Package 'DExMA'

November 11, 2026

Type Package

Title Differential Expression Meta-Analysis

Version 1.20.0

Description performing all the steps of gene expression meta-analysis considering the possible existence of missing genes. It provides the necessary functions to be able to perform the different methods of gene expression meta-analysis. In addition, it contains functions to apply quality controls, download GEO datasets and show graphical representations of the results.

License GPL-2

Depends R (>= 4.1), DExMAdata

Suggests BiocStyle, qpdf, BiocGenerics, RUnit

Imports Biobase, GEOquery, impute, limma, pheatmap, plyr, scales, snpStats, sva, swamp, stats, methods, utils, bnstruct, RColorBrewer, grDevices

biocViews DifferentialExpression, GeneExpression, StatisticalMethod, QualityControl

Encoding UTF-8

LazyData false

git_url https://git.bioconductor.org/packages/DExMA

git_branch RELEASE_3_23

git_last_commit ffaffe0

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Juan Antonio Villatoro-García [aut, cre], Pedro Carmona-Sáez [aut]

Maintainer Juan Antonio Villatoro-García <juanantoniovillatorogarcia@gmail.com>

Contents

allSameID 2
batchRemove 3
calculateES 4
createObjectMA 5
dataLog 6
downloadGEOData 7
elementObjectMA 8
heterogeneityTest 9
makeHeatmap 10
metaAnalysisDE 13
missGenesImput 15
pvalueIndAnalysis 16

Index **18**

`allSameID` *Set all datasets in the same ID*

Description

Set all datasets in the same ID (Official Gene Symbol, Entrez or Ensembl)

Usage

```
allSameID(objectMA, finalID = "GeneSymbol",
    organism="Homo sapiens")
```

Arguments

`objectMA`	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the different samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
`finalID`	A character that indicates the final ID all the different studies will have. To know the available ids, you can write avaliableIDs.
`organism`	A character that indicates the organism of the data. To know the avaliable organisms write avaliableOrganism

Value

The same list with all the datasets in the same selected gene ID.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

createObjectMA

Examples

```
data(DExMAExampleData)
sameData <- allSameID(objectMA = maObjectDif, finalID = "GeneSymbol",
organism = "Homo sapiens")
sameData
```

---

batchRemove *Elimination of covariates batch effect or bias*

Description

It eliminates the effects of batch or bias of the covariates

Usage

```
batchRemove(expressionMatrix, pheno, formula, mainCov = NULL, nameGroup, ...)
```

Arguments

expressionMatrix	A matrix or data frame with genes in rows and samples in columns. An ExpressionSet object can be used too
pheno	A dataframe with samples in rows and covariates in colums.
formula	Formula of the covariates that are wanted to be corrected
mainCov	Name of the main covariate to be corrected
nameGroup	Name of the column of the Phenodata object in which the reference groups (cases and controls) are
...	other arguments are passed to lmFit fucntion of limma package

Value

The Expression Matrix with the bias or batch effect corrected. Moreover a plot of the visualization of the association between principal components and covariates is shown.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Martin Lauss (2019). swamp: Visualization, Analysis and Adjustment of High-Dimensional Data in Respect to Sample Annotations. R package version 1.5.1. https://CRAN.R-project.org/package=swamp

Examples

```
data(DExMAExampleData)
batchRemove(listMatrixEX$Study2, listPhenodatas$Study2, formula=~gender+race,
nameGroup="condition")
```

`calculateES` *Calculation of Effects Sizes and their variance*

Description

This function uses the Hedges'g estimator to calulate the different Effects size and their variances for each genes and for each dataset.

Usage

```
calculateES(objectMA, missAllow = 0.3)
```

Arguments

`objectMA`	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the diffenrent samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
`missAllow`	a number that indicates the maximun proportion of missing values allowed in a sample. If the sample has more proportion of missing values the sample will be eliminated. In the other case the missing values will be imputed using the K-NN algorithm.

Value

A list formed by three elements:

- First element (ES) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe represents the Effect Size.
- Second element (Var) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe represents the variance of the Effect size.
- Third element (logFC) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe represents the log Fold Changes.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

`createObjectMA`, `metaAnalysisDE`

Examples

```
data(DExMAExampleData)

resultsEffects <- calculateES(objectMA = maObject, missAllow = 0.3)
resultsEffects
```

`createObjectMA` *Creation of the object to use in meta-analysis*

Description

It allows the creation of an object to perform meta-analysis.

Usage

```
createObjectMA(
    listEX,
    listPheno = NULL,
    namePheno = c(rep(1, length(listEX))),
    expGroups = c(rep(1, length(listEX))),
    refGroups = c(rep(2, length(listEX)))
)
```

Arguments

`listEX`	A list of dataframes or matrix (genes in rows and sample in columns). A list of ExpressionSets can be used too
`listPheno`	A list of phenodatas (dataframes or matrix). If the object listEX is a list of ExpressionSets this element can be null.
`namePheno`	A list or vector of the different colunm names or positions from the phenodatas where the experimental and reference groups are identified. Each element of namePheno correspont to its equivalent element in the listPheno (default a vector of 1, all the first columns of each elements of listPheno are selected).
`expGroups`	A list of vectors or a vector containing the names or the positions with which we identify the elements of the experiment groups (cases) of the namePheno element (default a vector of 1, all the first groups are selected)
`refGroups`	A list of vectors or a vector containing the names or the positions with which we identify the elements of the reference groups (control) of the namePheno elements (default a vector of 1, all the first groups are selected)

Value

The object needed to perform meta-analysis. This object is list of nested lists. Each list contains two elements:

- The first element is the expression matrix (genes in rows and sample in columns)
- The second element is a vector of zeros and ones that represents the state of the diffenrent samples of the expression matrix. 0 represents reference group (controls) and 1 represents experimental group (cases).

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

elementObjectMA

Examples

```
data(DExMAExampleData)

phenoGroups = c("condition", "condition", "state", "state")
phenoCases = list(Study1 = "Diseased", Study2 = c("Diseased", "ill"),
                  Study3 = "Diseased", Study4 = "ill")
phenoControls = list(Study1 = "Healthy", Study2 = c("Healthy", "control"),
                     Study3 = "Healthy", Study4 = "control")

newObjectMA <- createObjectMA(listEX=listMatrixEX, listPheno = listPhenodatas,
                              namePheno=phenoGroups, expGroups=phenoCases,
                              refGroups = phenoControls)
newObjectMA
```

---

dataLog *Auxiliary function to check if data are log transfromed and transformed if it are not log-transformed*

Description

Auxiliary function to check if data are log transfromed and transformed if it are not log-transformed

Usage

```
dataLog(objectMA)
```

Arguments

objectMA	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the different samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA should be used.

Value

The same object with log-transformed expression matrix

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

createObjectMA

Examples

```
data(DExMAExampleData)

dataLog(maObject)
```

---

downloadGEOData *Download datasets from GEO database*

Description

Download different ExpressionSets objects from the GEO database

Usage

```
downloadGEOData(GEOobject, directory = getwd())
```

Arguments

GEOobject	a vector of character where each element represents a GEO object for downloading.
directory	The directory where the different downloaded GSE Series Matrix files from GEO will be stored. By default they are downloaded to the working directory

Details

This function internally uses getGEO function of GEOquery package. However, downloadGEO allows you to download multiple files at the same time.

Value

A list of the different ExpressionSets

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Davis, S. and Meltzer, P. S. GEOquery: a bridge between the Gene Expression Omnibus (GEO) and BioConductor. Bioinformatics, 2007, 14, 1846-1847

Examples

```
## Not run:
GEOobjects<- c("GSE4588", "GSE10325")
dataGEO<-downloadGEOData(GEOobjects)
dataGEO

## End(Not run)
```

---

elementObjectMA *Creation of the object to use in meta-analysis*

Description

It allows the creation of a element of the object needed to perform meta-analysis

Usage

```
elementObjectMA(
    expressionMatrix,
    pheno = NULL,
    groupPheno,
    expGroup = 1,
    refGroup = 2
)
```

Arguments

expressionMatrix	A dataframe or matrix that contanining genes in rows and samples if columns. An ExpressionSet object can be used too.
pheno	A data frame or a matrix containing samples in rows and covariates in columns. If NULL (default), pheno is extracted from the ExpressionSet object
groupPheno	The column name or position from pheno where experimental group (cases) and reference group (control) are identified

expGroup	The group name or position from groupPheno variable used as experimental group (cases). By default the first group (character) is taken
refGroup	The group name or position from groupPheno variable used as reference group (control). By default the second group (character) is taken

Value

An element that can be included in meta-analysis object.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

createObjectMA

Examples

```
data(DExMAExampleData)

ExpressionSetStudy5
newElem <-elementObjectMA(expressionMatrix = ExpressionSetStudy5,
                          groupPheno = "condition",
                          expGroup = c("Diseased", "ill"),
                          refGroup = c("Healthy", "control"))
```

---

heterogeneityTest *Checking the heterogeneity of the different studies*

Description

Shows a QQ-plot of the Cochran's test and the quantiles of I^2 statistic values to mesuare heterogeneity

Usage

```
heterogeneityTest(objectMA, probs = c(0, 0.25, 0.5, 0.75))
```

Arguments

objectMA	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the different samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
probs	Numeric vector of probabilities with values between 0 and 1. It indicates the I^2 quantiles that will be returned

Details

If in the QQ-plot of the Cochran's test most of the values are close to the central line (most of the Cochran's test values are close to the expected distribution), it can be said that there is homogeneity. In the case that these values deviate greatly from the expected distribution, it must be assumed that there is heterogeneity. I^2 measures the percentage of variation across studies due to heterogeneity. To assume homogeneity in the gene expression meta-analysis, almost all I^2 values (quantiles) must be 0 or at least less than 0.25.

Value

Quantiles of the I^2 values

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Higgins JPT, Thompson SG. Quantifying heterogeneity in a meta-analysis. Stat Med 2002;21:1539–58.

Higgins JPT, Thompson SG, Deeks JJ, et al. Measuring inconsistency in meta-analyses. BMJ 2003;327:557–60.

See Also

createObjectMA

Examples

```
data(DExMAExampleData)

heterogeneityTest(maObject)
```

---

makeHeatmap *Visualization of the meta-analysis results*

Description

It allows to see how the different significant genes are expressed in the different samples. Missing genes appear in gray

Usage

```
makeHeatmap(
    objectMA,
    resMA,
    scaling=c("zscor","rscale","swr","none"),
    regulation=c("all", "up","down"),
    breaks=c(-2,2),
    fdrSig = 0.05,
    logFCSig = 1.5,
    numSig = "all",
    color = colorRampPalette(rev(brewer.pal(n = 7, name = "RdYlBu")))(100),
    na_col = "darkgrey",
    legend = TRUE,
    cluster_cols = FALSE,
    cluster_rows = FALSE,
    show_rownames = TRUE,
    show_colnames = FALSE)
```

Arguments

objectMA	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the different samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
resMA	Output generated by the differents functions that performs meta-analysis (metaES, metaPvalue, metaRank or metaAnalysisDE)
scaling	Character variable to choose between different scaling approaches. See "Details" for more information.
regulation	Character variable that indicates whether we want the heatmap to show all significant genes ("all"), only the up-regulated genes ("up") or only the down-regulated genes("down")
breaks	Numeric vector of length 2 that contains the extreme values (minimum and maximum) of the range of values in which the heatmap color scale will be distributed. Default a vector By default a vector of -2 and 2 as extreme values.
fdrSig	Adjusted p-value from which a gene is considered significant. Default 0.05
logFCSig	In absolute value. Log Fold Change threshold from which genes are considered in the heatmap.
numSig	The number of most significant genes to be represented. If numSig = "all", all significant genes that meet the selected parameters will be represented.
color	Vector of colors used in heatmap
na_col	Color of the NA cell in the heatmap
legend	Logical to determine if legend should be drawn or not
cluster_cols	boolean values determining if columns should be clustered.
cluster_rows	boolean values determining if rows should be clustered.

show_rownames	boolean specifying if row names are be shown.
show_colnames	boolean specifying if column names are be shown.

Details

Scaling approaches that can be used are:

- "rscale": it applies rescale function of *scales* package. Values will be between -1 and 1)
- "zscor": It calculates a z-score value for each gene, that is, the mean gene expression from each gene is subtracted from each gene expression value and then it is divided by the standard deviation
- "swr": it applys scaling relative to reference dataset approach
- "none": any scaling approach it is applied.

Value

The matrix represented in the heatmap

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Hadley Wickham and Dana Seidel (2020). scales: Scale Functions for Visualization. R package version 1.1.1. https://CRAN.R-project.org/package=scales

Lazar, C, Meganck, S, Taminau, J, and et al. 2013. "Batch Effect Removal Methods for Microarray Gene Expression Data Integration: A Survey," 469–90.

Raivo Kolde 2019. pheatmap: Pretty Heatmaps. R package version 1.0.12. https://CRAN.R-project.org/package=pheatmap

See Also

createObjectMA, metaAnalysisDE

Examples

```
data(DExMAExampleData)

resultsMA <- metaAnalysisDE(maObject, typeMethod="REM")
makeHeatmap(objectMA=maObject, resMA=resultsMA,
scaling = "zscor", regulation = "all",breaks=c(-2,2),
fdrSig = 0.05, logFCSig = 1.5, numSig=40)
```

`metaAnalysisDE` *Performing Meta-analysis*

Description

It performs meta-analysis using seven different methods.

Usage

```
metaAnalysisDE(
    objectMA = NULL,
    effectS = NULL,
    pvalues = NULL,
    weight = NULL,
    typeMethod = c("FEM", "REM", "maxP", "minP", "Fisher",
                   "Stouffer", "ACAT"),
    missAllow = 0.3,
    proportionData = 0.5
)
```

Arguments

`objectMA`	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the diffenrent samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
`effectS`	A list of two elements. The first element is a dataframe with genes in rows and studies in columns. Each component of the dataframe is the effect of a gene in a study. The second element of the list is also a dataframe with the same structure, but in this case each component of the dataframe represent the variance of the effect of a gene in a study. The third element of the list is also a dataframe with the same structure, but in this case each component of the dataframe represent the log fold change of a gene in a study. This argument should be only used in the case that objectMA argument is null.
`pvalues`	A list of two elements. The first element is a dataframe with genes in rows and studies in columns. Each component of the dataframe is the p-value of a gene in a study. The second element of the list is also a dataframe with the same structure, but in this case each component of the dataframe represent the log fold change of a gene in a study. This argument should be only used in the case that objectMA argument is null.
`weight`	A vector of the weights of each dataset. This argument should only be included in case objectMA is null and you want to use "Stouffer" or "ACAT" method.
`typeMethod`	A character that indicates the method to be peformed. See "Details"for more information

`missAllow` A number that indicates the maximun proportion of missing values allowed in a sample. If the sample has more proportion of missing values the sample will be eliminated. In the other case the missing values will be imputed using the K-NN algorithm. In case the objectMA has been previously imputed, this element is not necessary.

`proportionData` The minimum proportion of datasets in which a gene must be contained to be included. By default, the gene must be contained in at least half of the datasets. In case the objectMA has been previously imputed, this element is not necessary.

Details

The different meta-analysis methods that can be applied are:

1. **Effects sizes methods:**
   - "FEM": Fixed Effects model
   - "REM": Random Effects model
2. **P-value combination mehods**
   - "Fisher": Fisher's methods
   - "Stouffer": Stouffer's method
   - "maxP": maximum p-value method (Wilkinson's method)
   - "minP": minimum p-value method (Tippett's method)
   - "ACAT": Aggregated Cauchy Association Test method

Value

A dataframe with the meta-analysis results. Depending on the applied method, a different dataframe is obtained. For more information see the package vignette.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Daniel Toro-Domínguez, Juan Antonio Villatoro-García, Jordi Martorell-Marugán, Yolanda Román-Montoya, Marta E Alarcón-Riquelme, Pedro Carmona-Sáez, A survey of gene expression meta-analysis: methods and applications, Briefings in Bioinformatics, 2020;, bbaa019, https://doi.org/10.1093/bib/bbaa019

Michael Dewey (2020). metap: meta-analysis of significance values.

Liu, Y., Chen, S., Li, Z., Morrison, A. C., Boerwinkle, E., & Lin, X. (2019). ACAT: A Fast and Powerful p Value Combination Method for Rare-Variant Analysis in Sequencing Studies. The American Journal of Human Genetics, 104(3), 410-421. https://doi.org/10.1016/j.ajhg.2019.01.002

Examples

```
data(DExMAExampleData)
ResultsMA <- metaAnalysisDE(objectMA=maObject, typeMethod="REM",
                            missAllow=0.3, proportionData=0.5)
ResultsMA
```

`missGenesImput` *Imputation of unmeasured genes*

Description

missGenesImput uses k-nearest neighbors in the space of samples to impute the unmeasured genes of the different datasets.

Usage

```
missGenesImput(objectMA, k = 7)
```

Arguments

`objectMA`	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the different samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateobjectMA can be used too.
`k`	Number of neighbors to be used in the imputation (default=7).

Value

A list formed by two elements:

- First element (objectMA) the same objectMA with missign genes imputed
- Second element (imputIndicators) a list with 4 different objects:
  - imputValuesSample: Number of missing values imputed per sample
  - imputPercentageSample: Percentage of missing values imputed per sample
  - imputValuesGene: Number of missing values imputed per gene
  - imputPercentageGene: Percentage of missing values imputed per gene

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

References

Christopher A Mancuso, Jacob L Canfield, Deepak Singla, Arjun Krishnan, A flexible, interpretable, and accurate approach for imputing the expression of unmeasured genes, Nucleic Acids Research, Volume 48, Issue 21, 2 December 2020, Page e125, https://doi.org/10.1093/nar/gkaa881

Alberto Franzin, Francesco Sambo, Barbara di Camillo. bnstruct: an R package for Bayesian Network structure learning in the presence of missing data. Bioinformatics, 2017; 33 (8): 1250-1252, Oxford University Press, https://doi.org/10.1093/bioinformatics/btw807

See Also

createObjectMA, metaAnalysisDE

Examples

```
data(DExMAExampleData)
missGenesImput(maObject)
```

---

pvalueIndAnalysis *Calculation p-value for each gene and study*

Description

This function uses t-test based on limma package in other to obtain the individual p-values for each study and gene

Usage

```
pvalueIndAnalysis(objectMA, missAllow = 0.3)
```

Arguments

objectMA	A list of list. Each list contains two elements. The first element is the expression matrix (genes in rows and sample in columns) and the second element is a vector of zeros and ones that represents the state of the diffenrent samples of the expression matrix. 0 represents one group (controls) and 1 represents the other group (cases). The result of the CreateObjectMA can be used too.
missAllow	a number that indicates the maximun proportion of missing values allowed in a sample. If the sample has more proportion of missing values the sample will be eliminated. In the other case the missing values will be imputed using the K-NN algorithm.

Value

A list formed by two elements:

- First element (p) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe represents the p-value.
- Second element (logFC) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe is the logFC.
- Third element (weights_z) is a dataframe were columns are each of the studies (datasets) and rows are the genes. Each element of the dataframe represents the necessary weights for Stouffer's method.

Author(s)

Juan Antonio Villatoro Garcia, <juanantoniovillatorogarcia@gmail.com>

See Also

createObjectMA, metaAnalysisDE

Examples

```
data(DExMAExampleData)

pvalues <- pvalueIndAnalysis(objectMA=maObject, missAllow=0.3)
pvalues
```

Index

allSameID, 2

batchRemove, 3

calculateES, 4
createObjectMA, *3*, *4*, 5, *7*, *9*, *10*, *12*, *16*, *17*

dataLog, 6
downloadGEOData, 7

elementObjectMA, *6*, 8

heterogeneityTest, 9

makeHeatmap, 10
metaAnalysisDE, *4*, *12*, 13, *16*, *17*
missGenesImput, 15

pvalueIndAnalysis, 16