarray data matrix and calculates the residuals.

### Usage

```
Xprep.resid(data=M, formula=~as.factor(g), design=NULL)
```

### Arguments

| | |
|---|---|
| `data` | The data matrix |
| `formula` | a linear model formula in the lm format |
| `design` | the design matrix |

## Value

A matrix with the residuals

## Author(s)

Per Broberg

# Index