pcot2

April 19, 2009

`aveProbe` *Transform Affymetrix data so that unique genes with multiple probes are represented by a single expression value on each array.*

Description

In Affymetrix gene expression data, a unique gene can often link to multiple probe sets, with such genes then having a greater influence on the analysis (particularly if the gene is differentially expressed). To overcome this problem the median is taken across all probes sets which represent a unique gene.

Usage

```
aveProbe(x, imat = NULL, ids)
```

Arguments

`x`	A matrix with no missing values; Each row represents a gene and each column represents a sample.
`imat`	A matrix indicating presence or absence of genes in the gene sets. The indicator matrix contains rows representing gene identifiers of genes present in the expression data and columns representing group (gene set) names.
`ids`	A vector of identifiers (e.g., UniGene or LocusLink identifiers) representing unique genes which match to the probe ids in the expression data.

Value

`newx`	A data matrix with rows representing the input identifiers and columns representing samples.
`newimat`	A new imat (indicator matrix) with rows representing the unique gene identifiers and columns representing gene sets.

Author(s)

Sarah Song and Mik Black

See Also

`pcot2`,`corplot`,`corplot2`

Examples

```
library(multtest)
library(hu6800.db)
data(golub)
rownames(golub) <- golub.gnames[,3]
colnames(golub) <- golub.cl
KEGG.list <- as.list(hu6800PATH)
imat <- getImat(golub, KEGG.list, ms=10)
colnames(imat) <- paste("KEGG", colnames(imat), sep="")

pathlist <- as.list(hu6800PATH)
pathlist <- pathlist[match(rownames(golub), names(pathlist))]
ids <- unlist(mget(names(pathlist), env=hu6800SYMBOL))
#### transform data matrix only ####
newdat <- aveProbe(x=golub, ids=ids)$newx
#### transform both data and imat ####
output <- aveProbe(x=golub, imat=imat, ids=ids)
newdat <- output$newx
newimat <- output$newimat
newimat <- newimat[,apply(newimat, 2, sum)>=10]
```

`corplot` *Produce a plot for jointly visualizing pooled correlation information and expression data for selected genes*

Description

This plot is used for looking at pooled inter-gene correlation within a pre-defined group of genes, in conjunction with information about differences in expression activity between classes.

Usage

```
corplot(x, sel, cla = NULL, inputP = NULL, main, gene.locator = FALSE, add.name
```

Arguments

`x`	A matrix with no missing values; Each row represents a gene and each column represents a sample.
`sel`	A vector of selected gene identifiers.
`cla`	Class labels representing two distinct experimental conditions (e.g., normal and disease).
`inputP`	This option allows users to input p-values for each gene (e.g., if produced by another software package).
`main`	A title for the plot.
`gene.locator`	This option allows users to click of the plot to identify groups of genes. Clicking twice on the diagonal of the plot returns the identifiers of genes between the points clicked.
`add.name`	Specifies whether gene identifiers should be printed on the plot.
`font.size`	Adjusts the size of gene names printed on the plot.

`dist.method` Specifies the method for calculating inter-gene distance (used when ordering the rows and columns of the correlation plot). The available distance methods are "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson","correlation" or "spearman". For additional details see the `amap` package and the help documentation for the `Dist` function.

Author(s)

Sarah Song and Mik Black

See Also

`pcot2`,`corplot2`,`aveProbe`

Examples

```
library(multtest)
library(hu6800.db)
data(golub)
rownames(golub) <- golub.gnames[,3]
colnames(golub) <- golub.cl
KEGG.list <- as.list(hu6800PATH)
imat <- getImat(golub, KEGG.list, ms=10)
colnames(imat) <- paste("KEGG", colnames(imat), sep="")
sel <- c("04620","04120")
main <- paste("KEGG", sel, sep="")
for(i in 1:length(sel)){
fname <- paste("corplot-KEGG",sel[i] , ".jpg", sep="")
jpeg(fname, width=1600, height=1200, quality=100)
selgene <- rownames(imat)[imat[,match(paste("KEGG",sel,sep="")[i],colnames(imat))]==1]
corplot(golub, selgene, golub.cl, main=main[i])
dev.off()
}
```

---

corplot2 *Produce a plot for jointly visualizing unpooled correlation information and expression data for selected genes*

Description

This plot is used for looking at unpooled inter-gene correlation within a pre-defined group of genes, in conjunction with information about differences in expression activity between classes.

Usage

```
corplot2(x, sel, cla = NULL, inputP = NULL, main, gene.locator = FALSE, add.name
```

Arguments

`x` A matrix with no missing values; Each row represents a gene and each column represents a sample.

`sel` A vector of selected gene identifiers.

cla	Class labels representing two distinct experimental conditions (e.g., normal and disease).
inputP	This option allows users to input p-values for each gene (e.g., if produced by another software package).
main	A title for the plot.
gene.locator	This option allows users to click of the plot to identify groups of genes. Clicking twice on the diagonal of the plot returns the identifiers of genes between the points clicked.
add.name	Specifies whether gene identifiers should be printed on the plot.
font.size	Adjusts the size of gene names printed on the plot.
dist.method	Specifies the method for calculating inter-gene distance (used when ordering the rows and columns of the correlation plot). The available distance methods are "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson","correlation" or "spearman". For additional details see the amap package and the help documentation for the Dist function.

Author(s)

Sarah Song and Mik Black

See Also

pcot2,corplot,aveProbe

Examples

```
library(multtest)
library(hu6800.db)
data(golub)
rownames(golub) <- golub.gnames[,3]
colnames(golub) <- golub.cl
KEGG.list <- as.list(hu6800PATH)
imat <- getImat(golub, KEGG.list, ms=10)
colnames(imat) <- paste("KEGG", colnames(imat), sep="")
sel <- c("04620","04120")
pvalue <- c(0.001, 0.72)
library(KEGG.db)
pname <- unlist(mget(sel, env=KEGGPATHID2NAME))
main <- paste("KEGG", sel, ": ", pname, ": ", "P=", pvalue, sep="")
for(i in 1:length(sel)){
fname <- paste("corplot2-KEGG",sel[i] , ".jpg", sep="")
jpeg(fname, width=1600, height=1200, quality=100)
selgene <- rownames(imat)[imat[,match(paste("KEGG",sel,sep="")[i],colnames(imat))]==1]
corplot2(golub, selgene, golub.cl, main=main[i])
dev.off()
}
```

getImat *Generate an indicator matrix*

Description

This function is used to generate an indicator matrix as an input to the pcot2 function. The gene category indicator matrix indicates presence or absence of genes in pre-defined gene sets (e.g., gene pathways). The indicator matrix contains rows representing gene identifiers of genes present in the expression data and columns representing pre-defined group names. A value of 1 indicates the presence of a gene and 0 indicates the absence for the gene in a particular group.

Usage

```
getImat(x, pathlist, ms = 10)
```

Arguments

x	A matrix with no missing values; Each row represents a gene and each column represents a sample.
pathlist	A list of gene sets.
ms	The minimum gene set size. Gene sets containing less than this number of genes will be excluded from the analysis.

Value

An indicator matrix is returned. The matrix value is 1 (gene in) or 0 (gene out)

Author(s)

Sarah Song and Mik Black

See Also

pcot2,corplot,corplot2,aveProbe

Examples

```
library(multtest)
library(hu6800.db)
data(golub)
rownames(golub) <- golub.gnames[,3]
colnames(golub) <- golub.cl
KEGG.list <- as.list(hu6800PATH)
imat <- getImat(golub, KEGG.list, ms=10)
```

pcot2 *Principal Coordinates and Hotelling's T-Square*

Description

The pcot2 function implements the PCOT2 testing method, which is a two-stage permutation-based approach for testing changes in activity in pre-specified gene sets.

Usage

```
pcot2(emat, class = NULL, imat, permu = "ByColumn", iter = 1000, alpha = 0.05, a
```

Arguments

emat	A gene expression matrix with no missing values; Each row represents a gene and each column represents a sample.
class	Class labels representing two distinct experimental conditions (e.g., normal and disease).
imat	The gene category indicator matrix indicates presence or absence of genes in pre-defined gene sets (e.g., gene pathways). The indicator matrix contains rows representing gene identifiers of genes present in the expression data and columns representing pre-defined group names. A value of 1 indicates the presence of a gene and 0 indicates the absence for the gene in a particular group.
permu	Specifies whether genes or samples are permuted. By default, permutations are performed by sample ("ByColumn").
iter	The number indicates how many permutations will be performed in the analysis.
alpha	alpha determines the significance threshold for the permutation p-values.
adjP.method	Specifies that p-values be adjusted by one of the following methods: "bonferroni", "holm", "hochberg", "hommel", "BH" (Benjamini and Hochberg), or "BY" (Benjamini and Yekutieli).
var.equal	Specifies the use of either a pooled estimate of correlation for the two classes or an unpooled estimate for calculating each T-squared statistic. By default, the pooled estimate is used.
ncomp	The dimensionality to which the data matrix is reduced via principal coordinates. The default dimensionality is set as ncomp=2.
dist.method	Specifies the method for calculating distance in the PCO procedure. The available distance methods are "euclidean", "maximum", "manhattan", "canberra", "binary", "pearson","correlation" or "spearman". For additional details see the amap package and the help documentation for the Dist function.

Details

The raw permutation p-values are adjusted for multiple testing by a call to 'p.adjust'.

Value

res.all	A data frame which prints information for all pathways
res.sig	A data frame which prints information for significant pathways at a given alpha level
comparison	Print the contrast used in the analysis
...	

Author(s)

Sarah Song and Mik Black

See Also

corplot,corplot2,aveProbe

Examples

```
ns <- 40  ## 40 samples
cla <- rep(c("Trt","Ctr"),each=ns/2)
ngene <- 10  ## 10 genes per group
npath <- 10  ## 10 groups

nreal <- 3  ## alter groups ##
nnull <- npath-nreal   ## null groups ##
pname <- c(paste("RealP",1:nreal, sep=""), paste("NullP",1:nnull, sep=""))

## Three main inputs in the function ##
## [1] Simulate (gene) expression matrix (emat) ##
rmv <- function(mn, covm, nr, nc){
   sigma <- diag(nr)
   sigma[sigma==0] <- covm
   x1 <- rmvnorm(nc/2, mean=mn, sigma=sigma)
   x0 <- rmvnorm(nc/2, mean=rep(0,nr), sigma=sigma)
   mat <- t(rbind(x1,x0))
  return(mat)
}

covm <- 0.9  ##covariance
ct <- c(6,8,10)  ##mean

library(mvtnorm)
emat <- c()
for (i in 1:nreal) emat <- rbind(emat, rmv(rep(ct[i],ngene),covm=covm, ngene, ns))  # for
for (i in 1:(npath-nreal)) emat <- rbind(emat, rmv(mn=rep(0,ngene),covm=covm, nr=ngene, n
dimnames(emat) <- list(paste("Gene", 1:(ngene*npath),sep=""), cla)

## [2] class label ##
cla

## [3] indicator matrix (row: genes and col: pathways)
imat <- kronecker(diag(npath),rep(1,ngene))
dimnames(imat) <- list(paste("Gene",1:(ngene*npath), sep=""), pname)

results.pcot2 <- pcot2(emat, cla, imat)
results.pcot2$res.sig
```

```
results.pcot2$res.all
```

Index