

# Package ‘MOSClip’

November 4, 2024

**Title** Multi Omics Survival Clip

**Version** 1.1.0

**Depends** R (>= 4.4.0)

**Description** Topological pathway analysis tool able to integrate multi-omics data. It finds survival-associated modules or significant modules for two-class analysis. This tool have two main methods: pathway tests and module tests. The latter method allows the user to dig inside the pathways itself.

**License** AGPL-3

**Imports** MultiAssayExperiment, methods, survminer, graph, graphite, AnnotationDbi, checkmate, ggplot2, gridExtra, igraph, pheatmap, survival, RColorBrewer, SuperExactTest, reshape, NbClust, S4Vectors, grDevices, graphics, stats, utils, ComplexHeatmap, FactoMineR, circlize, corpcor, coxrobust, elasticnet, gRbase, ggplotify, qqgraph, org.Hs.eg.db, Matrix

**Suggests** RUnit, BiocGenerics, MASS, BiocStyle, knitr, EDASeq, rmarkdown, kableExtra, testthat (>= 3.0.0)

**Encoding** UTF-8

**LazyData** false

**biocViews** Software, StatisticalMethod, GraphAndNetwork, Survival, Regression, DimensionReduction, Pathways, Reactome

**BugReports** <https://github.com/CaluraLab/MOSClip/issues>

**URL** <https://github.com/CaluraLab/MOSClip/>

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/MOSClip>

**git\_branch** devel

**git\_last\_commit** 412bda8

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-11-03

**Author** Paolo Martini [aut, cre] (ORCID: <https://orcid.org/0000-0002-0146-1031>),  
 Anna Bortolato [aut] (ORCID: <https://orcid.org/0009-0009-9327-6084>),  
 Anna Tanada [aut] (ORCID: <https://orcid.org/0000-0003-3224-0538>),  
 Enrica Calura [aut] (ORCID: <https://orcid.org/0000-0001-8463-2432>),  
 Stefania Pirrotta [aut] (ORCID: <https://orcid.org/0009-0004-0030-217X>),  
 Federico Agostinis [aut]

**Maintainer** Paolo Martini <paolo.martini@unibs.it>

## Contents

annotatePathwayToFather . . . . .	3
availableOmicMethods . . . . .	5
checkOrder . . . . .	5
compPCs . . . . .	6
computeFreqs . . . . .	7
computeOmicsIntersections . . . . .	8
computePCs . . . . .	9
convertPathway . . . . .	10
createCoxObj . . . . .	10
createDataModule . . . . .	11
createMOMView . . . . .	11
downloadPathwayRelationFromReactome . . . . .	12
estimateExprCov . . . . .	12
extractCliquesFromDag . . . . .	13
extractSummaryFromBinary . . . . .	13
extractSummaryFromCluster . . . . .	14
extractSummaryFromNumberOfEvents . . . . .	15
extractSummaryFromPCA . . . . .	16
getPathFathers . . . . .	17
glmTest . . . . .	17
guessInvolvement . . . . .	18
guessInvolvementPathway . . . . .	19
id2name . . . . .	20
makeOmics . . . . .	20
makePositiveDefinite . . . . .	22
mapPathwaysIDfromGraphite . . . . .	22
minOrNA . . . . .	23
MOSClip . . . . .	23
multiOmics . . . . .	25
MultiOmicsModules-class . . . . .	25
MultiOmicsPathway-class . . . . .	26
multiOmicsSurvivalModuleTest . . . . .	27
multiOmicsSurvivalPathwayTest . . . . .	28
multiOmicsTopo . . . . .	29

multiOmicsTwoClassModuleTest . . . . .	30
multiOmicsTwoClassPathwayTest . . . . .	31
multiPathwayModuleReport . . . . .	32
multiPathwayReport . . . . .	33
Omics-class . . . . .	34
ovarianDataset . . . . .	35
plotFrequencies . . . . .	35
plotModuleHeat . . . . .	36
plotModuleInGraph . . . . .	38
plotModuleKM . . . . .	39
plotModuleReport . . . . .	41
plotMultiPathwayReport . . . . .	42
plotPathwayHeat . . . . .	43
plotPathwayKM . . . . .	45
pvalueSummary . . . . .	46
reactSmall . . . . .	47
removeSelfLoops . . . . .	47
resamplingModulesSurvival . . . . .	48
resamplingModulesTwoClass . . . . .	49
runSupertest . . . . .	50
selectStablePathwaysModules . . . . .	52
showModule . . . . .	53
showMOSpalette . . . . .	54
showOmics . . . . .	54
showPathway . . . . .	55
sparseCompPCs . . . . .	56
stripModulesFromPathways . . . . .	56
summarizeInCluster . . . . .	57
summarizeOmicsResByMinPvalue . . . . .	58
summarizeToBinaryDirectionalEvents . . . . .	59
summarizeToBinaryEvents . . . . .	60
summarizeToNumberOfDirectionalEvents . . . . .	61
summarizeToNumberOfEvents . . . . .	62
summarizeWithPca . . . . .	63
survivalcox . . . . .	64
survivalcoxr . . . . .	64
topoCompPCs . . . . .	65

**Index**

**Description**

Given the hierarchy of the pathways, this formula finds the fathers of the respective pathway (e.g. pathway: 'PI3K Cascade'; father: 'Signaling Pathways'). This function is necessary for calculating the contribution of different omics to survival prediction in different biological processes, grouping the pathways by hierarchy.

**Usage**

```
annotatePathwayToFather(pathways, graphiteDB, hierarchy)
```

**Arguments**

pathways	vector of pathway names
graphiteDB	graphite DB object (e.g. an object containing all reactome pathways)
hierarchy	a graph object with the pathway hierarchy

**Value**

a vector of the pathway fathers' names

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_list <- lapply(reactSmall[1:2], function(g) {
  fcl <- multiOmicsSurvivalModuleTest(multiOmics, g,
    survFormula = "Surv(days, status) ~",
    autoCompleteFormula = TRUE,
    useTheseGenes = genesToUse
  )
  fcl
})

moduleSummary <- multiPathwayModuleReport(MOM_list)

pathHierarchy <- downloadPathwayRelationFromReactome()
pathHierarchyGraph <- igraph::graph.data.frame(
  d = pathHierarchy,
  directed = TRUE
)

omicsClasses2pathways <- computeOmicsIntersections(
  moduleSummary,
  pvalueThr = 1, zscoreThr = 1,
  excludeColumns = c("pathway", "module")
)

omicsClasses2pathways <- lapply(
```

```

    omicsClasses2pathways,
    stripModulesFromPathways
  )

# This step requires to download the whole reactome graph, which usually
# takes a lot of time.
# reactome <- graphite::pathways('hsapiens', 'reactome')
# reactome <- graphite::convertIdentifiers(reactome, 'entrez')
# omicsClasses2fathers <- lapply(omicsClasses2pathways,
#                               annotatePathwayToFather,
#                               graphiteDB = reactome,
#                               hierarchy = pathHierarchyGraph)

```

---

availableOmicMethods    *Get available Omics Summarizing Methods*

---

### Description

Gives a vector of the available methods to summarize omics.

### Usage

```
availableOmicMethods()
```

### Value

character vector with the implemented methods.

### Examples

```
availableOmicMethods()
```

---

checkOrder                    *Check if all the list object have the same order of pathway module*

---

### Description

For internal use only

For internal use only

For internal use only

For internal use only

For internal use only

Prepare subset of patients for permutations

**Usage**

```

checkOrder(li)

resolveAndOrder(li)

mergeCol(li, col = "PC1", resolve = FALSE)

filterExpr(exp, samples)

filterMultiOmicsForSamples(MO, samples)

preparePerms(fullMultiOmics, nperm = 100, nPatients = 3)

```

**Arguments**

li	a list of summaries
col	the column to merge
resolve	weather to resolve the issues
exp	a matrix
samples	the vector of samples to select
MO	a multiOmic object
fullMultiOmics	a multiOmic object
nperm	number of permutations
nPatients	number of patients to remove for resampling

**Value**

a matrix  
a filtered matrix  
a filtered MultiOmics objects  
list of sampled patients for each permutation

---

compPCs

*Regular PCA*


---

**Description**

Regular PCA

**Usage**

```
compPCs(exp, shrink, k)
```

**Arguments**

exp	a matrix
shrink	logical, whether to shrink or not.
k	the number of components to use

**Value**

a list with the following elements:

x	the computed PCs
sdev	the standard deviation captured by the PCs
loadings	the loadings

---

computeFreqs	<i>Compute Frequencies in a Named List</i>
--------------	--

---

**Description**

Compute frequencies in a named list. This function is necessary for [plotFrequencies](#), in which it will calculate the frequency of each pathway father for every omics intersection.

**Usage**

```
computeFreqs(elementsIntersections)
```

**Arguments**

elementsIntersections	a named list
-----------------------	--------------

**Value**

a data.frame of the frequencies

**Examples**

```
omicsIntersection <- list(
  "exp;met" = c("PathwayA", "PathwayB", "PathwayC"),
  "exp;mut" = c("PathwayA", "PathwayC"),
  "cnv;mut" = c("PathwayB")
)
freqDf <- computeFreqs(omicsIntersection)
```

---

```
computeOmicsIntersections
      Compute Omics Intersections
```

---

## Description

Finds the modules that have any intersection among the available omics

## Usage

```
computeOmicsIntersections(
  multiPathwayReportData,
  pvalueThr = 0.05,
  zscoreThr = 0.05,
  resampligThr = NULL,
  excludeColumns = NULL
)
```

## Arguments

multiPathwayReportData	data.frame, the output of the <a href="#">multiPathwayReport</a> or <a href="#">multiPathwayModuleReport</a> functions.
pvalueThr	numeric value. Overall pvalue cut-off to be used
zscoreThr	numeric value. Covariates coefficient cut-off to be used.
resampligThr	numeric value. Filters the modules according to the number of success in the resampling procedure, takes only the modules above this threshold.
excludeColumns	a vector of characters listing the columns of multiPathwayReportData object to be excluded by the analysis. In the case multiPathwayReportData derives from <a href="#">multiPathwayModuleReport</a> you should set excludeColumns = c('pathway', 'module').

## Value

a list of pathway modules present for every intersection of omics present

## Examples

```
df <- data.frame(
  pvalue = c(0.06, 0.04, 0.04, 0.03, 0.02),
  cnv = c(0.07, 0.03, 0.02, 0.04, 0.01),
  mut = c(0.08, 0.02, 0.01, 0.04, 0.04),
  row.names = c(
    "PathwayA", "PathwayB", "PathwayC",
    "PathwayD", "PathwayE"
  )
)
```



```
omicsClasses2Pathways <- computeOmicsIntersections(df,  
  pvalueThr = 0.1,  
  zscoreThr = 0.1  
)
```

---

computePCs

*compute PCs.*

---

## Description

For internal use only. Performs Principal Component analysis.

## Usage

```
computePCs(  
  exp,  
  shrink = FALSE,  
  method = c("regular", "topological", "sparse"),  
  cliques = NULL,  
  maxPCs = 3  
)
```

## Arguments

exp	a matrix
shrink	logical, whether to shrink or not.
method	one of 'regular', 'topological' and 'sparse'
cliques	the pathway topology summarized in a list of cliques
maxPCs	the maximum number of PCs to consider

## Details

Three methods are implemented:

- regular: a regular PCA ('prcomp')
- topological: PCA using a pathway topology.
- sparse: sparse PCA analysis implemented by 'elasticnet'

## Value

a list with the following elements:

x	the computed PCs
sdev	the standard deviation captured by the PCs
loadings	the loadings

---

convertPathway	<i>A generic function to convert pathway</i>
----------------	--

---

**Description**

A generic function to convert pathway

**Usage**

```
convertPathway(graph, useTheseGenes)
```

**Arguments**

graph	a graphNEL object
useTheseGenes	list of genes to be used

**Value**

NULL. No value is returned

---

createCoxObj	<i>Create Cox Object</i>
--------------	--------------------------

---

**Description**

Create the coxObj from the covariates used in the test

**Usage**

```
createCoxObj(colData, moView)
```

**Arguments**

colData	colData from multiOmic object
moView	modulesView or pathView from multiOmicsModules or multiOmicsPathway object

**Value**

data.frame, samples in the rows, covariates in the columns

---

createDataModule	<i>Create Data Module</i>
------------------	---------------------------

---

**Description**

Extract sub-matrix for the genes of a module or pathway from data matrix of a specific omic

**Usage**

```
createDataModule(omic, multiOmicObj)
```

**Arguments**

omic	modulesView or pathView object
multiOmicObj	object of class 'Omics'

**Value**

matrix, genes in the rows, samples in the columns

---

createMOMView	<i>Create the list of covariates that are going to be tested</i>
---------------	--

---

**Description**

Create the list of covariates that are going to be tested

**Usage**

```
createMOMView(omicsObj, genes)
```

**Arguments**

omicsObj	Omics class object
genes	genes of the clique

**Value**

list with 1 reduced representation of the omics 2 sdev 3 loadings or eigenvector 4 usedGenes 5 method 6 namesCov 7 omicName

---

`downloadPathwayRelationFromReactome`*Download Reactome Pathway Relations*

---

**Description**

Download Pathway Relations from Reactome. The file is retrieved from the [url](#)

**Usage**

```
downloadPathwayRelationFromReactome(url = NULL, speciesAbbr = "HSA")
```

**Arguments**

<code>url</code>	the location of the file. Can be local. If NULL pick the package reactome file.
<code>speciesAbbr</code>	species acronym

**Value**

A data frame with 2 columns:

<code>parent</code>	The Reactome pathway ID of the parent pathway.
<code>child</code>	The Reactome pathway ID of the child pathway.

**Examples**

```
downloadPathwayRelationFromReactome()
```

---

`estimateExprCov`*Estimate Single Covariance Matrix*

---

**Description**

For internal use only. Estimate Covariance from one matrix

**Usage**

```
estimateExprCov(expr, shrink)
```

**Arguments**

<code>expr</code>	a numeric matrix
<code>shrink</code>	logical wheter to shrink the matrix

**Value**

a covariance matrix



**Value**

Meant for internal use only. The summary for omic summarized using binary events.

sigModule	the original data for significant features
discrete	the discrete version of the significant covariates converted (when needed) into the discrete version
subset	data.frame(row.names=names(topGenes), cov=sum binary events)
covsConsidered	the name of the considered omic

---

extractSummaryFromCluster

*Extract Summary Cluster from MultiOmics Objects*

---

**Description**

Given an omic summarized by 'summarizeInCluster' extract the most important features.

**Usage**

```
extractSummaryFromCluster(omic, multiOmicObj, n = 3)
```

**Arguments**

omic	a summarized omic
multiOmicObj	Omics object
n	maximum number of features to retrieve

**Value**

summary for omic summarized using clusters

sigModule	the original data for significant features
discrete	the discrete version of the significant covariates converted (when needed) into the discrete version
subset	data.frame(row.names=names(topGenes), metClust=topGenes)
pvalues	Kruskal Wallis pvalues of the selected features
covsConsidered	the name of the considered omic

---

 extractSummaryFromNumberOfEvents

*Extract Summary Binary from MultiOmics Objects*


---

### Description

Given an omic summarized by 'summarizeToNumberOfEvents' extract the most important features.

### Usage

```
extractSummaryFromNumberOfEvents(
  omic,
  multiOmicObj,
  moduleCox,
  analysis,
  n = 3,
  minprop = 0.1,
  labels = c("few", "many")
)
```

### Arguments

omic	a summarized omic
multiOmicObj	Omics object
moduleCox	the coxObj of the interesting module
analysis	two-class or survival type
n	maximum number of features to retrieve
minprop	the minimal proportion of cutp
labels	the category labels

### Value

Meant for internal use only. The summary for omic summarized using counting of events.

sigModule	the original data for significant features
discrete	the discrete version of the significant covariates converted (when needed) into the discrete version
subset	data.frame(row.names=names(topGenes), covariates=covariate)
covsConsidered	the name of the considered omic

---

extractSummaryFromPCA *Extract Summary PCA from MultiOmics Objects*

---

### Description

Given an omic summarized by 'summarizeWithPca' extract the most important features.

### Usage

```
extractSummaryFromPCA(
  omic,
  multiOmicObj,
  moduleCox,
  analysis,
  loadThr = 0.6,
  atleast = 1,
  minprop = 0.1
)
```

### Arguments

omic	a summarized omic
multiOmicObj	Omics object
moduleCox	the coxObj of the interesting module
analysis	two-class or survival type
loadThr	the thr value to select the most influent features according to the loading
atleast	the minimum number of gene to retrieve
minprop	the minimal proportion of cutp

### Value

summary for omic summarized using pca	
sigModule	the original data for significant features
discrete	the discrete version of the significant covariates converted (when needed) into the discrete version
subset	data.frame(row.names=names(topGenes), covariate)
covsConsidered	the name of the considered omic



---

getPathFathers	<i>Retrieves pathways relatives</i>
----------------	-------------------------------------

---

**Description**

For internal use only. Retrieves relatives given a pathway id.

**Usage**

```
getPathFathers(pathway, hierarchyGraph, ord = 3, plot = FALSE)
```

**Arguments**

pathway	a pathway id
hierarchyGraph	a igraph with pathway hierarchy
ord	how far you need to go backward
plot	plot relatives. For checking purpose

**Details**

Pathway Hierarchy is needed as igraph object.

**Value**

a character vector with the relatives

---

glmTest	<i>Two-classes glm test.</i>
---------	------------------------------

---

**Description**

Two-classes glm test.

**Usage**

```
glmTest(data, fullModelFormula, nullModelFormula)
```

**Arguments**

data	data
fullModelFormula	complete model
nullModelFormula	null model formula

**Value**

Two-classes glm test results

---

guessInvolvement	<i>Guess the most influent features from MultiOmics Survival or Two-class results.</i>
------------------	--

---

**Description**

Given a pathway analyzed by `multiOmicsModuleSurvivalTest` or `multiOmicsTwoClassModuleTest`, it retrieves for each omic the most influent features.

**Usage**

```
guessInvolvement(
  pathway,
  moduleNumber,
  loadThr = 0.6,
  n = 3,
  atleast = 1,
  min_prop_pca = 0.1,
  min_prop_events = 0.1,
  ...
)
```

**Arguments**

<code>pathway</code>	MultiOmicsModules object from a pathway
<code>moduleNumber</code>	the module number
<code>loadThr</code>	the loading threshold to select genes (PCA only)
<code>n</code>	the maximum number of genes to retrieve (cluster and binary only)
<code>atleast</code>	the minimum number of features to select (PCA only)
<code>min_prop_pca</code>	the minimal proportion to compute the PCA classes
<code>min_prop_events</code>	the minimal proportion to compute the event classes
<code>...</code>	additional arguments passed to get function

**Value**

a list. Each item of the list corresponds to an omic that is summarized with the specific 'extract-Summary' functions. Each item is the summary for an omic summarized using the setted method: pvalues are present only for cluster method.

---

guessInvolvementPathway

*Guess the most influent features from MultiOmics Survival or Two-class results.*

---

### Description

Given a pathway analyzed by `multiOmicsSurvivalPathwayTest` or `multiOmicsTwoClassPathwayTest`, it retrieves for each omic the most influent features.

### Usage

```
guessInvolvementPathway(
  pathway,
  loadThr = 0.6,
  n = 3,
  atleast = 1,
  min_prop_pca = 0.1,
  min_prop_events = 0.1,
  ...
)
```

### Arguments

<code>pathway</code>	MultiOmicsModules object from a pathway
<code>loadThr</code>	the loading threshold to select genes (PCA only)
<code>n</code>	the maximum number of genes to retrieve (cluster and binary only)
<code>atleast</code>	the minimum number of features to select (PCA only)
<code>min_prop_pca</code>	the minimal proportion to compute the PCA classes
<code>min_prop_events</code>	the minimal proportion to compute the event classes
<code>...</code>	additional arguments passed to get function

### Value

a list. Each item of the list corresponds to an omic that is summarized with the specific 'extract-Summary' functions. Each item is the summary for an omic summarized using the setted method: pvalues are present only for cluster method.

---

id2name	<i>Convert id to pathway name</i>
---------	-----------------------------------

---

**Description**

For internal use only. Retrieves name from pathway id.

**Usage**

```
id2name(idList, namedVect)
```

**Arguments**

idList	a list of pathway id
namedVect	a named vector

**Details**

You must provide a namedVect to be used as translator.

**Value**

a character vector with the names

---

makeOmics	<i>Omics class initializer function</i>
-----------	---

---

**Description**

makeOmics creates the Omics object necessary to perform most of the analyses of this package. It contains all the omics data in the format of a ExperimentList, the clinical data, and all the information necessary for the dimensionality reduction step.

**Usage**

```
makeOmics(  
  experiments = ExperimentList(),  
  colData = S4Vectors::DataFrame(),  
  sampleMap = S4Vectors::DataFrame(assay = factor(), primary = character(), colname =  
    character()),  
  metadata = list(),  
  drops = list(),  
  modelInfo = character(),  
  specificArgs = list()  
)
```

**Arguments**

experiments	A list or <a href="#">ExperimentList</a> of all combined experiments
colData	A <a href="#">DataFrame</a> or data.frame of characteristics for all biological units
sampleMap	A DataFrame or data.frame of assay names, sample identifiers, and colname samples
metadata	An optional argument of 'ANY' class (usually list) for content describing the experiments
drops	A list of unmatched information (included after subsetting)
modelInfo	A list with length equal to length(data) that are modelInfo to process each dataset
specificArgs	a list with length equal to length(data) to set additional parameters specific of the modelInfo

**Value**

an Omics class object

**Examples**

```

data(ovarianDataset)

myColData <- data.frame(
  status = sample(c(0, 1), 50, replace = TRUE),
  days = sample(c(0, 500), 50, replace = TRUE),
  row.names = colnames(ovarianDataset$exp)
)

myOmicsObj <- makeOmics(
  experiments = ovarianDataset,
  colData = myColData,
  modelInfo = c(
    "summarizeWithPca",
    "summarizeInCluster",
    "summarizeToNumberOfEvents",
    "summarizeToNumberOfDirectionalEvents"
  ),
  specificArgs = list(
    pcaArgs = list(
      name = "exp", shrink = "FALSE",
      method = "sparse", maxPCs = 3
    ),
    clusterArgs = list(
      name = "met",
      max_cluster_number = 3
    ),
    countEvent = list(name = "mut", min_prop = 0.05),
    cnvAgv = list(name = "cnv", min_prop = 0.05)
  )
)

```

---

makePositiveDefinite *Make positive and definite covariance matrix*

---

**Description**

Make positive and definite covariance matrix

**Usage**

```
makePositiveDefinite(m1, m2 = NULL, m3 = NULL, threshold = 0.1)
```

**Arguments**

m1	matrix 1
m2	matrix 2
m3	matrix 3
threshold	threshold of difference

**Value**

list with

m1	the matrix m1 positive and definite
m2	the matrix m2 positive and definite
m3	the matrix m3 positive and definite
correction	the magneturde of the correction
value	the value

---

mapPathwaysIDfromGraphite  
*Map Pathways ID from Graphite*

---

**Description**

For internal use only. Retrieve pathway id and names from Pathways object.

**Usage**

```
mapPathwaysIDfromGraphite(pathways, pathwayNames = NULL)
```

**Arguments**

pathways	a PathwayList object
pathwayNames	in not NULL, a subset of pathway to extract

**Value**

a data frame, id and pathway name

---

minOrNA	<i>Minimum or NA</i>
---------	----------------------

---

**Description**

For internal use only. Get back minimum or NA.

**Usage**

```
minOrNA(x)
```

**Arguments**

x                    a numeric

**Value**

a numeric. The minimum or NA

**Examples**

```
# minOrNA(c(1,5,0.1,NA))
# minOrNA(c(NA,NA,NA))
```

---

MOSClip	<i>MOSClip: Multi-Omics Survival Clip</i>
---------	---

---

**Description**

MOSClip R package implements a statistical approach able to integrate multi-omic data and look for survival associated gene modules. It integrates multiple omics - transcriptomics, methylomics, genomic mutations, and genomic copy number variations - using various data dimensionality reduction strategies and multivariate models. Exploiting graph theory, pathways can be decomposed into their connected components, that we call modules. The analysis can then be performed at the level of entire pathways or pathway modules. MOSClip pathway analysis serves two primary purposes: testing the survival association of pathways or modules using the Cox proportional hazard model, and conducting a two-class analysis with a generalized linear model. Additionally, the package offers valuable graphical tools to visualize and interpret the results.

## Details

To conduct a multi-omic survival analysis on pathways or modules use:

- [multiOmicsSurvivalPathwayTest](#)
- [multiOmicsSurvivalModuleTest](#)

To perform a two-class comparison enrichment analysis on pathways or modules use:

- [multiOmicsTwoClassPathwayTest](#)
- [multiOmicsTwoClassModuleTest](#)

## Author(s)

**Maintainer:** Paolo Martini <paolo.martini@unibs.it> ([ORCID](#))

Authors:

- Anna Bortolato <anna.bortolato@unipd.it> ([ORCID](#))
- Anna Tanada <annacarolinaemi.delimatanada@phd.unipd.it> ([ORCID](#))
- Enrica Calura <enrica.calura@unipd.it> ([ORCID](#))
- Stefania Pirrotta <stefania.pirrotta@phd.unipd.it> ([ORCID](#))
- Federico Agostinis <federico.agostinis@outlook.com>

## References

Paolo Martini, Monica Chiogna, Enrica Calura, and Chiara Romualdi. 2019. “MOSClip: Multi-Omic and Survival Pathway Analysis for the Identification of Survival Associated Gene and Modules.” *Nucleic Acids Research* 47 (14): e80. <https://doi.org/10.1093/nar/gkz324>

## See Also

Useful links:

- <https://github.com/CaluraLab/MOSClip/>
- Report bugs at <https://github.com/CaluraLab/MOSClip/issues>



---

`multiOmics`*Omics class object with TCGA ovarian data*

---

### Description

An Omics class object containing data from TCGA ovarian cancer. The TCGA data was manually selected and preprocessed. It contains 4 omics: expression, methylation, mutation, and copy number variation. Additionally, it contains specific arguments to perform the dimensionality reduction. The datasets were downloaded from TCGA using TCGABiolink R package, selecting only patients with primary solid tumors. Expression matrix was processed first, converting gene identifiers into Entrez IDs. The profiles of genes present more than once were averaged. Genes with at least 100 counts in at least one patients were selected, to avoid data sparsity. Mutation matrix was filtered, keeping only genes with expression data available. We chose to consider only missense and non-sense mutations and mutation impact was also considered following Mutect2 pipeline. CNV values were transformed into numeric values. Methylation data were processed with Methyl Mix R package. Patients that had both normal and primary tumors samples were selected. With the help of a dictionary array probes were connected to CpG clusters, and finally CpG clusters were mapped to genes (Entrez ID). Survival annotation curated by Liu et al. (2018) was used to extract PFS information. Only patients with matched data across the four omics were considered. After the selection of patients and genes, we performed expression normalization and log2 of the counts+1 transformation. This will ensure us to work with expression data approximately close to a normal distribution, the most suitable distribution for the subsequent MOSClip tests. Genes and samples were manually selected to create this small example dataset for demonstration purposes.

### Usage

```
data('multiOmics')
```

### Format

`multiOmics:`

An Omics with 4 omics:

**exp** Matrix with 151 rows and 50 columns of RNA expression values

**met** A matrix with 178 rows and 50 columns of methylation data with probes clustered

**mut** A matrix with 107 rows and 50 columns of mutation counts

**cnv** A matrix with matrix with 145 rows and 50 columns of copy number ...

---

`MultiOmicsModules-class`*Multi Omics Modules.*

---

### Description

This class organizes the results of the Multi Omics Module Test analysis, in which corresponds to one pathway decomposed into modules.

**Usage**

```
## S4 method for signature 'MultiOmicsModules'
showModule(object)
```

**Arguments**

object            an object of class MultiOmicsModules

**Methods (by generic)**

- showModule(MultiOmicsModules): shows module info

**Slots**

alphas a numeric vector of the pvalues of all the modules.

zlists a list of numeric vectors with the zs of the covariates for each module.

modulesView a list of module information: for each omic, the name of the omic, the genes used, the method, the name of the covariates analyzed and other specific information based on the omic.

modules a list with the genes that belong to the module.

title the name of the pathway.

analysis the type of analysis done: survival or two-class.

---

MultiOmicsPathway-class

*Multi Omics Pathway.*

---

**Description**

This class organize the results of the Multi-Omics Pathway Survival Test analysis.

**Usage**

```
## S4 method for signature 'MultiOmicsPathway'
showPathway(object)
```

**Arguments**

object            an object of class MultiOmicsPathway

**Methods (by generic)**

- showPathway(MultiOmicsPathway): shows module info

**Slots**

`pvalue` a numeric vector of the pvalues of the pathways.  
`zlist` a numeric vector with the zs of all the covariates.  
`pathView` a list of pathway information: for each omic, the name of the omic, the genes used, the method, the name of the covariates analyzed and other specific information based on the omic.  
`title` the name of the pathway.  
`analysis` the type of analysis done: survival or two-class.

---

multiOmicsSurvivalModuleTest

*Compute Multi Omics Survival in Pathway Modules*


---

**Description**

Performs survival analysis using an Omics object. The pathway (graph) used is decomposed in modules (cliques) using graph theory.

**Usage**

```
multiOmicsSurvivalModuleTest(
  omicsObj,
  graph,
  survFormula = "Surv(days, status) ~",
  autoCompleteFormula = TRUE,
  useTheseGenes = NULL,
  pathName = NULL,
  robust = FALSE,
  include_from_annot = FALSE
)
```

**Arguments**

<code>omicsObj</code>	Object of class <code>Omics</code>
<code>graph</code>	a pathway in <code>graphNEL</code> , <code>Pathway</code> or <code>geneset</code> format
<code>survFormula</code>	a character with the formula to compute survival
<code>autoCompleteFormula</code>	logical. If <code>TRUE</code> autocomplete the <code>survFormula</code> using all the available covariates
<code>useTheseGenes</code>	vector of genes used to filter pathways
<code>pathName</code>	title of the pathway. If <code>NULL</code> and <code>graph</code> is <code>Pathway</code> , <code>graph@title</code> is used as title
<code>robust</code>	logical, whether the robust mode should be used for cox model analysis
<code>include_from_annot</code>	logical. If <code>TRUE</code> compute cox model analysis using additional covariates from <code>colData</code>

**Value**

MultiOmicsModules object

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)
```

---

multiOmicsSurvivalPathwayTest

*Compute Multi Omics Survival in Pathways*

---

**Description**

Performs topological survival analysis using an Omics object.

**Usage**

```
multiOmicsSurvivalPathwayTest(
  omicsObj,
  graph,
  survFormula = "Surv(days, status) ~",
  autoCompleteFormula = TRUE,
  useTheseGenes = NULL,
  pathName = NULL,
  robust = FALSE,
  include_from_annot = FALSE
)
```

**Arguments**

omicsObj	Object of class Omics
graph	a pathway in graphNEL, Pathway or geneset format
survFormula	a character with the formula to compute survival
autoCompleteFormula	logical. If TRUE autocomplete the survFormula using all the available covariates
useTheseGenes	vector of genes used to filter pathways

**pathName** title of the pathway. If NULL and graph is Pathway, graph@title is used as title  
**robust** logical, whether the robust mode should be used for cox model analysis  
**include\_from\_annot** logical. If TRUE compute cox model analysis using additional covariates from colData

### Value

MultiOmicsPathway object

### Examples

```

data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOP_survival <- multiOmicsSurvivalPathwayTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

```

---

multiOmicsTopo	<i>Omics class object with TCGA ovarian data for topological analysis</i>
----------------	---

---

### Description

An Omics class object containing data from TCGA ovarian cancer. The data are the same as in [multiOmics](#) object. Arguments in specificArgs slot have been set to efficiently run a topological pathway analysis, i.e., the topological method is used for PCA and shrink parameter is set to TRUE. This method can't be used for analyses on modules.

### Usage

```
data('multiOmicsTopo')
```

### Format

**multiOmicsTopo:**

An Omics with 4 omics:

**exp** Matrix with 151 rows and 50 columns of RNA expression values

**met** A matrix with 178 rows and 50 columns of methylation data with probes clustered

**mut** A matrix with 107 rows and 50 columns of mutation counts

**cnv** A matrix with matrix with 145 rows and 50 columns of copy number ...

---

 multiOmicsTwoClassModuleTest

*Computes Multi Omics Two-Class in Pathway Modules*


---

### Description

Performs topological two-class analysis using an Omics object. It decomposes graphs (pathways) into modules.

### Usage

```
multiOmicsTwoClassModuleTest(
  omicsObj,
  graph,
  classAnnot,
  baseFormula = "classes ~",
  autoCompleteFormula = TRUE,
  useTheseGenes = NULL,
  nullModel = "classes ~ 1",
  pathName = NULL
)
```

### Arguments

omicsObj	object of class Omics
graph	a pathway as a graphNEL object.
classAnnot	a data.frame with the class annotation. It is necessary at least a column with the classes labels, and the row.names as the samples labels
baseFormula	model formula to be used for the test. It should be written as 'classes ~', while 'classes' being the column name for the class labels
autoCompleteFormula	a logical value. If TRUE. It autocompletes the formula used to fit generalized linear models function using all the available covariates (omics)
useTheseGenes	(optional) vector of specific genes to be used
nullModel	the null model formula. It should be written the same as the baseFormula, followed by ' 1'. (e.g. 'classes ~ 1')
pathName	(optional) title of the pathway. If NULL, graph@title is used as title

### Value

MultiOmicsModule object

**Examples**

```

data("multiOmics")
data("reactSmall")

genesToUse <- row.names(multiOmics[[1]])

classAnnot <- data.frame(
  "treatment" = c(rep("A", 25), rep("B", 25)),
  row.names = colnames(multiOmics[[1]])
)

MOM_twoclasses <- multiOmicsTwoClassModuleTest(
  multiOmics, reactSmall[[1]], classAnnot,
  baseFormula = "treatment ~ ", nullModel = "treatment ~ 1",
  useTheseGenes = genesToUse
)

```

---

```

multiOmicsTwoClassPathwayTest
  Compute Multi Omics Two-Class in Pathways

```

---

**Description**

Performs topological two-class analysis using an Omics object.

**Usage**

```

multiOmicsTwoClassPathwayTest(
  omicsObj,
  graph,
  classAnnot,
  baseFormula = "classes ~ ",
  autoCompleteFormula = TRUE,
  useTheseGenes = NULL,
  nullModel = "classes ~ 1",
  pathName = NULL
)

```

**Arguments**

omicsObj	object of class Omics
graph	a pathway as a graphNEL object.
classAnnot	a data.frame with the class annotation. It is necessary at least a column with the classes labels, and the row.names as the samples labels
baseFormula	model formula to be used for the test. It should be written as 'classes ~ ', while 'classes' being the column name for the class labels

**autoCompleteFormula** a logical value. If TRUE. It autocompletes the formula used to fit generalized linear models function using all the available covariates (omics)  
**useTheseGenes** (optional) vector of specific genes to be used  
**nullModel** the null model formula. It should be written the same as the baseFormula, followed by ' 1'. (e.g. 'classes ~ 1')  
**pathName** (optional) title of the pathway. If NULL, graph@title is used as title

**Value**

MultiOmicsPathway object

**Examples**

```

data("multiOmics")
data("reactSmall")

genesToUse <- row.names(multiOmics[[1]])

classAnnot <- data.frame(
  "treatment" = c(rep("A", 25), rep("B", 25)),
  row.names = colnames(multiOmics[[1]])
)

MOP_twoClasses <- multiOmicsTwoClassPathwayTest(
  multiOmics, reactSmall[[1]], classAnnot,
  baseFormula = "treatment ~ ", nullModel = "treatment ~ 1",
  useTheseGenes = genesToUse
)

```

---

multiPathwayModuleReport

*Provides a Table of the Modules Test Results*

---

**Description**

Summarizes the results of a multi omics module test given a list of MultiOmicsModules objects

**Usage**

```
multiPathwayModuleReport(multiPathwayModuleList, priority_to = NULL)
```

**Arguments**

**multiPathwayModuleList** a list of MultiOmicsModules objects resulting from a multi-omics module test.  
**priority\_to** a vector with the covariates (the omics names) that should appear first in the dataframe columns



**Value**

a data.frame class object. Rows correspond to the modules, and the columns to the overall and covariates pvalues of the test.

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_list <- lapply(reactSmall[1:2], function(g) {
  fcl <- multiOmicsSurvivalModuleTest(multiOmics, g,
    survFormula = "Surv(days, status) ~",
    autoCompleteFormula = TRUE,
    useTheseGenes = genesToUse
  )
  fcl
})

moduleSummary <- multiPathwayModuleReport(MOM_list)
```

---

multiPathwayReport	<i>Summarize pathways' info from a list of MultiOmicsPathway objects (MOP)</i>
--------------------	--

---

**Description**

Given the list of MOPs, it creates the table.

**Usage**

```
multiPathwayReport(multiPathwayList, priority_to = NULL)
```

**Arguments**

**multiPathwayList** a list of MultiOmicsPathway objects resulting from a multi-omics pathway test.

**priority\_to** a vector with the covariates (omic name) that should go first.

**Value**

a data.frame, pathways in rows, overall pvalue of the coxph, followed by covariates pvalues, in columns.

### Examples

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOP_list <- lapply(reactSmall, function(g) {
  fcl <- multiOmicsSurvivalPathwayTest(multiOmics, g,
    survFormula = "Surv(days, status) ~",
    autoCompleteFormula = TRUE,
    useTheseGenes = genesToUse
  )
  fcl
})

pathwaysSummary <- multiPathwayReport(MOP_list)
```

---

Omics-class

*Omics.*

---

### Description

This class is the storage for the different omic datasets that we need to analyze. It is based on MultiAssayExperiment.

### Usage

```
## S4 method for signature 'Omics'
showOmics(object)
```

### Arguments

object            an object of class Omics

### Methods (by generic)

- showOmics(Omics): shows model parameters

### Slots

modelInfo a list with length equal to length(data) that are modelInfo to process each dataset.

specificArgs a list with length equal to length(data) to set additional parameters specific of the modelInfo.

---

ovarianDataset	<i>ExperimentList class object with TCGA ovarian data</i>
----------------	---

---

### Description

An ExperimentList class object containing data from TCGA ovarian cancer. The TCGA data was manually selected and preprocessed. It contains 4 omics: expression, methylation, mutation, and copy number variation.

### Usage

```
data('ovarianDataset')
```

### Format

ExperimentList:

An ExperimentList with 4 omics:

**exp** Matrix with 101 rows and 50 columns of RNA expression values

**met** A matrix with 97 rows and 50 columns of methylation data with probes clustered

**mut** A matrix with 55 rows and 50 columns of mutation counts

**cnv** A matrix with matrix with 101 rows and 50 columns of copy number ...

---

plotFrequencies	<i>Plot Frequencies of Pathway Fathers for Omics intersection</i>
-----------------	---

---

### Description

Plots the frequencies of the pathway fathers by every omics intersection from a data.frame of the frequencies returned with the function [computeFreqs](#).

### Usage

```
plotFrequencies(
  frequencies,
  manualColors = NULL,
  minSize = 4,
  maxSize = 20,
  width = 20,
  relMagnificationOfLegend = 0.5,
  lineSize = 1
)
```

**Arguments**

frequencies	a data.frame created from 'computeFreqs'
manualColors	optional vector of colors to be used
minSize	the minimal font size. Maximal frequencies will be added for each class
maxSize	the maximal font size dimension, all values above are clipped
width	the number of character to wrap the labels
relMagnificationOfLegend	the relative magnification of the text of the legend
lineSize	the thickness of the lines

**Value**

a circular plot of the frequencies of pathway fathers

**Examples**

```
df <- data.frame(
  category = c("PathwayA", "PathwayB", "PathwayC", "PathwayD"),
  frequencies = c(1, 2, 1, 3),
  class = rep("Mut", 4), stringsAsFactors = FALSE
)
plotFrequencies(df)
```

---

plotModuleHeat	<i>Plot a Heatmap of a Module by Omics</i>
----------------	--

---

**Description**

It creates a heatmap of the most involved genes of each omic of a specific module from a MultiOmicsModule object.

**Usage**

```
plotModuleHeat(
  moduleobj,
  moduleNumber,
  sortBy = NULL,
  paletteNames = NULL,
  additionalAnnotations = NULL,
  additionalPaletteNames = NULL,
  withSampleNames = TRUE,
  fontsize_row = 10,
  fontsize_col = 1,
  nrowsHeatmaps = 3,
  orgDbi = "org.Hs.eg.db",
```

```

    discr_prop_pca = 0.15,
    discr_prop_events = 0.05,
    ...
)

```

### Arguments

moduleobj	MultiOmicsModule class object
moduleNumber	module number of interest
sortBy	a covariate (omic) to sort by
paletteNames	a palette containing three colors
additionalAnnotations	optional additional sample annotations
additionalPaletteNames	optional additional colors for annotations
withSampleNames	show sample names
fontsize_row	font size for row labels
fontsize_col	font size for column labels
nrowsHeatmaps	magnification respect to annotation of sample (annotations take 1 row)
orgDbi	a Dbi organism to be used. Default is org.Hs.eg.db
discr_prop_pca	the minimal proportion to compute the PCA classes
discr_prop_events	the minimal proportion to compute the event classes
...	additional arguments passed to guessInvolvement function

### Value

A heatmap of a pathway module (results of the module test)

### Examples

```

data(multiOmics)
data(reactSmall)

survAnnot <- data.frame(
  status = multiOmics$status,
  days = multiOmics$days,
  row.names = colnames(multiOmics[[1]])
)

genesToUse <- row.names(multiOmics[[1]])

MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

```

```

plotModuleHeat(MOM_survival, 1,
  sortBy = c("mut", "expPC1", "status", "days"),
  additionalAnnotations = survAnnot,
  additionalPaletteNames = list(status = "teal", days = "violet"),
  withSampleNames = F
)

```

---

plotModuleInGraph      *Plot a Directed Graph of the MultiOmicsModules Object*

---

### Description

From a MultiOmicsModules object, it plots the position of a given module in the pathway. The omics are also represented in the graph.

### Usage

```

plotModuleInGraph(
  modulesobj,
  pathList,
  moduleNumber,
  orgDbi = "org.Hs.eg.db",
  paletteNames = NULL,
  legendLabels = NULL,
  fileName = NULL,
  discr_prop_pca = 0.15,
  discr_prop_events = 0.05,
  pathTitle = NULL,
  ...
)

```

### Arguments

modulesobj	a MultiOmicsModule class object
pathList	a PathwayList from graphite package that contains the pathways to be used
moduleNumber	a module number
orgDbi	if needed, indicates an organism DbI to translate the vectors
paletteNames	named vector of MOSpalettes, names replace makeLegend arguments
legendLabels	set up your favourite names for the omics
fileName	optional filenames to save the plot
discr_prop_pca	the minimal proportion to compute the PCA classes
discr_prop_events	the minimal proportion to compute the event classes
pathTitle	title of the graph, to be searched in pathList
...	additional arguments passed to guessInvolvement function

**Value**

a MOSClip plot in form of a list class object

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

plotModuleInGraph(MOM_survival, reactSmall,
  moduleNumber = 1,
  paletteNames = c(exp = "red", met = "green",
    mut = "blue", cnv = "yellow")
)
```

---

plotModuleKM

*Plot Kaplan-Meier survival curves of a specific module*


---

**Description**

Given a MultiOmicsModule class object and a specific module number, it plots Kaplan-Meier curves, in which the strata corresponds to the omics

**Usage**

```
plotModuleKM(
  MOM,
  moduleNumber,
  formula = "Surv(days, status) ~ PC1",
  fileName = NULL,
  paletteNames = NULL,
  h = 9,
  w = 7,
  risk.table = TRUE,
  pval = TRUE,
  size = 1,
  inYears = FALSE,
  discr_prop_pca = 0.15,
  discr_prop_events = 0.05,
  additional_discrete = NULL,
  additional_continuous = NULL,
```

```
    ...
  )
```

### Arguments

MOM	a MultiOmicsModule class object
moduleNumber	numeric value. The module number of interest
formula	a formula for the survival analysis. It should be written as 'Surv(days, status) ~ omic'. To plot more than one omic, write them separated by a '+' character after the separator (~)
fileName	optional filenames to save the plot
paletteNames	a palette name to be used
h	the height of the plot
w	the width of the plot
risk.table	logical value. If TRUE, shows the risk.table. Default is TRUE.
pval	logical value. If TRUE, shows the p-value of the curves. Default is TRUE.
size	line width of the KM curves
inYears	set time in years
discr_prop_pca	the minimal proportion to compute the PCA classes
discr_prop_events	the minimal proportion to compute the event classes
additional_discrete	names of the additional discrete variables to include
additional_continuous	names of the additional continuous variables to include
...	additional arguments passed to guessInvolvement and get function

### Value

a ggsurvplot class object

### Examples

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

plotModuleKM(MOM_survival, 1,
  formula = "Surv(days, status) ~ mut + expPC2",
  paletteNames = "Paired", inYears = TRUE
)
```



---

plotModuleReport	<i>Plot a table of a MultiOmicsModules (MOM) object</i>
------------------	---

---

### Description

Given a MultiOmicsModules object, it plots its results in a tabular fashion

### Usage

```
plotModuleReport(  
  modulesObj,  
  MOcolors = NULL,  
  priority_to = NULL,  
  fontsize = 12,  
  ...  
)
```

### Arguments

modulesObj	MultiOmicsModules class object
MOcolors	character vector with the omic colors. The colors should be among the colors in <a href="#">showMOSpalette</a>
priority_to	a vector with the covariates (omic names) that should go first
fontsize	Size of the font to be used in the plot
...	additional argument to be passed to pheatmap

### Value

a Heatmap list object from ComplexHeatmap package of the results contained in the MultiOmicsModules object provided

### Examples

```
data(multiOmics)  
data(reactSmall)  
  
genesToUse <- row.names(multiOmics[[1]])  
  
MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],  
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,  
  useTheseGenes = genesToUse  
)  
  
plotModuleReport(MOM_survival,  
  MOcolors = c(  
    exp = "red", met = "green", mut = "blue",  
    cnv = "yellow"
```

```
)
)
```

---

plotMultiPathwayReport

*Summarize and plot pathways' info from a list of MultiOmicPathway (MOP)*

---

### Description

Given the list of MOPs, it plots a table of its results.

### Usage

```
plotMultiPathwayReport(
  multiPathwayList,
  top = 25,
  MOcolors = NULL,
  priority_to = NULL,
  fontsize = 6,
  ...
)
```

### Arguments

multiPathwayList	a list of MultiOmicPathway class objects
top	numeric value. Plot only the top number of pathways
MOcolors	character vector with the omic colors. The colors should be among the colors in <a href="#">showMOSpalette</a>
priority_to	a vector with the covariates (omic names) that should go first
fontsize	the font size to be used. Default is 12.
...	additional argument to be passed to pheatmap

### Value

a Heatmap list object from ComplexHeatmap package of the results contained in the MultiOmicPathway object provided

**Examples**

```

data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOP_list <- lapply(reactSmall, function(g) {
  fcl <- multiOmicsSurvivalPathwayTest(multiOmics, g,
    survFormula = "Surv(days, status) ~",
    autoCompleteFormula = TRUE,
    useTheseGenes = genesToUse
  )
  fcl
})

plotMultiPathwayReport(MOP_list,
  MOcolors = c(
    exp = "red", met = "green", mut = "blue",
    cnv = "yellow"
  ),
  fontsize = 12
)

```

---

plotPathwayHeat	<i>Plot heatmaps of the pathway by omics</i>
-----------------	--

---

**Description**

Given the pathway, it creates the heatmaps of the mostly involved genes for each omic.

**Usage**

```

plotPathwayHeat(
  pathway,
  sortBy = NULL,
  paletteNames = NULL,
  additionalAnnotations = NULL,
  additionalPaletteNames = NULL,
  discr_prop_pca = 0.15,
  discr_prop_events = 0.05,
  withSampleNames = TRUE,
  nrowsHeatmaps = 3,
  orgDbi = "org.Hs.eg.db",
  ...
)

```

**Arguments**

pathway	MultiOmicsPathway class object
sortBy	one or more covariates to sort the samples
paletteNames	name of the colors for each omic
additionalAnnotations	optional additional sample annotations (e.g. survival annotation)
additionalPaletteNames	colors for additional annotations. The colors available are the ones in <a href="#">showMOSpalette</a>
discr_prop_pca	the minimal proportion to compute the PCA classes
discr_prop_events	the minimal proportion to compute the event classes
withSampleNames	show the sample names in the plot
nrowsHeatmaps	magnification respect to annotation of sample (annotations take 1 row)
orgDbi	a Dbi organism to be used. Default is org.Hs.eg.db
...	additional arguments passed to guessInvolvementPathway function (internal use)

**Value**

An object of class ggplot plotted with ComplexHeatMap package.

**Examples**

```

data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

survAnnot <- data.frame(
  status = multiOmics$status,
  days = multiOmics$days,
  row.names = colnames(multiOmics[[1]])
)

# Creating the MultiOmicsPathway object
MOP_survival <- multiOmicsSurvivalPathwayTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

# Plotting
plotPathwayHeat(MOP_survival,
  sortBy = c("expPC2", "mut", "status", "days"),
  paletteNames = c(exp = "red", met = "green",
    mut = "blue", cnv = "yellow"),
  additionalAnnotations = survAnnot,
  additionalPaletteNames = list(status = "teal", days = "violet"),

```

```

    nrowsHeatmaps = 2, withSampleNames = F
  )

```

---

plotPathwayKM

*Plot Kaplan-Meier survival curves of a specific pathway*


---

### Description

Given a MultiOmicsPathway class object, it plots Kaplan-Meier curves, in which the strata corresponds to the chosen omics

### Usage

```

plotPathwayKM(
  pathway,
  formula = "Surv(days, status) ~ PC1",
  fileName = NULL,
  paletteNames = NULL,
  h = 9,
  w = 7,
  risk.table = TRUE,
  pval = TRUE,
  size = 1,
  inYears = FALSE,
  discr_prop_pca = 0.15,
  discr_prop_events = 0.05,
  additional_discrete = NULL,
  additional_continuous = NULL,
  ...
)

```

### Arguments

pathway	MultiOmicsPathway class object
formula	a formula to compute the plot
fileName	optional filenames to save the plot
paletteNames	a palette containing three colors
h	the height of the plot
w	the width of the plot
risk.table	logical value. If TRUE, shows the risk.table. Default is TRUE.
pval	logical value. Shows p-value of the curves
size	line width of the KM curves
inYears	logical value. If TRUE, converts days to years

`discr_prop_pca` the minimal proportion to compute the PCA classes  
`discr_prop_events`  
                   the minimal proportion to compute the event classes  
`additional_discrete`  
                   names of the additional discrete variables to include  
`additional_continuous`  
                   names of the additional continuous variables to include  
`...` additional arguments passed to `guessInvolvementPathway` and `get` function  
                   (internal use)

**Value**

a `ggsurvplot` class object

**Examples**

```

data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

# Creating the MultiOmicsPathway object
MOP_survival <- multiOmicsSurvivalPathwayTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

plotPathwayKM(MOP_survival,
  formula = "Surv(days, status) ~ mut + expPC2",
  paletteNames = "Paired", inYears = TRUE
)

```

---

pvalueSummary

*Compute pvalue Summary*

---

**Description**

Compute pvalue Summary

**Usage**

```
pvalueSummary(multiPathwayReportData, excludeColumns = NULL, as.list = FALSE)
```

**Arguments**

- `multiPathwayReportData` data.frame, the output of the `multiPathwayReport` or `multiPathwayModuleReport` functions.
- `excludeColumns` a vector of characters listing the columns of `multiPathwayReportData` object to be excluded by the analysis. In the case `multiPathwayReportData` derives from `multiPathwayModuleReport` you should set `excludeColumns = c('pathway', 'module')`.
- `as.list` return a list rather than a data.frame

**Value**

a list

---

reactSmall	<i>PathwayList of pathways from Reactome</i>
------------	--

---

**Description**

A `PathwayList` with three pathways necessary for the analysis: 'Activation of Matrix Metalloproteinases', 'FGFR1 mutant receptor activation', and 'VEGFA-VEGFR2 Pathway'. Pathways were downloaded using `graphite` package and the names of the nodes were converted into Entrez IDs.

**Usage**

```
data('reactSmall')
```

**Format**

`reactSmall`:  
A `PathwayList` with Reactome pathways for `hsapiens`  
**entries** Three Reactome pathways with their nodes

---

removeSelfLoops	<i>Remove self loops from a graphNEL</i>
-----------------	--

---

**Description**

Remove the self loops that are present in the graph `graphNEL` object

**Usage**

```
removeSelfLoops(graph)
```





**Value**

list of the resampling tables of results

list of the resampling tables of results

**Examples**

```
data(multiOmics)
data(reactSmall)

perms <- resamplingModulesSurvival(
  fullMultiOmics = multiOmics, reactSmall,
  nperm = 10,
  pathwaySubset =
    "FGFR1 mutant receptor activation"
)
```

---

resamplingModulesTwoClass

*Resampling function for two-class analysis on modules*

---

**Description**

Resampling function for two-class analysis on modules

Resampling function for pathways (two-class analysis)

**Usage**

```
resamplingModulesTwoClass(
  fullMultiOmics,
  classAnnot,
  pathdb,
  nperm = 100,
  pathwaySubset = NULL,
  nPatients = 3,
  genesToConsider = NULL
)
```

```
resamplingPathwayTwoClass(
  fullMultiOmics,
  classAnnot,
  pathdb,
  nperm = 100,
  pathwaySubset = NULL,
  nPatients = 3,
  genesToConsider = NULL
)
```

**Arguments**

`fullMultiOmics` a multiOmic object  
`classAnnot` patients class annotations  
`pathdb` pathway database  
`nperm` number of permutations  
`pathwaySubset` a list of pathways to resample  
`nPatients` number of patients to remove for resampling  
`genesToConsider` vector of genes used to filter pathways; if NULL, genes found in the first experiment of the multiOmic object are used

**Value**

list of the resampling tables of results  
list of the resampling tables of results

**Examples**

```
data(multiOmics)
data(reactSmall)

classAnnot <- data.frame(
  "treatment" = c(rep("A", 25), rep("B", 25)),
  row.names = colnames(multiOmics[[1]])
)

perms <- resamplingModulesTwoClass(
  fullMultiOmics = multiOmics,
  classAnnot, reactSmall,
  nperm = 10,
  pathwaySubset =
    "FGFR1 mutant receptor activation"
)
```

---

`runSupertest`*Performs a Exact test - analysis of omics intersection*

---

**Description**

This function performs a exact test implementing a theoretical framework using the SuperExactTest package. It calculates the statistical distributions of multi omics set intersections. It can be used with both a MultiOmicsModules or MultiOmicsPathway class objects.

**Usage**

```
runSupertest(
  multiPathwayReportData,
  pvalueThr = 0.05,
  zscoreThr = 0.05,
  resampligThr = NULL,
  plot = c("circular", "landscape", "noplot"),
  sort.by = c("set", "size", "degree", "p-value"),
  excludeColumns = NULL,
  color.on = "#f6bb42",
  color.off = "#D3D3D3"
)
```

**Arguments**

multiPathwayReportData	data.frame, the output of the <a href="#">multiPathwayReport</a> or <a href="#">multiPathwayModuleReport</a> functions.
pvalueThr	numeric value. Overall pvalue cut-off to be used
zscoreThr	numeric value. Covariates coefficient cut-off to be used.
resampligThr	numeric value. Filters the modules according to the number of success in the resampling procedure, takes only the modules above this threshold.
plot	character indicating the layout for plotting. It is one of circular, landscape or noplot. By default, plot='circular', if plot='noplot' no plot will be provided.
sort.by	character indicating how to sort the intersections in the plot. It is one of 'set' (by omics), 'size' (by intersection size), 'degree' (by number of intersected omics), and 'p-value'.
excludeColumns	a vector of characters listing the columns of multiPathwayReportData object to be excluded by the analysis. In the case multiPathwayReportData derives from <a href="#">multiPathwayModuleReport</a> you should set excludeColumns = c('pathway', 'module').
color.on	color that represent the active omics in the sector
color.off	color that represent the omics mnot considered in the sector

**Details**

This function calculates intersection sizes between multiple set of pathways or modules and performs statistical test of the intersections using the total amount of analyzed pathways or modules as background. The super exact test of this function was described in Wang et al 2015.

**Value**

a data.frame containing all the numeric information of the plot included the pathways shared by different omics.

## References

Minghui Wang, Yongzhong Zhao, and Bin Zhang (2015). Efficient Test and Visualization of Multi-Set Intersections. *Scientific Reports* 5: 16923.

## Examples

```
df <- data.frame(
  pvalue = c(0.06, 0.04, 0.04, 0.03, 0.02),
  cnv = c(0.07, 0.03, 0.02, 0.04, 0.01),
  mut = c(0.08, 0.02, 0.01, 0.04, 0.04),
  row.names = c(
    "PathwayA", "PathwayB", "PathwayC",
    "PathwayD", "PathwayE"
  )
)

runSupertest(df, pvalueThr = 0.05, zscoreThr = 0.05)
```

---

```
selectStablePathwaysModules
      Select stable pathway modules
```

---

## Description

Select stable pathway modules  
 Count the resampling success  
 Add resampling counts to module summary

## Usage

```
selectStablePathwaysModules(perms, moduleSummary, success = 90, col = "pvalue")

getPathwaysModulesSuccess(perms, moduleSummary, col = "pvalue", thr = 0.05)

addResamplingCounts(moduleSummary, resamplingCounts)
```

## Arguments

perms	a list. Result of resampling function
moduleSummary	summary of modules or pathways obtained from <code>multiPathwayModuleReport</code> or <code>multiPathwayReport</code>
success	number of success to consider the pathway or module stable
col	the name of the column in the summary to be used to evaluate resampling success
thr	the threshold for significance
resamplingCounts	the counts of success obtained with <code>getPathwaysModulesSuccess</code>

**Value**

the subset of stable modules

the counts of success for each pathway or module

a module or pathway summary with resampling counts column appended

**Examples**

```
data("multiOmics")
data("reactSmall")

perms <- resamplingPathwaySurvival(multiOmics, reactSmall, nperm = 5)
res <- lapply(reactSmall, function(g) {
  multiOmicsSurvivalPathwayTest(multiOmics, g,
    useTheseGenes = row.names(multiOmics[[1]])
  )
})
pathSummary <- multiPathwayReport(res)
getPathwaysModulesSuccess(perms, pathSummary)
```

---

 showModule

*A generic function showing pathway's module info*


---

**Description**

A generic function showing pathway's module info

**Usage**

```
showModule(object)
```

**Arguments**

object            an object of class MultiOmicsModules

**Value**

NULL. No value is returned

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOM_survival <- multiOmicsSurvivalModuleTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
```

```

        useTheseGenes = genesToUse
    )
    showModule(MOM_survival)

```

---

showMOSpalette	<i>Shows the MOSClip palette.</i>
----------------	-----------------------------------

---

**Description**

This function shows the MOSClip palette. Each omic should be coupled to a color panel, this match will be preserved in plots.

**Usage**

```
showMOSpalette()
```

**Value**

NULL. No value is returned

**Examples**

```
showMOSpalette()
```

---

showOmics	<i>A generic functions showing parameter associated with each omics</i>
-----------	---

---

**Description**

A generic functions showing parameter associated with each omics

**Usage**

```
showOmics(object)
```

**Arguments**

object            an object of class Omics

**Value**

NULL. No value is returned

**Examples**

```
data(multiOmics)
showOmics(multiOmics)
```

---

showPathway	<i>A generic function showing pathway info</i>
-------------	--

---

**Description**

A generic function showing pathway info

**Usage**

```
showPathway(object)
```

**Arguments**

object            an object of class MultiOmicsPathway

**Value**

NULL. No value is returned

**Examples**

```
data(multiOmics)
data(reactSmall)

genesToUse <- row.names(multiOmics[[1]])

MOP_survival <- multiOmicsSurvivalPathwayTest(multiOmics, reactSmall[[1]],
  survFormula = "Surv(days, status) ~", autoCompleteFormula = TRUE,
  useTheseGenes = genesToUse
)

showPathway(MOP_survival)
```

---

sparseCompPCs	<i>Sparse PCA</i>
---------------	-------------------

---

**Description**

Sparse PCA

**Usage**

```
sparseCompPCs(exp, shrink, k)
```

**Arguments**

exp	a matrix
shrink	logical, whether to shrink or not.
k	the number of components to use

**Value**

a list with the following elements:

x	the computed PCs
sdev	the standard deviation captured by the PCs
loadings	the loadings

---

stripModulesFromPathways

*Remove Module Number From Pathway Name*

---

**Description**

Function to remove the suffix corresponding to the module number of the pathway name. Necessary step for [annotatePathwayToFather](#) and [plotFrequencies](#)

**Usage**

```
stripModulesFromPathways(pathways)
```

**Arguments**

pathways	vector of pathway names
----------	-------------------------

**Value**

list of pathway names without the module number



## Examples

```
pathwaysModules <- list(
  "Intrinsic Pathway for Apoptosis.1",
  "Intrinsic Pathway for Apoptosis.2",
  "Opioid Signalling.1", "Opioid Signalling.2"
)

resPathwayNames <- stripModulesFromPathways(pathwaysModules)
```

---

summarizeInCluster      *Summarize Using Cluster Analysis*

---

## Description

Given a matrix it summarize in classes

## Usage

```
summarizeInCluster(
  data,
  features,
  name = "clu",
  dictionary = NULL,
  max_cluster_number = 3,
  cliques = NULL
)
```

## Arguments

data	a data matrix
features	a vector with the features to analyze
name	prefix of the covariates
dictionary	translate features (genes) into sets (row.names of the data)
max_cluster_number	the maximum number of cluster to evaluate
cliques	the features organized in cliques. Only use for topology

## Details

The user can define a maximum of classes. The function guess the optimal number of clusters using NbClust methods.

**Value**

a list with summary of the omic:

x	summary of the omic for each sample
usedGenes	genes list of genes used to calculate the summary
namesCov	names of the covariates
cls	the genes in clusters
method	method used for the analysis
omicName	name of the omic

---

summarizeOmicsResByMinPvalue

*Summarize Omics Covaraites By Min Pvalue*

---

**Description**

For internal use only. for each line extrac 'col' and get the minimum.

**Usage**

```
summarizeOmicsResByMinPvalue(col, mat)
```

**Arguments**

col	columns to extract from the line
mat	the matrix to be summarized (were to extract lines and 'col')

**Value**

a summarized version of the matrix.

**Examples**

```
# summarizeOmicsResByMinPvalue(2:3, mat=matrix(c(1,2,4,1,2,5), nrow=2))
```

---

`summarizeToBinaryDirectionalEvents`*Summarize To Binary Directional Events*

---

**Description**

Given a matrix it summarize the positive and negative to 0 or 1 in two vectors

**Usage**

```
summarizeToBinaryDirectionalEvents(  
  data,  
  features,  
  name = "dirBin",  
  binaryClassMin = 10,  
  eventThr = 2,  
  cliques = NULL  
)
```

**Arguments**

<code>data</code>	a data matrix
<code>features</code>	a vector with the features to analyze
<code>name</code>	prefix of the covariates
<code>binaryClassMin</code>	the minimum number of event to include the covariate
<code>eventThr</code>	the absolute value to threshold an event
<code>cliques</code>	the features organized in cliques. Only use for topology

**Value**

a list with summary of the omic:

<code>x</code>	summary of the omic for each sample
<code>usedGenes</code>	genes list of genes used to calculate the summary
<code>namesCov</code>	names of the covariates
<code>method</code>	method used for the analysis
<code>omicName</code>	name of the omic
<code>evenThr</code>	threshold fot event counting

---

`summarizeToBinaryEvents`*Summarize To Binary Events*

---

**Description**

Given a matrix it summarize to a 0 or 1

**Usage**

```
summarizeToBinaryEvents(  
  data,  
  features,  
  name = "bin",  
  binaryClassMin = 10,  
  cliques = NULL  
)
```

**Arguments**

<code>data</code>	a data matrix
<code>features</code>	a vector with the features to analyze
<code>name</code>	prefix of the covariates
<code>binaryClassMin</code>	the minimum number of event to include the covariate
<code>cliques</code>	the features organized in cliques. Only use for topology

**Value**

a list with summary of the omic:

<code>x</code>	summary of the omic for each sample
<code>usedGenes</code>	genes list of genes used to calculate the summary
<code>namesCov</code>	names of the covariates
<code>method</code>	method used for the analysis
<code>omicName</code>	name of the omic
<code>evenThr</code>	threshold fot event counting

---

```
summarizeToNumberOfDirectionalEvents
      Summarize With Directed Sum
```

---

**Description**

Given a matrix it summarize the positive and negative in two vectors, with counts of the events

**Usage**

```
summarizeToNumberOfDirectionalEvents(
  data,
  features,
  name = "dCount",
  eventThr = 2,
  min_prop = 0.1,
  cliques = NULL
)
```

**Arguments**

data	a data matrix
features	a vector with the features to analyze
name	prefix of the covariates
eventThr	the absolute value to threshold an event
min_prop	minimal proportion in classes
cliques	the features organized in cliques. Only use for topology

**Value**

a list with summary of the omic:

x	summary of the omic for each sample
usedGenes	genes list of genes used to calculate the summary
namesCov	names of the covariates
method	method used for the analysis
omicName	name of the omic
eventThr	threshold fot event counting
min_prop	minimum proportion of samples to exclude to check the variability of values

---

`summarizeToNumberOfEvents`*Summarize To Number of Binary Events*

---

**Description**

Given a matrix it summarize to a 0 or 1

**Usage**

```
summarizeToNumberOfEvents(  
  data,  
  features,  
  name = "event",  
  min_prop = 0.1,  
  cliques = NULL  
)
```

**Arguments**

<code>data</code>	a data matrix
<code>features</code>	a vector with the features to analyze
<code>name</code>	prefix of the covariates
<code>min_prop</code>	minimal proportion in classes
<code>cliques</code>	the features organized in cliques. Only use for topology

**Value**

a list with summary of the omic:

<code>x</code>	summary of the omic for each sample
<code>usedGenes</code>	genes list of genes used to calculate the summary
<code>namesCov</code>	names of the covariates
<code>method</code>	method used for the analysis
<code>omicName</code>	name of the omic
<code>evenThr</code>	threshold for event counting
<code>min_prop</code>	minimum proportion of samples to exclude to check the variability of values

---

summarizeWithPca	<i>Summarize Using PCA</i>
------------------	----------------------------

---

**Description**

Given a matrix it summarize to principal components. The user can specify the number of principal components. Default 3.

**Usage**

```
summarizeWithPca(
  data,
  features,
  name = "pca",
  shrink = FALSE,
  method = "regular",
  cliques = NULL,
  maxPCs = 3,
  loadThr = 0.6
)
```

**Arguments**

data	a data matrix
features	a vector with the features to analyze
name	prefix of the covariates
shrink	shrink or not the covariance matrix.
method	either 'regular', 'sparse' or 'topological'
cliques	the features organized in cliques. Only use for topology.
maxPCs	maximum number of pcs to consider
loadThr	loading threshold

**Value**

a list with summary of the omic:

x	summary of the omic for each sample (principal components)
sdev	standard deviation of the principal components
loadings	loadings of PCA
usedGenes	genes list of genes used to calculate the summary
namesCov	names of the covariates
method	method used for the analysis
omicName	name of the omic

---

 survivalcox

*Cox Model Analysis*


---

**Description**

Cox Analysis

**Usage**

```
survivalcox(coxObj, formula)
```

**Arguments**

coxObj	data.frame: patients x covariates
formula	formula to use

**Details**

For internal use only

**Value**

A list with

pvalue	pvalue of the model
zlist	pvalues of single covariates
coxObj	the original coxObj passed to the function

---

 survivalcoxr

*Cox Robust Model Analysis*


---

**Description**

Cox Robust Analysis

**Usage**

```
survivalcoxr(coxObj, formula)
```

```
coxrsummary(x)
```

**Arguments**

coxObj	data.frame: patients x covariates
formula	formula to use
x	a coxr.obj



**Details**

For internal use only

**Value**

A list with

pvalue	pvalue of the model
zlist	pvalues of single covariates
coxObj	the original coxObj passed to the function
	a list with wald test and robust and partial coefficients

---

topoCompPCs	<i>Topological PCA</i>
-------------	------------------------

---

**Description**

Topological PCA

**Usage**

```
topoCompPCs(exp, shrink, cliques, k)
```

**Arguments**

exp	a matrix
shrink	logical, whether to shrink or not.
cliques	the pathway topology summarized in a list of cliques
k	the number of components to use

**Value**

a list with the following elements:

x	the computed PCs
sdev	the standard deviation captured by the PCs
loadings	the loadings

# Index

- \* **datasets**
  - multiOmics, 25
  - multiOmicsTopo, 29
  - ovarianDataset, 35
  - reactSmall, 47
- addResamplingCounts
  - (selectStablePathwaysModules), 52
- annotatePathwayToFather, 3, 56
- availableOmicMethods, 5
- checkOrder, 5
- compPCs, 6
- computeFreqs, 7, 35
- computeOmicsIntersections, 8
- computePCs, 9
- convertPathway, 10
- coxrsurvival (survivalcox), 64
- createCoxObj, 10
- createDataModule, 11
- createMOMView, 11
- DataFrame, 21
- downloadPathwayRelationFromReactome, 12
- estimateExprCov, 12
- ExperimentList, 21
- extractCliquesFromDag, 13
- extractSummaryFromBinary, 13
- extractSummaryFromCluster, 14
- extractSummaryFromNumberOfEvents, 15
- extractSummaryFromPCA, 16
- filterExpr (checkOrder), 5
- filterMultiOmicsForSamples
  - (checkOrder), 5
- getPathFathers, 17
- getPathwaysModulesSuccess
  - (selectStablePathwaysModules), 52
- glmTest, 17
- guessInvolvement, 18
- guessInvolvementPathway, 19
- id2name, 20
- makeOmics, 20
- makePositiveDefinite, 22
- mapPathwaysIDfromGraphite, 22
- mergeCol (checkOrder), 5
- minOrNA, 23
- mmoralize (extractCliquesFromDag), 13
- MOSClip, 23
- MOSClip-package (MOSClip), 23
- multiOmics, 25, 29
- MultiOmicsModules-class, 25
- MultiOmicsPathway-class, 26
- multiOmicsSurvivalModuleTest, 24, 27
- multiOmicsSurvivalPathwayTest, 24, 28
- multiOmicsTopo, 29
- multiOmicsTwoClassModuleTest, 24, 30
- multiOmicsTwoClassPathwayTest, 24, 31
- multiPathwayModuleReport, 8, 32, 47, 51
- multiPathwayReport, 8, 33, 47, 51
- Omics (Omics-class), 34
- Omics-class, 34
- ovarianDataset, 35
- plotFrequencies, 7, 35, 56
- plotModuleHeat, 36
- plotModuleInGraph, 38
- plotModuleKM, 39
- plotModuleReport, 41
- plotMultiPathwayReport, 42
- plotPathwayHeat, 43
- plotPathwayKM, 45

preparePerms (checkOrder), 5  
pvalueSummary, 46

reactSmall, 47  
removeSelfLoops, 47  
resamplingModulesSurvival, 48  
resamplingModulesTwoClass, 49  
resamplingPathwaySurvival  
    (resamplingModulesSurvival), 48  
resamplingPathwayTwoClass  
    (resamplingModulesTwoClass), 49  
resolveAndOrder (checkOrder), 5  
runSupertest, 50

selectStablePathwaysModules, 52  
showModule, 53  
showModule, MultiOmicsModules-method  
    (MultiOmicsModules-class), 25  
showMOSpalette, 41, 42, 44, 54  
showOmics, 54  
showOmics, Omics-method (Omics-class), 34  
showPathway, 55  
showPathway, MultiOmicsPathway-method  
    (MultiOmicsPathway-class), 26  
sparseCompPCs, 56  
stripModulesFromPathways, 56  
summarizeInCluster, 57  
summarizeOmicsResByMinPvalue, 58  
summarizeToBinaryDirectionalEvents, 59  
summarizeToBinaryEvents, 60  
summarizeToNumberOfDirectionalEvents,  
    61  
summarizeToNumberOfEvents, 62  
summarizeWithPca, 63  
survivalcox, 64  
survivalcoxr, 64

topoCompPCs, 65