Package 'MultiBaC'

June 2, 2025

Type Package

Title Multiomic Batch effect Correction

Version 1.18.0

Author person(``Manuel'', ``Ugidos'', email = ``manuelugidos@gmail.com''), person(``Sonia'', ``Tarazona'', email = ``sotacam@gmail.com''), person(``María José'', ``Nueda'', email = ``mjnueda@ua.es'')

Maintainer The package maintainer <manuelugidos@gmail.com>

Description MultiBaC is a strategy to correct batch effects from multiomic datasets distributed across different labs or data acquisition events. MultiBaC is the first Batch effect correction algorithm that dealing with batch effect correction in multiomics datasets. MultiBaC is able to remove batch effects across different omics generated within separate batches provided that at least one common omic data type is included in all the batches considered.

License GPL-3

Encoding UTF-8

biocViews Software, StatisticalMethod, PrincipalComponent, DataRepresentation, GeneExpression, Transcription, BatchEffect

Imports Matrix, ggplot2, MultiAssayExperiment, ropls, graphics, methods, plotrix, grDevices, pcaMethods

Suggests knitr, rmarkdown, BiocStyle, devtools

VignetteBuilder knitr

Collate 'auxfunctions.R' 'ARSyNcomponents.R' 'ASCA1f.R' 'PCA-GENES.R' 'ASCAfunres.R' 'ASCAfun12.R' 'ASCAfun1.R' 'ASCA2f.R' 'ARSyNbac.R' 'MultiBaC.R' 'createMbac.R' 'plot-methods.R' 'summary-function.R'

NeedsCompilation no

RoxygenNote 7.1.1

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

git_branch RELEASE_3_21

git_last_commit aeda3f8

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-06-01

Contents

`A.gro` *Transcription rate data from yeast*

Description

This data set gives the transcription rates of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 6 rows and 200 variables.

Source

Gene Expression Omnimus: GSE11521

References

[1] Pelechano, V. and Pérez-Ortín, J.E. (2010), There is a steady-state transcriptome in exponentially growing yeast cells. Yeast, 27: 413-422. https://doi.org/10.1002/yea.1768

[2] García-Martínez, J. and Aranda, A. and Pérez-Ortín, J. E. (2004). Genomic Run-On Evaluates Transcription Rates for All Yeast Genes and Identifies Gene Regulatory Mechanisms, Molecular Cell, 15. https://doi.org/10.1016/j.molcel.2004.06.004

[3] Pelechano, V., Chávez, S., & Pérez-Ortín, J. E. (2010). A complete set of nascent transcription rates for yeast genes. PloS one, 5(11), e15442. https://doi.org/10.1371/journal.pone.0015442

`A.rna` *Gene expression data from yeast*

Description

This data set gives the expression of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 6 rows and 200 variables.

Source

Gene Expression Omnimus: GSE11521

References

[1] Pelechano, V. and Pérez-Ortín, J.E. (2010), There is a steady-state transcriptome in exponentially growing yeast cells. Yeast, 27: 413-422. https://doi.org/10.1002/yea.1768

[2] García-Martínez, J. and Aranda, A. and Pérez-Ortín, J. E. (2004). Genomic Run-On Evaluates Transcription Rates for All Yeast Genes and Identifies Gene Regulatory Mechanisms, Molecular Cell, 15. https://doi.org/10.1016/j.molcel.2004.06.004

[3] Pelechano, V., Chávez, S., & Pérez-Ortín, J. E. (2010). A complete set of nascent transcription rates for yeast genes. PloS one, 5(11), e15442. https://doi.org/10.1371/journal.pone.0015442

ARSyNbac *ARSyNbac*

Description

ARSyNbac

Usage

```
ARSyNbac(
  mbac,
  batchEstimation = TRUE,
  filterNoise = TRUE,
  Interaction = FALSE,
  Variability = 0.9,
  beta = 2,
  modelName = "Model 1",
  showplot = TRUE
)
```

Arguments

mbac	mbac object generated by *createMbac*.
batchEstimation	Logical. If TRUE (default) the batch effect is estimated and used to correct the data. Use TRUE when the source of the batch effect is known.
filterNoise	Logical. If TRUE (default) structured noise is removed form residuals. Use this option when there is an unknown source of batch effect in data.
Interaction	Logical. Whether to model the interaction between factors or not (FALSE by default).
Variability	From 0 to 1. Minimum percent of data variability that must be explained by each model. By default, 0.90.
beta	Numeric. Components that represent more than beta times the average variability are identified as systematic noise in residuals. Used in noise reduction mode. By default, 2.
modelName	Name of the model created. This name will be showed if you use the explained_var plot function. By default, "Model 1".
showplot	Logical. If TRUE (default), the explained_var plot is showed. This plot represents the number of components selected for the ARSyN model.

Value

Custom mbac object. Elements in a mbac object:

1. ListOfBatches: A list of MultiAssayExperiment objects (one per batch).

2. commonOmic Name of the common omic between the batches. It must be one of the names in omicNames argument. If NULL (default), the omic names that appears more times is selected as commonOmic.
3. CorrectedData: Same structure than ListOfBatches but with the corrected data instead of the original.
4. ARSyNmodels: ARSyN models created during MultiBaC performance (one per omic data type).

References

Nueda MJ, Ferrer A, Conesa A. ARSyN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments. Biostatistics. 2012;13:553–66.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, B.rna, C.rna),
                       batchFactor = c("A", "B", "C"),
                       experimentalDesign = list("A" =  c("Glu+",
                       "Glu+", "Glu+", "Glu-",
                       "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = "RNA")
my_final_mbac <- ARSyNbac (my_mbac, batchEstimation = TRUE, filterNoise = TRUE,
                  Interaction=TRUE, Variability = 0.90, beta = 2,
                  modelName = "Model 1",
                  showplot = FALSE)
```

B.ribo *Gene translation rate data from yeast*

Description

This data set gives the translation rates of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 4 rows and 200 variables.

Source

Gene Expression Omnimus: GSE56622

References

[1] Zid, B. M., & O'Shea, E. K. (2014). Promoter sequences direct cytoplasmic localization and translation of mRNAs during starvation in yeast. Nature, 514(7520), 117–121. https://doi.org/10.1038/nature13578

B.rna *Gene expression data from yeast*

Description

This data set gives the expression of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 4 rows and 200 variables.

Source

Gene Expression Omnimus: GSE56622

References

[1] Zid, B. M., & O'Shea, E. K. (2014). Promoter sequences direct cytoplasmic localization and translation of mRNAs during starvation in yeast. Nature, 514(7520), 117–121. https://doi.org/10.1038/nature13578

---

batchCorrection *batchCorrection*

Description

This function performs the ARSyNbac correction [1] for each omic contained in mulBatchDesign input object.

Usage

```
batchCorrection(mbac, multiBatchDesign, Interaction = FALSE,
  Variability = 0.9)
```

Arguments

mbac	mbac object generated by createMbac. PLS models slot must be present.
multiBatchDesign	A list containing the original and predicted omic for each batch. All omics must be present in every batch. Output object of genMissingOmics function
Interaction	Logical. Whether to model the interaction between experimental factors and bacth factor in ARSyN models. By default, FALSE.
Variability	From 0 to 1. Minimum percent of data variability that must be explained for each ARSyN model. By default, 0.90.

Value

Custom mbac object. Elements in a mbac object:

1. ListOfBatches: A list of MultiAssayExperiment objects (one per batch).
2. commonOmic: Name of the common omic between the batches.
3. CorrectedData: Same structure than ListOfBatches but with the corrected data instead of the original.
4. PLSmodels: PLS models created during MultiBaC method performance (one model per non-common omic data type).
5. ARSyNmodels: ARSyN models created during MultiBaC performance (one per omic data type).
6. InnerRelation: Table of class data.frame containing the inner correlation (i.e. correlation between the scores of X (t) and Y (u) matrices) for each PLS model across all components.

References

[1] Nueda MJ, Ferrer A, Conesa A. ARSyN: A method for the identification and removal of systematic noise in multifactorial time course microarray experiments. Biostatistics. 2012;13(3):553-566. doi:10.1093/biostatistics/kxr042

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+", "Glu+",
                       "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                       commonOmic = "RNA")

my_mbac_2 <- genModelList (my_mbac, test.comp = NULL,
                           scale = FALSE, center = TRUE,
                           crossval = NULL,
                           showinfo = TRUE)
```

```
multiBatchDesign <- genMissingOmics(my_mbac_2)
my_finalwise_mbac <- batchCorrection(my_mbac_2,
                                     multiBatchDesign = multiBatchDesign,
                                     Interaction = FALSE,
                                     Variability = 0.9)
```

---

batchEstPlot *batchEstPlot*

Description

This function uses linear models to estimate the batch effect magnitude using the common data across batches. It compares the result with theoretical distribution of diferrent levels of batch magnitude.

Usage

```
batchEstPlot(mbac, ...)
```

Arguments

mbac	Object of class mbac generated by *createMbac*.
...	Other graphical parameters.

Value

Batch estimation plot is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

batchEstPlot (my_mbac)
```

C.par *Protein binded mRNA data from yeast*

Description

This data set gives the protein binding information of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 4 rows and 200 variables.

Source

Gene Expression Omnimus: GSE43747

References

[1] Freeberg, M. A., Han, T., Moresco, J. J., Kong, A., Yang, Y. C., Lu, Z. J., Yates, J. R., & Kim, J. K. (2013). Pervasive and dynamic protein binding sites of the mRNA transcriptome in Saccharomyces cerevisiae. Genome biology, 14(2), R13. https://doi.org/10.1186/gb-2013-14-2-r13

C.rna *Gene expression data from yeast*

Description

This data set gives the expression of 200 genes from yeast measured at two different conditions (glucose vs glucose starvation).

Usage

```
multiyeast
```

Format

A matrix containing 4 rows and 200 variables.

Source

Gene Expression Omnimus: GSE43747

References

[1] Freeberg, M. A., Han, T., Moresco, J. J., Kong, A., Yang, Y. C., Lu, Z. J., Yates, J. R., & Kim, J. K. (2013). Pervasive and dynamic protein binding sites of the mRNA transcriptome in Saccharomyces cerevisiae. Genome biology, 14(2), R13. https://doi.org/10.1186/gb-2013-14-2-r13

`createMbac` *createMbac*

Description

This function creates a list object to be used by MultiBaC function from a set of matrix R objects.

Usage

```
createMbac(inputOmics, batchFactor = NULL, experimentalDesign, omicNames,
  commonOmic = NULL)
```

Arguments

`inputOmics`	A list containing all the matrices or data.frame objects to be analysed. MultiAssayExperiment objects can alternatively be provided.
`batchFactor`	Either a vector or a factor indicating the batch were each input matrix belongs to (i.e. study, lab, time point, etc.). If NULL (default) no batch is considered and just ARSyNbac noise reduction mode could be applied.
`experimentalDesign`	A list with as many elements as batches. Each element can be a factor, a character vector or a data.frame indicating the experimental conditions for each sample in that batch. When being a data.frame with more than one column (multifactorial experimental designs), the different columns will be combined into a single one to be used by MultiBaC or ARSyNbac. In any case, the experimental setting must be the same for all batches. In addition, the names of the elements in this list must be the same as declared in batches argument. If not (or if NULL), names are forced to be the same in as in batches argument and in the same order.
`omicNames`	Vector of names for each input matrix. The common omic is required to have the same name across batches.
`commonOmic`	Name of the common omic between the batches. It must be one of the names in omicNames argument. If NULL (default), the omic name which is common to all batches is selected as commonOmic.

Value

Custom mbac object. Elements in a mbac object:

1. ListOfBatches: A list of MultiAssayExperiment objects (one per batch).
2. commonOmic Name of the common omic between batches.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                       commonOmic = "RNA")
```

---

explained_varPlot *explained_varPlot*

Description

explained_varPlot

Usage

```
explained_varPlot(mbac, ...)
```

Arguments

mbac	Object of class mbac generated by *ARSyNbac*, *MultiBaC* or *batchCorrection*.
...	Other graphical parameters.

Value

Explained variance plot for ARSyN models is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
                           center = TRUE, crossval = NULL,
```

```
                          Variability = 0.90,
                          Interaction = TRUE ,
                          showplot = FALSE,
                          showinfo = FALSE)

explained_varPlot (my_final_mbac)
```

---

genMissingOmics *genMissingOmics*

Description

This function generates for all the batches the omic data they had not originally. This is the previous step to apply ARSyNbac [1] correction.

Usage

```
genMissingOmics(mbac)
```

Arguments

mbac	mbac object generated by *createMbac*. PLS models slot must be present.

Value

A list of *MultiAssayExperiment* structures. In this case, each batch contains all the omics introduced in MultiBaC. For instance, if two batches are being studying, "A" and "B", given that "A" contains "RNA-seq" and "GRO-seq" data and "B" contains "RNA-seq" and "Metabolomica" data, after applying *genMissingOmics* function batch "A" will contain "RNA-seq", "GRO-seq" and predicted "Metabolomics" data.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                       commonOmic = "RNA")

my_mbac_2 <- genModelList (my_mbac, test.comp = NULL,
                           scale = FALSE, center = TRUE,
                           crossval = NULL,
                           showinfo = TRUE)
multiBatchDesign <- genMissingOmics(my_mbac_2)
```

genModelList *genModelList*

Description

This function performs PLS models for every batch. A PLS model is generated for each non-common omic in each batch.

Usage

```
genModelList(mbac, test.comp = NULL, scale = FALSE, center = TRUE,
  crossval = NULL, showinfo = TRUE)
```

Arguments

mbac	mbac object generated by *createMbac*.
test.comp	Maximum number of components allowed for PLS models. If NULL (default), the minimal effective rank of the matrices is used as the maximum number of components.
scale	Logical. Whether X and Y matrices must be scaled. By default, FALSE.
center	Logical. Whether X and Y matrices must be centered. By default, TRUE.
crossval	Integer: number of cross-validation segments. The number of samples (rows of 'x') must be at least >= crossvalI. If NULL (default) leave-one-out crossvalidation is performed.
showinfo	Logical. Whether to show the information about the function progress. By default, TRUE.

Value

Custom mbac object. Elements in a mbac object:

1. ListOfBatches: A list of MultiAssayExperiment objects (one per batch).
2. commonOmic: Name of the common omic between the batches.
3. PLSmodels: PLS models created during MultiBaC method performance (one model per non-common omic data type).
4. InnerRelation: Table of class data.frame containing the inner correlation (i.e. correlation between the scores of X (t) and Y (u) matrices) for each PLS model across all components.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
```

```
                          "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                           "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                         omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                         commonOmic = "RNA")

my_mbac_2 <- genModelList (my_mbac, test.comp = NULL,
                            scale = FALSE, center = TRUE,
                            crossval = NULL,
                            showinfo = TRUE)
```

---

inner_relPlot *inner_relPLot*

Description

inner_relPLot

Usage

```
inner_relPlot(mbac, comp2plot = c(1, 2), ...)
```

Arguments

mbac	Object of class mbac generated by *MultiBaC* or *genModelList*.
comp2plot	Indicates which components are plotted. Default is "c(1,2)", which means that just the inner relation plot of components 1 and 2 are shown. If more components are indicated, the function will return as many plots as needed to show all the components.
...	Other graphical parameters

Value

Inner relation plot for PLS models is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
```

```
                              center = TRUE, crossval = NULL,
                              Variability = 0.90,
                              Interaction = TRUE ,
                              showplot = FALSE,
                              showinfo = FALSE)

inner_relPlot (my_final_mbac)
```

MultiBaC *MultiBaC*

Description

MultiBaC performs a multi-omic, multi-batch correction

MultiBaC is a strategy to correct batch effects from multiomic datasets distributed across different labs or data acquisition events. MultiBaC is the first Batch effect correction algorithm that dealing with batch effect correction in multiomics datasets. MultiBaC is able to remove batch effects across different omics generated within separate batches provided that at least one common omic data type is included in all the batches considered.

Usage

```
MultiBaC(mbac, test.comp = NULL, scale = FALSE, center = TRUE,
  showplot = TRUE, crossval = NULL, Interaction = FALSE,
  Variability = 0.9, showinfo = TRUE)
```

Arguments

mbac	mbac object generated by createMbac.
test.comp	Maximum number of components allowed for PLS models. If NULL (default), the minimal effective rank of the matrices is used as the maximum number of components.
scale	Logical. Whether X and Y matrices must be scaled. By default, FALSE.
center	Logical. Whether X and Y matrices must be centered. By default, TRUE.
showplot	Logical. If TRUE (default), the Q2 and the explained variance plots are shown.
crossval	Integer: number of cross-validation segments. The number of samples (rows of 'x') must be at least >= crossvalI. If NULL (default), a leave-one-out crossvalidation is performed.
Interaction	Logical. Whether to model the interaction between experimental factors and bacth factor in ARSyN models. By default, FALSE.
Variability	From 0 to 1. Minimum percent of data variability that must be explained by each ARSyN model. By default, 0.90.
showinfo	Logical. If TRUE (default), the information about the function progress is shown.

Value

Custom mbac object. Elements in a mbac object:

1. ListOfBatches: A list of MultiAssayExperiment objects (one per batch).
2. commonOmic: Name of the common omic between the batches.
3. CorrectedData: Same structure than ListOfBatches but with the corrected data instead of the original.
4. PLSmodels: PLS models created during MultiBaC method performance (one model per non-common omic data type).
5. ARSyNmodels: ARSyN models created during MultiBaC performance (one per omic data type).
6. InnerRelation: Table of class data.frame containing the inner correlation (i.e. correlation between the scores of X (t) and Y (u) matrices) for each PLS model across all components.

References

Ugidos, M., Tarazona, S., Prats-Montalbán, J. M., Ferrer, A., & Conesa, A. (2020). MultiBaC: A strategy to remove batch effects between different omic data types. Statistical Methods in Medical Research. https://doi.org/10.1177/0962280220907365

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                       commonOmic = "RNA")

my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
                           center = TRUE, crossval = NULL,
                           Variability = 0.90,
                           Interaction = TRUE ,
                           showplot = FALSE,
                           showinfo = FALSE)
```

`multiyeast` *A distributed yeast multiomic dataset*

Description

The yeast expression data sets were collected from the Gene Expression Omnibus (GEO) database and from three different studies. All of them analyzed the effects of glucose starvation in yeast. Lab A is the Department of Biochemistry and Molecular Biology from Universitat de Valencia (accession number GSE11521) [@JE1, @JE2, @JE3]; Lab B is the Department of Molecular and Cellular Biology from Harvard University (accession number GSE56622) [@RIBO]; and Lab C is the Department of Biology from Johns Hopkins University (accession number GSE43747) [@PARCLIP]. These studies used equivalent yeast strains and experimental conditions but, as shown in Figure 1, the main effect on expression is due to data belonging to different labs, which are the batches in this case.

Usage

```
data("multiyeast")
```

Format

6 matrices with 6 or 4 observations on 200 variables (genes).

`A.rna` Gene expression data matrix from lab A.

`A.gro` Transcription rates data matrix from lab A.

`B.rna` Gene expression data matrix from lab B.

`B.ribo` Translation rates data matrix from lab B.

`C.rna` Gene expression data matrix from lab C.

`C.par` Protein binded RNA data matrix from lab C.

Details

The yeast expression data sets were collected from the Gene Expression Omnibus (GEO) database and from three different studies. All of them analyzed the effects of glucose starvation in yeast. Lab A is the Department of Biochemistry and Molecular Biology from Universitat de Valencia (accession number GSE11521) [@JE1, @JE2, @JE3]; Lab B is the Department of Molecular and Cellular Biology from Harvard University (accession number GSE56622) [@RIBO]; and Lab C is the Department of Biology from Johns Hopkins University (accession number GSE43747) [@PARCLIP]. These studies used equivalent yeast strains and experimental conditions but, as shown in Figure 1, the main effect on expression is due to data belonging to different labs, which are the batches in this case. After a proper data pre-processing for each case, a voom transformation (limma R package) was applied when necessary. Finally TMM normalization was performed on the whole set of samples from all labs. A reduced dataset was obtained by selecting 200 omic variables from each data matrix and just 3 samples from lab A. This yeast multiomic reduced dataset is included in *MutiBaC* package to illustrate the usage of the package. The gene expression matrices can be loaded by using the *data("multiyeast")* instruction.

Source

https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi

References

[1] Pelechano, V. and Pérez-Ortín, J.E. (2010), There is a steady-state transcriptome in exponentially growing yeast cells. Yeast, 27: 413-422. https://doi.org/10.1002/yea.1768

[2] García-Martínez, J. and Aranda, A. and Pérez-Ortín, J. E. (2004). Genomic Run-On Evaluates Transcription Rates for All Yeast Genes and Identifies Gene Regulatory Mechanisms, Molecular Cell, 15. https://doi.org/10.1016/j.molcel.2004.06.004

[3] Pelechano, V., Chávez, S., & Pérez-Ortín, J. E. (2010). A complete set of nascent transcription rates for yeast genes. PloS one, 5(11), e15442. https://doi.org/10.1371/journal.pone.0015442

[4] Zid, B. M., & O'Shea, E. K. (2014). Promoter sequences direct cytoplasmic localization and translation of mRNAs during starvation in yeast. Nature, 514(7520), 117–121. https://doi.org/10.1038/nature13578

[5] Freeberg, M. A., Han, T., Moresco, J. J., Kong, A., Yang, Y. C., Lu, Z. J., Yates, J. R., & Kim, J. K. (2013). Pervasive and dynamic protein binding sites of the mRNA transcriptome in Saccharomyces cerevisiae. Genome biology, 14(2), R13. https://doi.org/10.1186/gb-2013-14-2-r13

Examples

```
data(multiyeast)
head(A.rna)
```

`plot.mbac` *Plot Method for mbac*

Description

plot function for `mbac` class.

Usage

```
## S3 method for class 'mbac'
plot(x, y = NULL, typeP = "def", col.by.batch = TRUE,
  col.per.group = NULL, comp2plot = c(1, 2), legend.text = NULL,
  args.legend = NULL, ...)
```

Arguments

`x`	mbac object generated by *createMbac*, *ARSyNbac* or *MultiBaC*.
`y`	Currently not used.
`typeP`	type of plot showed. See details.
`col.by.batch`	Argument for PCA plots. TRUE or FALSE. If TRUE (default), samples are colored according to the batch factor. If FALSE, samples are colored according to the experimental conditions.

col.per.group	Argument for PCA plots. Color for each group (given by batches or experimental conditions). If NULL (default), the colors are taken from a predefined pallete.
comp2plot	Argument for PCA or InnerRel plot. It indicates which components are to be plotted. The default is c(1,2), which means that, in PCA plots, component 1 is plotted in "x" axis and component 2 in "y" axis, and for InnerRel plots, the inner relation plots of components 1 and 2 are to be shown. If more components are indicated, the function will return as many plots as needed to show all the components.
legend.text	A vector of text used to construct a legend for the plot. Argument for PCA plot. If NULL (default) batch or conditions names included in the mbac object are used.
args.legend	list of additional arguments to pass to legend(); names of the list are used as argument names. Only used if legend.text is supplied.
...	Other graphical arguments.

Details

typeP options are: "def" (default option, "Q2 plot" and "Explained variance plot" in case of MultiBaC and "Explained variance plot" in case of ARSyNbac outputs), "inner" (inner correlation plots for each PLS model acroos the components for MultiBaC output), "pca.org" (PCA plot of original data for MultiBaC or ARSyNbac outputs), "pca.cor" (PCA plot of corrected data for MultiBaC or ARSyNbac outputs), "pca.both" (PCA plots for both original and corrected data for MultiBaC or ARSyNbac outputs), and "batch" ("Batch effect estimation" plot for all the output). Remember that PCA plots can only be generated when all the omics share the same variable space (e.g. gene identifiers are provided as names of variables for all data matrices). While the *plot* function can generate all the plot types described above, each plot can also be independently generated by its corresponding function: *Q2_plot (mbac)*, *explained_varPlot (mbac)*, *plot_pca (mbac, typeP = c("pca.org", "pca.cor", "pca.both"), col.by.batch, col.per.group, comp2plot, legend.text, args.legend)*, *batchEstPlot (mbac, commonOmic)*, or *inner_relPlot (mbac, comp2plot = c(1,2))*.

Value

A plot is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

my_final_mbac <- MultiBaC (my_mbac,
```

```
                          test.comp = NULL, scale = FALSE,
                          center = TRUE, crossval = NULL,
                          Variability = 0.90,
                          Interaction = TRUE ,
                          showplot = FALSE,
                          showinfo = FALSE)
plot(my_final_mbac)
```

---

plot_pca *plot_pca*

Description

plot_pca

Usage

```
plot_pca(mbac, col.by.batch = TRUE, col.per.group = NULL,
  comp2plot = c(1, 2), typeP = "pca.both", legend.text = NULL,
  args.legend = NULL, ...)
```

Arguments

mbac	Object of class mbac generated by *createMbac*, *ARSyNbac*, *MultiBaC*, *genModelList*, or *batchCorrection*.
col.by.batch	Argument for PCA plots. TRUE or FALSE. If TRUE (default) samples are gruped according to the batch factor. If FALSE samples are gruped according to the experimental desing.
col.per.group	Argument for PCA plot. Indicates the color for each group defined in "groups" argument. If NULL (default) the colors are taken from a predefined pallete.
comp2plot	Indicates which components are plotted. Default is "c(1,2)", which means that component 1 is plotted in "x" axis and component 2 in "y" axis. If more components are indicated, the function will return as many plots as needed to show all the components.
typeP	"pca.cor", "pca.org" or "pca.both". If inputOmics contains original matrices, set "pca.org". However, if inputOmics contains the corrected matrices, set "pca.cor".
legend.text	a vector of text used to construct a legend for the plot.
args.legend	list of additional arguments to pass to legend(); names of the list are used as argument names. Only used if legend.text is supplied.
...	Other graphical arguments.

Value

A PCA plot is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

plot_pca(my_mbac, typeP = "pca.org")

my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
                           center = TRUE, crossval = NULL,
                           Variability = 0.90,
                           Interaction = TRUE ,
                           showplot = FALSE,
                           showinfo = FALSE)

plot_pca(my_final_mbac, typeP = "pca.cor")
```

Q2_plot *Q2_plot*

Description

Q2_plot

Usage

```
Q2_plot(mbac, ...)
```

Arguments

mbac	Object of class mbac generated by *MultiBaC* or *genModelList*.
...	Other graphical parameters

Value

Q2 plot of PLS models is displayed.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"))

my_final_mbac <- MultiBaC (my_mbac,
                           test.comp = NULL, scale = FALSE,
                           center = TRUE, crossval = NULL,
                           Variability = 0.90,
                           Interaction = TRUE ,
                           showplot = FALSE,
                           showinfo = FALSE)

Q2_plot (my_final_mbac)
```

`summary.mbac` *summary.mbac*

Description

Displays the structure and the content of the object of class mbac.

Usage

```
## S3 method for class 'mbac'
summary(object, ...)
```

Arguments

`object`	An object of class mbac.
`...`	additional arguments affecting the summary produced.

Value

Custom mbac object structure.

Examples

```
data('multiyeast')

my_mbac <- createMbac (inputOmics = list(A.rna, A.gro, B.rna, B.ribo, C.rna, C.par),
                       batchFactor = c("A", "A", "B", "B", "C", "C"),
                       experimentalDesign = list("A" =  c("Glu+", "Glu+",
                       "Glu+", "Glu-", "Glu-", "Glu-"),
                       "B" = c("Glu+", "Glu+", "Glu-", "Glu-"),
                       "C" = c("Glu+", "Glu+", "Glu-", "Glu-")),
                       omicNames = c("RNA", "GRO", "RNA", "RIBO", "RNA", "PAR"),
                       commonOmic = "RNA")
summary(my_mbac)
```

Index