GeneMeta

March 24, 2012

`CountPlot` *Plots for Meta-analysis of gene expression data.*

Description

Plots for meta-analysis

Usage

```
IDRplot(m,CombineExp=1:(length(grep("zSco_Ex",colnames(m)))),colPos="black",colN
CountPlot(kkk,cols,Score=c("FDR","zSco"),kindof=c("two.sided","pos","neg"),type=
```

Arguments

`m`	result matrix of the function `zScores`
`type`	plot parameter
`ylab`	plot parameter
`xlab`	plot parameter
`pch`	plot parameter
`colPos`	color for positive z scores
`colNeg`	color for negative z scores
`pchPos`	symbol for positive z scores
`pchNeg`	symbol for negative z scores
`CombineExp`	vector of integer- which experiments should be combined-default:all experiments
`kkk`	result object of function zScoreFDR
`cols`	vector of cols, one for each experiment, and one for the combination
`Score`	should the FDR or the zScore be plotted
`kindof`	"pos", "neg" or "two.sided"
`...`	additional plot parameter

Details

IDRplot produces a plot described in Choi et al.

Author(s)

M.Ruschhaupt

References

Choi et al, Combining multiple microarray studies and modeling interstudy variation. Bioinformatics, 2003, i84-i90.

`Nevins` *Intensity data for 46 Affymetrix slides with tissue samples of breast tumors*

Description

Intensity data for 46 Affymetrix hu6800 slides with tissue samples of breast tumors. See vignette Nevins.pdf in the /scripts directory for details of the processing.

Usage

```
data(Nevins)
```

Format

`Nevins` is an `ExpressionSet` containing the data from 46 Affymetrix chips.

Source

http://data.cgt.duke.edu/west.php

References

Predicting the clinical status of human breast cancer by using gene expression profiles, West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA Jr, Marks JR, and Nevins JR. Proc Natl Acad Sci U S A 98(20):11462-7 (2001)

Examples

```
data(Nevins)
Nevins
```

dstar *Tools for Meta-analysis of gene expression data.*

Description

A small number of meta-analysis functions for comparing two gene expression experiments are provided.

Usage

```
dstar(d, n)
getdF(data, categ)
sigmad(d, ng1, ng2)
```

Arguments

d	A vector of t-statistics, i.e. the output of getdF.
n	The number of t-statistics.
data	The data used to compute t-statistics, either a matrix or an ExpressionSet.
categ	A vector of 0's and 1's indicating group membership.
ng1	The number of samples in group 1.
ng2	The number of samples in group 2.

Details

The functions getdF compute t-test statistics for the input data and group membership (note that group membership must be indicated by a vector of 0's and 1's).

The function dstar computes an unbiased estimate of the t-test. The function sigmad computes the variance estimate of dstar.

Value

The different functions have different return values, but generally they are vectors of the requested quantities.

Author(s)

L. Lusa, R. Gray and R. Gentleman

References

Choi et al, Combining multiple microarray studies and modeling interstudy variation. Bioinformatics, 2003, i84-i90.

Examples

```
x = matrix(rnorm(1000), ncol=10)
ds = getdF(x, rep(c(0,1), c(5,5)))
dst = dstar(ds, ncol(x))
sgd = sigmad(ds, 5, 5)
```

f.Q *Compute Cochran's Q statistic*

Description

Compute Cochran's Q statistic for testing whether the a fixed effects or a random effects model will be appropriate.

Usage

```
f.Q(dadj, varadj)
```

Arguments

dadj	A matrix, each row is a gene, each column a study, of the estimated t-statistics.
varadj	A matrix, each row is a gene, each column a study, of the estimated, adjusted variances of the t-statistics.

Details

A straightforward computation of Cochran's Q statistic. If the null hypothesis that the data are well modeled by a fixed effects design is true then the estimate Q values will have approximately a chi-squared distribution with degrees of freedom equal to the number of studies minus one.

Value

A vector of length equal to the number of rows of dadj with the Q statistics.

Author(s)

L. Lusa and R. Gentleman

References

Choi et al, Combining multiple microarray studies and modeling interstudy variation. Bioinformatics, 2003, i84-i90.

See Also

dstar,sigmad

Examples

```
##none now, this requires a pretty elaborate example
```

tau2.DL *estimating my and tau in a REM*

Description

tau2.DL is an estimation of tau in a random effects model (REM) using Cochran's Q statistic.

Usage

```
tau2.DL(Q, num.studies, my.weights)
mu.tau2(my.d, my.vars.new)
var.tau2(my.vars.new)
```

Arguments

Q	A vector of Cochran's Q statistics.
num.studies	The number of studies used for the meta-analysis.
my.weights	A matrix with one column for each experiment containing the variances of the effects that should be combined.
my.d	A matrix, with one column for each experiment, containing the effects that should be combined.
my.vars.new	A matrix, with one column for each experiment, containing the variances of the effects that should be combined.

Author(s)

L. Lusa and R. Gentleman

References

Choi et al, Combining multiple microarray studies and modeling interstudy variation. Bioinformatics, 2003, i84-i90.

See Also

dstar,sigmad

Examples

```
# please have a look at the vignette
```

zScores *Tools for Meta-analysis of gene expression data.*

Description

A small number of meta-analysis functions for computing zScores for FEM and REM and computing FDR.

Usage

```
zScores(esets, classes, useREM=TRUE, CombineExp=1:length(esets))
zScorePermuted(esets, classes, useREM=TRUE, CombineExp=1:length(esets))
zScoreFDR(esets, classes, useREM=TRUE, nperm=1000, CombineExp=1:length(esets))
multExpFDR(theScores, thePermScores, type="pos")
```

Arguments

esets	A list of ExpressionSets, one expression set per experiment. All experiments must have the same variables(genes).
classes	A list of class memberships, one per experiment. Each list can only contain 2 levels.
useREM	A logical value indicating whether or not to use a REM, TRUE, or a FEM, FALSE, for combining the z scores.
theScores	A vector of scores (e.g. t-statistics or z scores)
thePermScores	A vector of permuted scores (e.g. t-statistics or z scores)
type	"pos", "neg" or "two.sided"
nperm	number of permutations to calculate the FDR
CombineExp	vector of integer- which experiments should be combined-default:all experiments

Details

The function zScores implements the approach of Choi et al. for for a set of ExpressionSets. The function zScorePermuted applies zScore to a single permutation of the class labels. The function zScoreFDR computes a FDR for each gene, both for each single experiment and for the combined experiment. The FDR is calculated as described in Choi et al. Up to now ties in the zscores are not taken into account in the calculation. The function might produce incorrect results in that case. The function also computes zScores, both for the combines experiment and for each single experiment.

Value

A matrix with one row for each probe(set) and the following columns:

zSco_Ex_	For each single experiment the standardized mean difference, Effect_Ex_, divided by the estimated standard deviation, the square root of the EffectVar_Ex_ column.
MUvals	The combined standardized mean difference (using a FEM or REM)

MUsds	The standard deviation of the MUvals.
zSco	The z statistic - the MUvals divided by their standard deviations, MUsds.
Qvals	Cochran's Q statistic for each gene.
df	The degree of freedom for the Chi-square distribution. This is equal to the number of combined experiments minus one.
Qpvalues	The probability that a Chi-square random variable, with df degrees of freedom) has a higher value than the value from the Q statistic.
Chisq	The probability that a Chi-square random variate (with 1 degree of freedom) has a higher value than the value of $zSco^2$.
Effect_Ex_	The standardized mean difference for each single experiment.
EffectVar_Ex_	The variance of the standardized mean difference for each single experiment.

Note that the three column names that end in an underscore are replicated, once for each experiment that is being analyzed.

Author(s)

M. Ruschhaupt

References

Choi et al, Combining multiple microarray studies and modeling interstudy variation. Bioinformatics, 2003, i84-i90.

Examples

```
data(Nevins)

##Splitting
thestatus  <- Nevins$ER.status
group1     <- which(thestatus=="pos")
group2     <- which(thestatus=="neg")
rrr        <- c(sample(group1, floor(length(group1)/2)),
                sample(group2,ceiling(length(group2)/2)))
Split1     <- Nevins[,rrr]
Split2     <- Nevins[,-rrr]

#obtain classes
Split1.ER <- as.numeric(Split1$ER.status) - 1
Split2.ER <-as.numeric(Split2$ER.status) - 1

esets     <- list(Split1,Split2)
classes   <- list(Split1.ER,Split2.ER)
theScores <- zScores(esets,classes,useREM=FALSE)
theScores[1:2,]
```

Index