MergeMaid

April 20, 2011

`mergeCoeff` *Class mergeCoeff, a class for storing regression coefficients.*

Description

This is the class representation for output from regression coefficient calculations

Slots

coeff Object of class matrix, A matrix storing default coefficients.

coeff.std Object of class matrix, A matrix storing standardized coefficients.

zscore Object of class matrix, A matrix storing zscores.

Methods

Class-specific methods:

coeff (mergeCoeff) Accessor function for the coeff slot.

coeff<- (mergeCoeff) Replacement function for the coeff slot.

stdcoeff (mergeCoeff) Accessor function for the coeff.std slot.

stdcoeff<- (mergeCoeff) Replacement function for the coeff.std slot.

zscore (mergeCoeff) Accessor function for the zscore slot.

zscore<- (mergeCoeff) Replacement function for the zscore slot.

Standard generic methods:

plot (list) This method is not formally defined for mergeCoeff objects but for a matrix. This function would typically be called with the following syntax, plot(coeff(mergeCoeff)).The result is pairwise scatterplots of the columns of the selected matrix. If there are two studies, this is a single scatterplot.

See Also

`mergeExprs`,`modelOutcome`, `mergeExpressionSet-class`

Examples

```
if(require(Biobase) & require(MASS) & require(survival)){

data(mergeData)
merged  <- mergeExprs(sample1,sample2,sample3)

log.coeff  <- modelOutcome(merged,outcome=c(1,1,1),method="logistic")
plot(coeff(log.coeff))
plot(stdcoeff(log.coeff),pch=4,labels=c("study A","study B","study C"),col=3)

linear.coeff  <- modelOutcome(merged[1:2],outcome=c(3,3),method="linear")
plot(zscore(linear.coeff))
plot(zscore(linear.coeff),xlab="study A",ylab="study B",col=2)
}
```

`mergeCor` *Class mergeCor, a class for storing data relevant to integrative correlation coefficients.*

Description

This is the class representation for integrative correlation coefficients.

Details

If 'n' is the number of studies then for i < j <= n, the pairwise correlation of correlations for studies i and j is stored in column ((i-1)*(n-1)-(i-2)*(i-1)/2 + j-i) of the pairwise.cors slot.

Slots

pairwise.cors Object of class matrix. Each column contains correlation of correlation score for genes in a pair of study.

maxcors Object of class vector. Each slot represents maximal canonical correlation (pairwise canonical correlations) for each pair of studies.

notes Object of class vector. Each slot contains notes for each study.

Methods

Class-specific methods:

cors(mergeCor) Accessor function for the cors slot.

pairs.cors(mergeCor) Accessor function for the pairwise.cors slot

integrate.cors(mergeCor) Accessor function, returns integrative correlation (average pairwise correlation of correlations) for each gene. If adjust is TRUE, returns the integrate correlation devided by the correponding canonical correlation, and the default value is FALSE.

maxcors(mergeCor) Accessor function, returns maximal canonical correlation (pairwise canonical correlations) for each pair of studies.

Standard generic methods:

notes (mergeCor) An accessor function for the notes slot.

hist(mergeCor,...) Draw histograms of integrative correlation, here we use the approximate method to calculate the integrative correlation.

See Also

mergeCor-class,intCor,modelOutcome, mergeExpressionSet-class

Examples

```
if(require(Biobase) & require(MASS)){
data(mergeData)
merged  <- mergeExprs(sample1,sample2,sample3)
intcor3  <-intCor(merged,method="pearson")
plot(merged)
intcor2  <-intCor(merged[1:2],exact=FALSE)
plot(merged,pch=4,col=5)

pairwise.cors(intcor3)
integrative.cors(intcor3)
integrative.cors(intcor2)["Hs.12101"]
maxcors(intcor2)
}
```

---

mergeExpressionSet *Class mergeExpressionSet, a class for merged microarray data, and methods for processing them*

Description

This is class representation for merged Microarray Data.

Details

The mergeExpressionSet class is conceived as an extension of the ExpressionSet class provided in Biobase for the storage of expression array data. A mergeExpressionSet object is primarily a list of ExpressionSet objects, along with an incidence matrix indicating which genes appear in which studies. A mergeExpressionSet object with a single study reverts to the ExpressionSet class. A number of accessor functions are defiined for this class, as well as a few convenient analysis and plotting functions.

Slots

We assume there are K studies, representing a total of M unique genes.

data A list of ExpressionSet objects.

geneStudy Binary incidence matrix with M rows and K columns. Each column represents a study, and each row represents a gene. If study "s" contains gene "g", then geneStudy[g,s]=1, otherwise geneStudy[g,s]=0.

notes Object of class "character" This slot is available for storage of descriptive information.

Methods

Derived from ExpressionSet:

exprs(mergeExpressionSet) An accessor function for the data slot.

exprs<- (mergeExpressionSet) A replace function for the data slot

notes (mergeExpressionSet) An accessor function for the notes slot.

notes<- (mergeExpressionSet) A replace function for the notes slot.

geneNames (mergeExpressionSet) Accessor function for union of gene ids in all studies.

geneNames<- (mergeExpressionSet) A replace function for gene ids.

Class-specific methods:

geneStudy (mergeExpressionSet) Accessor function for the geneStudy slot.

phenoData (mergeExpressionSet) Accessor function for phenodata in ExpressionSet's. Returns a list, one phenodata matrix per study.

phenoData<- (mergeExpressionSet) A replace function for phenodata in ExpressionSet's. Returns a list, one phenodata matrix per study.

intersection (mergeExpressionSet) Represent data for genes common to all studies as a single ExpressionSet object.

modelOutcome (mergeExpressionSet) Calculate regression coefficients for each study/gene.

intCor (mergeExpressionSet) Calculate the integrative correlation coefficients for mergeExpressionSet data.

intcorDens (mergeExpressionSet) Plot the distribution of the integrative correlation coefficients and the null distribution obtained by permutation here we use the approximate method to calculate the integrative correlation.

Standard generic methods:

length (mergeExpressionSet) Function returning the number of studies in the mergeExpressionSet.

names (mergeExpressionSet) Function returning study names.

names<- (mergeExpressionSet) A replace function for study names.

[(mergeExpressionSet) A subset operator. Returns a mergeExpressionSet containing a subset of the studies. A mergeExpressionSet with only one study is returned as a single ExpressionSet.

summary (mergeExpressionSet) Obtain the basic information for 'mergeExpressionSet'.

plot (mergeExpressionSet) Draw scatterplots to compare integrative correlations for genes, here we use the approximate method to calculate the integrative correlation.

See Also

mergeExprs, intCor, modelOutcome, intcorDens, ExpressionSet

Examples

```
if(require(Biobase) & require(MASS)){
data(mergeData)
merged  <-mergeExprs(sample1,sample2,sample3)

merged[1:2]
i<-c(1,3)
```

```
merged[i]

exprs(merged)

names(merged)<-c("study1","study2","study3")

length(merged)

summary(merged)

plot(merged)

plot(merged[1:2])

intcorDens(merged)

inter  <- intersection(merged)
}
```

`intcorDens` *plot of density functions of integrative correlations*

Description

Given a mergeExpressionSet, this function calculates and plots the density function of the approximate integrative correlations, as well as densities for the "null distributions" obtained by randomly permuting sample IDs.

Usage

```
intcorDens(x,method,...)
```

Arguments

`x`	Object of class mergeExpressionSet.
`method`	The available method to use is "pearson".
`...`	Graphical parameters to be passed to plot.

Details

Here we use the approximate method to calculate the integrative correlation.

Value

The value is null. Returns a plot.

See Also

`mergeExpressionSet-class`,`intCor`,`modelOutcome`

Examples

```
if(require(Biobase) & require(MASS)){
data(mergeData)
merged  <-mergeExprs(sample1,sample2,sample3)

intcorDens(merged)

intcorDens(merged,cex.legend=1.5)

intcorDens(merged,lty=2)
}
```

`intCor` *Correlation of Correlations*

Description

Given a mergeExpressionSet, this function calculates the study specific correlation matrices, and, for each gene, the correlation of correlations.

Usage

```
intCor(x,method= c("pearson", "spearman"),exact,...)
```

Arguments

`x`	Object of class mergeExpressionSet.
`method`	Method used to calculate correlation coefficient. If exact is TRUE, the available methods to use is "spearman" and "pearson"; If exact is FALSE, the available methods to use is "pearson".
`exact`	If exact is TRUE, we use the standard method the calculate the integrative correlation; If exact is FALSE, we use the approximate method the calculate.
`...`	Not implemented at this time

Details

Integrative correlation coefficients are calcualted as follows. The first step is to identify the n genes common to all studies. Within each study, we calculate the correlation coefficient between gene g, and every other common gene. This gives a vector of length n-1. For a pair of studies, S1 and S2, we calculate the correlation of correlations for gene g. When there are more than 2 studies under consideration, all pairwise correlation of correlations are calculated and averaged.

Value

The output is an object of class mergeCor.

See Also

`mergeCor-class`,`intcorDens`

Examples

```
if(require(Biobase) & require(MASS)){
  data(mergeData)
  merged  <-mergeExprs(sample1,sample2,sample3)
  corcor  <-intCor(merged,method="spearman")

  plot(merged)
  hist(corcor)

  corcor  <-intCor(merged,method="pearson",exact=FALSE)
  corcor  <-intCor(merged[1:2])
  corcor  <-intCor(merged,exact=TRUE)

  vv<-c(1,3)
  corcor1  <-intCor(merged[vv])
  plot(merged,xlab="study A",ylab="study B",main="CORRELATION OF CORRELATION",col=3,pch=4
  hist(corcor1,xlab="CORRELATION OF CORRELATION")
 }
```

`intersection` *ExpressionSet with all common genes in a mergeExpressionSet*

Description

Given a mergeExpressionSet, this function returns a single ExpressionSet. Only genes common to all studies are included. Expression data for all studies sits side by side in the 'exprs' slot. The 'notes' slot is used for inormation about the study identity of each sample.

Usage

```
intersection(x)
```

Arguments

`x`	Object of class mergeExpressionSet.

Value

Returns an object of class ExpressionSet

See Also

`mergeExpressionSet-class`

Examples

```
data(mergeData)
merged  <-mergeExprs(sample1,sample2,sample3)

inter  <- intersection(merged)
```

mergeData *MergeMaid instance data for merging*

Description

These are 3 artificial data sets, generated using rnorm(), for the purpose of illustrating the package. All are list objects. For some of the "genes", the expression values in the 3 datasets are not independent.

Usage

```
data(mergeData)
```

mergeExprs *Merge gene expression data sets*

Description

Merges gene expression data from different studies.

Usage

```
mergeExprs(...)
```

Arguments

...	Input objects can be any combination of mergeExpressionSet, ExpressionSet, matrix or a list. A list should have the following slots: expression matrix, pheno data matrix, gene names vector, notes. The order of the four slots is fixed. A matrix should have genes ids as its row names, as should the exprs slot of an ExpressionSet. Since merging depends on geneids, these conventions are essential.

Details

The mergeExpressionSet object is the standard input for all functions in the MergeMaid package. Use the mergeExprs function when creating mergeExpressionSet objects to ensure that all necessary information is available for further analysis.

Value

The output is a mergeExpressionSet.

See Also

mergeExpressionSet-class

Examples

```
if(require(Biobase) & require(MASS)){
data(mergeData)
merged  <-mergeExprs(sample1,sample2,sample3)

rr<-rnorm(200*22,0,1)
mm<-matrix(rr,200,22)
rownames(mm)<-sample2[[3]]
merge.m<-mergeExprs(sample1,mm,sample2)
intcor.m<-intCor(merge.m)
plot(merge.m)

rr<-rnorm(200*50,0,1)
mm2<-matrix(rr,200,50)
ph.ll<-as.data.frame(rbinom(50,1,.5))
ll<-list(mm2,ph.ll,sample2[[3]],"list 2")
merge.t<-mergeExprs(sample1,mm,sample2,ll)
intcor.t<-intCor(merge.t)
plot(merge.t)

merge.a<-mergeExprs(sample3,merge.m,ll)
inter<-intersection(merge.a)
summary(merge.a)
}
```

---

modelOutcome *Compare regression coefficients across studies*

Description

Given a set of merged studies, this function calculates study specific regression coefficients for each gene.

Usage

```
modelOutcome(x,outcome,outcome2=NULL,method=c("linear","logistic","cox"),...)
```

Arguments

x	Object of class mergeExpressionSet.
method	Method specifies the model used to generate coefficients. At this time only linear regression, logistic regression, and Cox hazard rates are implemented.
outcome, outcome2	The format for the outcome variable depends on the model used. For linear regression, outcome should be a continous response variable, for logistic regression, it should be a binary response variable, and for Cox hazard rates it should be time of event. Outcome 2 is currently used only in the calculation of hazard rates, and should be a binary variable indicating censoring status for each subject. If outcome is a vector of length equal to number of studies, then each element represents the column in the ExpressionSet phenoData slot for that study. If outcome is a list, then each list element should have actual outcome data for the corresponding study.
...	Not implemented at this time

Value

The output is a mergeCoeff object.

See Also

modelOutcome, mergeCoeff-class

Examples

```
if(require(Biobase) & require(MASS) & require(survival)){
data(mergeData)
merged  <- mergeExprs(sample1,sample2,sample3)

log.coeff  <- modelOutcome(merged,outcome=c(1,1,1),method="logistic")
plot(coeff(log.coeff))

linear.coeff  <- modelOutcome(merged[1:2],outcome=c(3,3),method="linear")
plot(zscore(linear.coeff),xlab="study 1",ylab="study 2")

event1<-rbinom(100,1,.5)
event2<-rbinom(50,1,.5)
event3<-rbinom(70,1,.5)

out1<-rnorm(100,5,1)
out2<-rnorm(50,5,1)
out3<-rnorm(70,5,1)

out<-list(out1,out2,out3)
even<-list(event1,event2,event3)

cox.coeff<-modelOutcome(merged,outcome2=even,outcome=out,method="cox")
plot(coeff(cox.coeff))

}
```

Index