

# Package ‘MultiDataSet’

October 18, 2017

**Type** Package

**Title** Implementation of the BRGE's (Bioinformatic Research Group in Epidemiology from Center for Research in Environmental Epidemiology) MultiDataSet and MethylationSet

**Version** 1.4.0

**Date** 2017-02-21

**Description** Implementation of the BRGE's (Bioinformatic Research Group in Epidemiology from Center for Research in Environmental Epidemiology) MultiDataSet and MethylationSet. Multi-

DataSet

is designed for integrating multi omics data sets and MethylationSet to contain normalized methylation data.

These package contains base classes for MEAL and rexposome packages.

**License** file LICENSE

**LazyData** TRUE

**biocViews** Software, DataRepresentation

**Depends** R (>= 3.3), Biobase

**Imports** BiocGenerics, GenomicRanges, IRanges, S4Vectors, SummarizedExperiment, methods, utils

**RoxygenNote** 6.0.1

**Suggests** MEALData, minfi, minfiData, knitr, rmarkdown, testthat, methylumi, omicade4, iClusterPlus, GEOquery, MultiAssayExperiment

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Carlos Ruiz-Arenas [aut, cre],  
Carles Hernandez-Ferrer [aut],  
Juan R. Gonzalez [aut]

**Maintainer** Carlos Ruiz-Arenas <carlos.ruiz@isglobal.org>

## R topics documented:

add_eset . . . . .	2
add_genexp . . . . .	3
add_methy . . . . .	3
add_rnaseq . . . . .	4

add_rse	5
add_se	6
add_snps	6
add_table	7
checkProbes	8
checkSamples	8
chrNumToChar	9
commonIds	9
commonSamples	10
getMs	11
mae2mds	12
mds2mae	12
MethylationSet	13
MultiDataSet	14
MultiDataSet-class	14
rowRangesElements	18
w_iclustplus	19
w_mcia	19

## Index 21

---

add_aset	<i>Method to add an eSet to MultiDataSet.</i>
----------	---

---

### Description

This method adds or overwrites a slot of a `MultiDataSet` with the content of the given `eSet`.

### Usage

```
add_aset(object, set, dataset.type, dataset.name = NULL, warnings = TRUE,
         overwrite = FALSE, GRanges)
```

### Arguments

<code>object</code>	<code>MultiDataSet</code> that will be filled.
<code>set</code>	Object derived from <code>eSet</code> to be used to fill the slot.
<code>dataset.type</code>	Character with the type of data of the omic set (e.g. expression, methylation...)
<code>dataset.name</code>	Character with the specific name for this set (NULL by default). It is useful when there are several sets of the same type (e.g. multiple expression assays)
<code>warnings</code>	Logical to indicate if warnings will be displayed.
<code>overwrite</code>	Logical to indicate if the set stored in the slot will be overwritten.
<code>GRanges</code>	<code>GenomicRanges</code> to be included in <code>rowRanges</code> slot.

### Value

A new `MultiDataSet` with a slot filled.

### See Also

[add\\_methy](#), [add\\_genexp](#), [add\\_rnaseq](#), [add\\_snps](#)

**Examples**

```
multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(10), 5))
multi <- add_eset(multi, eset, "exampledata", GRanges = NA)
```

---

add_genexp	<i>Method to add an expression microarray dataset to MultiDataSet.</i>
------------	--

---

**Description**

This method adds or overwrites the slot "expression" of an MultiDataSet with the content of the given ExpressionSet. The fData of the ExpressionSet must contain the columns chromosome, start and end.

**Usage**

```
add_genexp(object, gexpSet, ...)
```

**Arguments**

object	MultiDataSet that will be filled.
gexpSet	ExpressionSet to be used to fill the slot.
...	Arguments to be passed to add_eset.

**Value**

A new MultiDataSet with the slot "expression" filled.

**Examples**

```
multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(4), 2))
fData(eset) <- data.frame(chromosome = c("chr1", "chr2"), start = c(12414, 1234321),
  end = c(121241, 124124114), stringsAsFactors = FALSE)
multi <- add_genexp(multi, eset)
```

---

add_methy	<i>Method to add a slot of methylation to MultiDataSet.</i>
-----------	---

---

**Description**

This method adds or overwrites the slot "methylation" of an MultiDataSet with the content of the given MethylationSet or RatioSet. The fData of the input object must contain the columns chromosome and position.

**Usage**

```
add_methy(object, methySet, ...)
```

**Arguments**

object            MultiDataSet that will be filled.  
 methySet        MethylationSet or RatioSet to be used to fill the slot.  
 ...              Further arguments to be passed to add\_eset.

**Value**

A new MultiDataSet with the slot "methylation" filled.

**Examples**

```
if(require(MEALData)){
  data(mset)
  multi <- createMultiDataSet()
  multi <- add_methy(multi, mset)
}
```

---

add\_rnaseq            *Method to add an expression RNA seq dataset to MultiDataSet.*

---

**Description**

This method adds or overwrites the slot "rnaseq" of an MultiDataSet with the content of the given ExpressionSet. The fData of the ExpressionSet must contain the columns chromosome, start and end.

**Usage**

```
add_rnaseq(object, rnaSet, ...)
```

**Arguments**

object            MultiDataSet that will be filled.  
 rnaSet           ExpressionSet to be used to fill the slot.  
 ...              Arguments to be passed to add\_eset.

**Value**

A new MultiDataSet with the slot "rnaseq" filled.

**Examples**

```
multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(4), 2))
fData(eset) <- data.frame(chromosome = c("chr1", "chr2"), start = c(12414, 1234321),
  end = c(121241, 12122414), stringsAsFactors = FALSE)
multi <- add_genexp(multi, eset)
```

---

add_rse	<i>Method to add a RangedSummarizedExperiment to MultiDataSet.</i>
---------	--

---

## Description

This method adds or overwrites a slot of a `MultiDataSet` with the content of the given `RangedSummarizedExperiment`.

## Usage

```
add_rse(object, set, dataset.type, dataset.name = NULL, warnings = TRUE,
        overwrite = FALSE)
```

## Arguments

<code>object</code>	<code>MultiDataSet</code> that will be filled.
<code>set</code>	Object derived from <code>RangedSummarizedExperiment</code> to be used to fill the slot.
<code>dataset.type</code>	Character with the type of data of the omic set (e.g. expression, methylation...)
<code>dataset.name</code>	Character with the specific name for this set (NULL by default). It is useful when there are several sets of the same type (e.g. multiple expression assays)
<code>warnings</code>	Logical to indicate if warnings will be displayed.
<code>overwrite</code>	Logical to indicate if the set stored in the slot will be overwritten.

## Value

A new `MultiDataSet` with a slot filled.

## Examples

```
if (require(GenomicRanges) & require(SummarizedExperiment)){
  multi <- createMultiDataSet()
  counts <- matrix(runif(200 * 6, 1, 1e4), 200)
  rowRanges <- GRanges(rep(c("chr1", "chr2"), c(50, 150)),
                      IRanges(floor(runif(200, 1e5, 1e6)), width=100),
                      strand=sample(c("+", "-"), 200, TRUE),
                      feature_id=sprintf("ID%03d", 1:200))
  colData <- DataFrame(Treatment=rep(c("ChIP", "Input"), 3),
                      row.names=LETTERS[1:6], id = LETTERS[1:6])
  names(rowRanges) <- 1:200
  rse <- SummarizedExperiment(assays=SimpleList(counts=counts),
                             rowRanges=rowRanges, colData=colData)
  multi <- add_rse(multi, rse, "rseEx")
}
```

---

add\_se *Method to add a SummarizedExperiment to MultiDataSet.*

---

### Description

This method adds or overwrites a slot of a `MultiDataSet` with the content of the given `SummarizedExperiment`.

### Usage

```
add_se(object, set, dataset.type, dataset.name = NULL, warnings = TRUE,
        overwrite = FALSE, GRanges)
```

### Arguments

<code>object</code>	<code>MultiDataSet</code> that will be filled.
<code>set</code>	Object derived from <code>SummarizedExperiment</code> to be used to fill the slot.
<code>dataset.type</code>	Character with the type of data of the omic set (e.g. expression, methylation...)
<code>dataset.name</code>	Character with the specific name for this set (NULL by default). It is useful when there are several sets of the same type (e.g. multiple expression assays)
<code>warnings</code>	Logical to indicate if warnings will be displayed.
<code>overwrite</code>	Logical to indicate if the set stored in the slot will be overwritten.
<code>GRanges</code>	<code>GenomicRanges</code> to be included in <code>rowRanges</code> slot.

### Value

A new `MultiDataSet` with a slot filled.

### Examples

```
multi <- createMultiDataSet()
se <- SummarizedExperiment::SummarizedExperiment(matrix(runif(10), 5))
multi <- add_se(multi, se, "exampledata", GRanges = NA)
```

---

add\_snps *Method to add a slot of SNPs to MultiDataSet.*

---

### Description

This method adds or overwrites the slot "snps" of an `MultiDataSet` with the content of the given `SnpSet`. The `fData` of the `SnpSet` must contain the columns `chromosome` and `position`.

### Usage

```
add_snps(object, snpSet, ...)
```

**Arguments**

object	MultiDataSet that will be filled.
snpSet	SnpSet to be used to fill the slot.
...	Arguments to be passed to add_eset.

**Value**

A new MultiDataSet with the slot "snps" filled.

**Examples**

```
multi <- createMultiDataSet()
geno <- matrix(c(3,1,2,1), ncol = 2)
colnames(geno) <- c("VAL0156", "VAL0372")
rownames(geno) <- c("rs3115860", "SNP1-1628854")
map <- AnnotatedDataFrame(data.frame(chromosome = c("chr1", "chr2"), position = c(12414, 1234321),
  stringsAsFactors = FALSE))
rownames(map) <- rownames(geno)
snpSet <- new("SnpSet", call = geno, featureData = map)
pheno <- data.frame(id = c("VAL0156", "VAL0372"))
rownames(pheno) <- c("VAL0156", "VAL0372")
pData(snpSet) <- pheno
multi <- add_snps(multi, snpSet)
```

---

 add\_table

---

*Method to add a matrix to MultiDataSet.*


---

**Description**

This method adds or overwrites a slot of a MultiDataSet with the content of the given matrix.

**Usage**

```
add_table(object, set, dataset.type, dataset.name = NULL, warnings = TRUE,
  overwrite = FALSE)
```

**Arguments**

object	MultiDataSet that will be filled.
set	matrix used to fill the slot.
dataset.type	Character with the type of data
dataset.name	Character with the specific name for this set (NULL by default). It is useful when there are several sets of the same type.
warnings	Logical to indicate if warnings will be displayed.
overwrite	Logical to indicate if the set stored in the slot will be overwritten.

**Value**

A new MultiDataSet with a slot filled.

**Examples**

```
multi <- createMultiDataSet()
mat <- matrix(runif(12), nrow = 3)
colnames(mat) <- paste0("S", 1:4)
rownames(mat) <- paste0("F", 1:3)
multi <- add_table(multi, mat, "exampledata")
```

---

checkProbes	<i>Filter MethylationSet probes</i>
-------------	-------------------------------------

---

**Description**

This function selects probes present in the annotation matrix. Probes without annotation and annotation values without beta values are discarded.

**Usage**

```
checkProbes(object)
```

**Arguments**

object	MethylationSet
--------	----------------

**Value**

MethylationSet containing the common samples.

**Examples**

```
if(require(MEALData)){
  data(mset)
  checkProbes(mset)
}
```

---

checkSamples	<i>Modify a MethylationSet to only contain common samples</i>
--------------	---

---

**Description**

This function removes samples that have beta values but no phenotypes and vice versa. If snps object is present, only samples present in the three set are retained.

**Usage**

```
checkSamples(object)
```

**Arguments**

object	MethylationSet
--------	----------------



**Value**

MethylationSet containing the common samples.

**Examples**

```
if(require(MEALData)){
  data(mset)
  checkSamples(mset)
}
```

---

 chrNumToChar

*Convert chr numbers to chr strings*


---

**Description**

Given a vector of number representing the chromosomes, convert them to string (e.g 1 to chr1). 23 is consider chrX, 24 is chrY, 25 is chrXY (probes shared between chromosomes X and Y) and 26 is chrMT.

**Usage**

```
chrNumToChar(vector)
```

**Arguments**

vector            The vector with the chromosome numbers

**Value**

A vector with the chromosomes in string format.

**Examples**

```
chromosomes <- c(1, 3, 4, 23, 15)
stringChrs <- chrNumToChar(chromosomes)
stringChrs
```

---

 commonIds

*Get the name of the ids common to all datasets*


---

**Description**

Get the name of the ids common to all datasets

**Usage**

```
commonIds(object)
```

**Arguments**

object            MultiDataSet that will be filtered.

**Value**

Character vector with the common ids.

**Examples**

```
multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(9), ncol = 3))
fData(eset) <- data.frame(chromosome = c("chr1", "chr1", "chr1"),
  start = c(1, 5, 10), end = c(4, 6, 14),
  stringsAsFactors = FALSE)
sampleNames(eset) <- c("S1", "S2", "S3")
pData(eset) <- data.frame(id = c("S1", "S2", "S3"))
rownames(pData(eset)) <- c("S1", "S2", "S3")
multi <- add_genexp(multi, eset, dataset.name = "g1")
eset <- new("ExpressionSet", exprs = matrix(runif(8), ncol = 2))
fData(eset) <- data.frame(chromosome = c("chr1", "chr1", "chr1", "chr1"),
  start = c(1, 14, 25, 104), end = c(11, 16, 28, 115),
  stringsAsFactors = FALSE)
sampleNames(eset) <- c("S1", "G2")
pData(eset) <- data.frame(id = c("S1", "G2"))
rownames(pData(eset)) <- c("S1", "G2")

multi <- add_genexp(multi, eset, dataset.name="g2")
commonIds(multi)
```

---

commonSamples

*Method to select samples that are present in all datasets.*

---

**Description**

This method subsets the datasets to only contain the samples that are in all datasets. All sets will have the samples in the same order, taking into account that there can be duplicates.

**Usage**

```
commonSamples(object, unify.names = FALSE)
```

**Arguments**

object	MultiDataSet that will be filtered.
unify.names	Logical indicating if sample names of the sets should be unified.

**Details**

If unify.names is TRUE, the sample names of the sets will be unified using the id column of phenodata. This option is only possible when there are no duplicated ids.

**Value**

A new MultiDataSet with only the common samples.

**Examples**

```

multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(9), ncol = 3))
fData(eset) <- data.frame(chromosome = c("chr1", "chr1", "chr1"),
                          start = c(1, 5, 10), end = c(4, 6, 14),
                          stringsAsFactors = FALSE)
sampleNames(eset) <- c("S1", "S2", "S3")
pData(eset) <- data.frame(id = c("S1", "S2", "S3"))
rownames(pData(eset)) <- c("S1", "S2", "S3")
multi <- add_genexp(multi, eset, dataset.name = "g1")
eset <- new("ExpressionSet", exprs = matrix(runif(8), ncol = 2))
fData(eset) <- data.frame(chromosome = c("chr1", "chr1", "chr1", "chr1"),
                          start = c(1, 14, 25, 104), end = c(11, 16, 28, 115),
                          stringsAsFactors = FALSE)
sampleNames(eset) <- c("S1", "G2")
pData(eset) <- data.frame(id = c("S1", "G2"))
rownames(pData(eset)) <- c("S1", "G2")

multi <- add_genexp(multi, eset, dataset.name="g2")
commonSamples(multi)

```

getMs

*Transforms beta values to M-values***Description**

Given a MethylationSet or a AnalysisResults returns the matrix of M values using a logit2 transformation. Betas equal to 0 will be transformed to threshold and betas equal to 1, to 1 - threshold.

**Usage**

```
getMs(object, threshold = 1e-04)
```

**Arguments**

object	MethylationSet or AnalysisResults
threshold	Numeric with the threshold to avoid 0s and 1s.

**Value**

Matrix with the M values.

**Examples**

```

if(require(MEALData)){
  data(mset)
  Ms <- getMs(mset)
}

```

---

mae2mds	<i>Convert a MultiAssayExperiment to a MultiDataSet</i>
---------	---

---

**Description**

This function creates a MultiDataSet using the data of a MultiAssayExperiment.

**Usage**

```
mae2mds(MAE, warnings = TRUE)
```

**Arguments**

MAE	a MultiAssayExperiment
warnings	Logical to indicate if warnings will be displayed.

**Value**

MultiDataSet with the of the incoming MultiAssayExperiment.

---

mds2mae	<i>Convert a MultiDataSet to a MultiAssayExperiment</i>
---------	---

---

**Description**

This function creates a MultiAssayExperiment using the data of a MultiDataSet.

**Usage**

```
mds2mae(MDS)
```

**Arguments**

MDS	a MultiDataSet
-----	----------------

**Value**

MultiAssayExperiment with the of the incoming MultiDataSet.

---

MethylationSet	<i>MethylationSet instances</i>
----------------	---------------------------------

---

**Description**

Container with the data needed to perform methylation analysis. MethylationSet inherits from eSet and contains meth matrix as assay data member.

**Usage**

```
methylationSet(betas, phenotypes, annotationDataFrame, annoString = "custom")
```

```
## S4 method for signature 'MethylationSet'
betas(object)
```

```
## S4 method for signature 'MethylationSet'
getMs(object, threshold = 1e-04)
```

```
## S4 method for signature 'MethylationSet'
checkProbes(object)
```

```
## S4 method for signature 'MethylationSet'
checkSamples(object)
```

**Arguments**

betas	Matrix of beta values
phenotypes	Data.frame or AnnotatedDataFrame with the phenotypes
annotationDataFrame	Data.frame or AnnotatedDataFrame with the annotation of the methylation sites.
annoString	Character with the name of the annotation used.
object	MethylationSet
threshold	Numeric with the threshold to avoid 0s and 1s.

**Details**

FeatureData, which contains annotation data, is required to perform any of the analysis.

**Value**

MethylationSet

**Methods (by generic)**

- betas: Get beta matrix
- getMs: Get Ms values
- checkProbes: Filter probes with annotation
- checkSamples: Modify a MethylationSet to only contain common samples

**Slots**

assayData Contains matrices with equal dimensions, and with column number equal to nrow(phenoData). assayData must contain a matrix meth with rows representing features (e.g., methylation probes sets) and columns representing samples.

phenoData See [eSet](#)

annotation See [eSet](#)

featureData See [eSet](#). fData should contain at least chromosome and positions columns.

**Examples**

```
showClass("MethylationSet")
```

---

MultiDataSet

*MultiDataSet: Implementation of the BRGE's basic classes*

---

**Description**

Implementation of the BRGE's (Bioinformatic Research Group in Epidemiology from Center for Research in Environmental Epidemiology) MultiDataSet and MethylationSet. MultiDataSet is designed for integrating multi omics data sets and MethylationSet to contain normalized methylation data. MultiDataSet for integrating multi omics data sets

**See Also**

[MultiDataSet](#)

---

MultiDataSet-class

*MultiDataSet instances*

---

**Description**

The class MultiDataSet is a superior class to store multiple datasets in form of triplets (assayData-phenoData-featureData). The datasets must be eSet or SummarizedExperiment.

**Usage**

```
## S4 method for signature 'MultiDataSet,eSet'
add_eset(object, set, dataset.type,
  dataset.name = NULL, warnings = TRUE, overwrite = FALSE, GRanges)

## S4 method for signature 'MultiDataSet,ExpressionSet'
add_genexp(object, gexpSet, ...)

## S4 method for signature 'MultiDataSet,ExpressionSet'
add_rnaseq(object, rnaSet, ...)

## S4 method for signature 'MultiDataSet,MethylationSet'
add_methy(object, methySet, ...)
```

```
## S4 method for signature 'MultiDataSet,RatioSet'  
add_methy(object, methySet, ...)  
  
## S4 method for signature 'MultiDataSet,RangedSummarizedExperiment'  
add_rse(object, set,  
  dataset.type, dataset.name = NULL, warnings = TRUE, overwrite = FALSE)  
  
## S4 method for signature 'MultiDataSet,SummarizedExperiment'  
add_se(object, set, dataset.type,  
  dataset.name = NULL, warnings = TRUE, overwrite = FALSE, GRanges)  
  
## S4 method for signature 'MultiDataSet,SnpSet'  
add_snps(object, snpSet, ...)  
  
## S4 method for signature 'MultiDataSet,matrix'  
add_table(object, set, dataset.type,  
  dataset.name = NULL, warnings = TRUE, overwrite = FALSE)  
  
## S4 method for signature 'MultiDataSet'  
as.list(x)  
  
## S4 method for signature 'MultiDataSet'  
commonIds(object)  
  
## S4 method for signature 'MultiDataSet'  
commonSamples(object, unify.names = FALSE)  
  
createMultiDataSet()  
  
## S4 method for signature 'MultiDataSet'  
dims(object)  
  
## S4 method for signature 'MultiDataSet'  
w_iclusterplus(object, commonSamples = TRUE, ...)  
  
## S4 method for signature 'MultiDataSet'  
length(x)  
  
## S4 method for signature 'MultiDataSet'  
w_mcia(object, ...)  
  
## S4 method for signature 'MultiDataSet'  
names(x)  
  
## S4 method for signature 'MultiDataSet'  
ncols(object)  
  
## S4 method for signature 'MultiDataSet'  
nrows(object)  
  
## S4 method for signature 'MultiDataSet'
```

```

rowRangesElements(object)

## S4 method for signature 'MultiDataSet'
sampleNames(object)

## S4 method for signature 'MultiDataSet'
assayData(object)

## S4 method for signature 'MultiDataSet'
fData(object)

## S4 method for signature 'MultiDataSet'
featureData(object)

## S4 method for signature 'MultiDataSet'
pData(object)

## S4 method for signature 'MultiDataSet'
phenoData(object)

## S4 method for signature 'MultiDataSet'
rowRanges(x)

## S4 method for signature 'MultiDataSet,ANY,ANY'
x[[i]]

## S4 method for signature 'MultiDataSet,ANY,ANY,ANY'
x[i, j, k, ..., drop = FALSE]

## S4 method for signature 'MultiDataSet'
subset(x, feat, phe, warnings = TRUE, keep = TRUE)

```

### Arguments

object	MultiDataSet
set	Object derived from eSet to be used to fill the slot.
dataset.type	Character with the type of data of the omic set (e.g. expression, methylation...)
dataset.name	Character with the specific name for this set (NULL by default). It is useful when there
warnings	Logical to indicate if warnings will be displayed.
overwrite	Logical to indicate if the set stored in the slot will be overwritten.
GRanges	GenomicRanges to be included in rowRanges slot.
gexpSet	ExpressionSet to be used to fill the slot.
...	Further arguments passed to add_eset.
rnaSet	ExpressionSet to be used to fill the slot.
methySet	MethylationSet to be used to fill the slot.
snpSet	SnpsSet to be used to fill the slot.
x	MultiDataSet
unify.names	Logical indicating if sample names of the sets should be unified.



commonSamples	Logical to indicate if common samples are selected
i	Character corresponding to selected sample names. They should match the id column of phenoData.
j	Character with the name of the selected tables.
k	GenomicRange used to filter the features.
drop	If TRUE, sets with no samples or features will be discarded
feat	Logical expression indicating features to keep
phe	Logical expression indicating the phenotype of the samples to keep
keep	If FALSE, sets where the expression cannot be evaluated will be discarded.

### Details

The names of the three lists (assayData, phenoData and featureData) must be the same.

### Value

MultiDataSet

MultiDataSet

### Methods (by generic)

- add\_eset: Method to add an eSet to MultiDataSet.
- add\_genexp: Method to add a slot of expression to MultiDataSet.
- add\_rnaseq: Method to add a slot of (RNASeq) expression to MultiDataSet.
- add\_methy: Method to add a slot of methylation to MultiDataSet.
- add\_methy: Method to add a slot of methylation to MultiDataSet.
- add\_rse: Method to add a RangedSummarizedExperiment to MultiDataSet.
- add\_se: Method to add a SummarizedExperiment to MultiDataSet.
- add\_snps: Method to add a slot of SNPs to MultiDataSet.
- add\_table: Method to add a matrix to MultiDataSet.
- as.list: Returns a list with the first matrix of each dataset.
- commonIds: Get the name of the ids common to all datasets
- commonSamples: Get a MultiDataSet only with the samples present in all the tables
- dims: Returns the dimensions of the sets
- w\_iclusterplus: Apply iClusterPlus clustering method to a MultiDataSet object
- length: Returns the number of sets into the object.
- w\_mcia: Apply mcia integration method to a MultiDataSet object
- names: Get the names of the slots.
- ncols: Get number of samples of each set
- nrows: Get number of features of each set
- rowRangesElements: Get the name of the datasets that have rowRanges
- sampleNames: Get sample names
- assayData: Retrieve all assay data blocks.
- fData: Retrieve information on features.

- featureData: Retrieve information on features.
- pData: Retrieve information on experimental phenotypes
- phenoData: Retrieve information on experimental phenotypes
- rowRanges: Retrieve information on feature ranges.
- [[]: Get a set from a slot
- []: Subset a MultiDataSet
- subset: Filter a subset using feature ids or phenotypes

### Slots

assayData List of assayData elements.

phenoData List of AnnotatedDataFrame containing the phenoData of each assayData.

featureData List of AnnotatedDataFrame containing the featureData of each assayData.

rowRanges List of GenomicRanges containing the rowRanges of each assayData.

return\_method List of functions used to create the original object.

### See Also

[add\\_eset](#), [add\\_rse](#)

### Examples

```
createMultiDataSet()
```

---

rowRangesElements	<i>Get the name of the datasets that have rowRanges</i>
-------------------	---

---

### Description

Get the name of the datasets that have rowRanges

### Usage

```
rowRangesElements(object)
```

### Arguments

object            MultiDataSet

### Value

Character vector with the slots that have rowRanges.

**Examples**

```

multi <- createMultiDataSet()
eset <- new("ExpressionSet", exprs = matrix(runif(10), 5))
eset2 <- new("ExpressionSet", exprs = matrix(runif(8), ncol = 2))
fData(eset2) <- data.frame(chromosome = c("chr1", "chr1", "chr1", "chr1"),
                          start = c(1, 14, 25, 104), end = c(11, 16, 28, 115),
                          stringsAsFactors = FALSE)
multi <- add_eset(multi, eset, "exampledata", GRanges = NA)
multi <- add_genexp(multi, eset2)
rowRangesElements(multi)

```

---

w\_iclusterplus

*Apply iClusterPlus clustering method to a MultiDataSet object*


---

**Description**

Method [iClusterPlus](#) is applied on a [MultiDataSet](#) object after getting the common samples along all the contained datasets.

**Usage**

```
w_iclusterplus(object, commonSamples = TRUE, ...)
```

**Arguments**

object	MultiDataSet
commonSamples	Logical to indicate if common samples are selected
...	Arguments passed to function <a href="#">iClusterPlus</a>

**Value**

A list of results from [iClusterPlus](#)

**Note**

Argument type for [iClusterPlus](#) is filled within the method.

---

w\_mcia

*Apply mcia integration method to a MultiDataSet object*


---

**Description**

Method [mcia](#) is applied on a [MultiDataSet](#) object after getting the common samples along all the contained datasets.

**Usage**

```
w_mcia(object, ...)
```

**Arguments**

object            MultiDataSet  
...                Arguments passed to function [mcia](#)

**Value**

A list of results from [mcia](#)

# Index

[ (MultiDataSet-class), 14  
[,MultiDataSet,ANY,ANY,ANY-method  
(MultiDataSet-class), 14  
[[,MultiDataSet,ANY,ANY-method  
(MultiDataSet-class), 14  
  
add\_eset, 2, 18  
add\_eset,MultiDataSet,eSet-method  
(MultiDataSet-class), 14  
add\_genexp, 2, 3  
add\_genexp,MultiDataSet,ExpressionSet-method  
(MultiDataSet-class), 14  
add\_methy, 2, 3  
add\_methy,MultiDataSet,MethylationSet-method  
(MultiDataSet-class), 14  
add\_methy,MultiDataSet,RatioSet-method  
(MultiDataSet-class), 14  
add\_rnaseq, 2, 4  
add\_rnaseq,MultiDataSet,ExpressionSet-method  
(MultiDataSet-class), 14  
add\_rse, 5, 18  
add\_rse,MultiDataSet,RangedSummarizedExperiment-method  
(MultiDataSet-class), 14  
add\_se, 6  
add\_se,MultiDataSet,SummarizedExperiment-method  
(MultiDataSet-class), 14  
add\_snps, 2, 6  
add\_snps,MultiDataSet,SnpsSet-method  
(MultiDataSet-class), 14  
add\_table, 7  
add\_table,MultiDataSet,matrix-method  
(MultiDataSet-class), 14  
as.list (MultiDataSet-class), 14  
as.list,MultiDataSet-method  
(MultiDataSet-class), 14  
assayData (MultiDataSet-class), 14  
assayData,MultiDataSet-method  
(MultiDataSet-class), 14  
  
betas (MethylationSet), 13  
betas,MethylationSet-method  
(MethylationSet), 13  
  
checkProbes, 8  
checkProbes,MethylationSet-method  
(MethylationSet), 13  
checkSamples, 8  
checkSamples,MethylationSet-method  
(MethylationSet), 13  
chrNumToChar, 9  
commonIds, 9  
commonIds,MultiDataSet-method  
(MultiDataSet-class), 14  
commonSamples, 10  
commonSamples,MultiDataSet-method  
(MultiDataSet-class), 14  
createMultiDataSet  
(MultiDataSet-class), 14  
  
dims (MultiDataSet-class), 14  
dims,MultiDataSet-method  
(MultiDataSet-class), 14  
eSet, 14  
  
fData (MultiDataSet-class), 14  
fData,MultiDataSet-method  
(MultiDataSet-class), 14  
featureData (MultiDataSet-class), 14  
featureData,MultiDataSet-method  
(MultiDataSet-class), 14  
  
getMs, 11  
getMs,MethylationSet-method  
(MethylationSet), 13  
  
iClusterPlus, 19  
  
length (MultiDataSet-class), 14  
length,MultiDataSet-method  
(MultiDataSet-class), 14  
  
mae2mds, 12  
mcia, 19, 20  
mds2mae, 12  
MethylationSet, 13  
methylationSet (MethylationSet), 13  
MethylationSet-class (MethylationSet),  
13

MethylationSet-methods  
    (MethylationSet), 13

MultiDataSet, 14, 14, 19

MultiDataSet-class, 14

MultiDataSet-methods  
    (MultiDataSet-class), 14

MultiDataSet-methods,  
    (MultiDataSet-class), 14

MultiDataSet-package (MultiDataSet), 14

  

names (MultiDataSet-class), 14

names, MultiDataSet-method  
    (MultiDataSet-class), 14

ncols (MultiDataSet-class), 14

ncols, MultiDataSet-method  
    (MultiDataSet-class), 14

nrows (MultiDataSet-class), 14

nrows, MultiDataSet-method  
    (MultiDataSet-class), 14

  

pData (MultiDataSet-class), 14

pData, MultiDataSet-method  
    (MultiDataSet-class), 14

phenoData (MultiDataSet-class), 14

phenoData, MultiDataSet-method  
    (MultiDataSet-class), 14

  

rowRanges (MultiDataSet-class), 14

rowRanges, MultiDataSet-method  
    (MultiDataSet-class), 14

rowRangesElements, 18

rowRangesElements, MultiDataSet-method  
    (MultiDataSet-class), 14

  

sampleNames, MultiDataSet-method  
    (MultiDataSet-class), 14

subset, MultiDataSet-method  
    (MultiDataSet-class), 14

  

w\_iclusterplus, 19

w\_iclusterplus, MultiDataSet-method  
    (MultiDataSet-class), 14

w\_mcia, 19

w\_mcia, MultiDataSet-method  
    (MultiDataSet-class), 14