Package 'RadioGx'

November 12, 2026

Type Package

Title Analysis of Large-Scale Radio-Genomic Data

Version 2.16.0

Date 2023-04-19

Description Computational tool box for radio-genomic analysis which integrates radio-response data, radio-biological modelling and comprehensive cell line annotations for hundreds of cancer cell lines. The 'RadioSet' class enables creation and manipulation of standardized datasets including information about cancer cells lines, radio-response assays and dose-response indicators. Included methods allow fitting and plotting dose-response data using established radio-biological models along with quality control to validate results. Additional functions related to fitting and plotting dose response curves, quantifying statistical correlation and calculating area under the curve (AUC) or survival fraction (SF) are included. For more details please see the included documentation, references, as well as:
Manem, V. et al (2018) <doi:10.1101/449793>.

License GPL-3

VignetteBuilder knitr

VignetteEngine knitr::rmarkdown

Encoding UTF-8

LazyData true

Depends R (>= 4.1), CoreGx

Imports SummarizedExperiment, BiocGenerics, data.table, S4Vectors, Biobase, parallel, BiocParallel, RColorBrewer, caTools, magicaxis, methods, reshape2, scales, grDevices, graphics, stats, utils, assertthat, matrixStats, downloader

Suggests rmarkdown, BiocStyle, knitr, pander, markdown

biocViews Software, Pharmacogenetics, QualityControl, Survival, Pharmacogenomics, Classification

RoxygenNote 7.2.3

Collate 'OERmodel.R' 'RadioSet-class.R' 'RadioSet-accessors.R'
'computeAUC.R' 'computeD10.R' 'computeSF2.R'
'computeSensitivity.R' 'datasets.R' 'doseResponseCurve.R'
'downloadRSet.R' 'geneDrugSensitivity.R' 'linearQuadratic.R'
'linearQuadraticInv.R' 'linearQuadraticModel.R'
'makeGrittyGuess.R' 'plotCurve.R' 'radSensitivitySig.R'
'rankGeneRadSensitivity.R' 'signatureClass.R'
'subsetTo-methods.R' 'summarizeMolecularProfiles-methods.R'
'summarizeSensitivityProfiles-methods.R'
'updateObject-methods.R' 'utilities.R' 'zzz.R'

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

git_branch RELEASE_3_23

git_last_commit 1b308e0

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Venkata Manem [aut],
Petr Smirnov [aut],
Ian Smith [aut],
Meghan Lambie [aut],
Christopher Eeles [aut],
Scott Bratman [aut],
Jermiah Joseph [aut],
Benjamin Haibe-Kains [aut, cre]

Maintainer Benjamin Haibe-Kains <benjamin.haibe.kains@utoronto.ca>

Contents

availableRSets . . . 3
checkRSetStructure . . . 4
clevelandSmall . . . 4
computeAUC . . . 5
computeD10 . . . 6
computeSF2 . . . 7
dim,RadioSet-method . . . 7
doseResponseCurve . . . 8
downloadRSet . . . 10
linearQuadraticModel . . . 11
plotCurve . . . 12
RadioSet . . . 13
RadioSet-accessors . . . 14
RadioSet-class . . . 23
RadioSig . . . 24
radSensitivitySig . . . 25
show,RadioSet-method . . . 27

show,RadioSig-method 27
showSigAnnot,RadioSig-method 28
subsetTo,RadioSet-method 29
summarizeMolecularProfiles,RadioSet-method 30
summarizeSensitivityProfiles,RadioSet-method 31
updateObject,RadioSet-method 32
[,RadioSet,ANY,ANY,ANY-method 33

Index **34**

`availableRSets` *Return a table of RadioSets available for download*

Description

The function fetches a table of all RadioSets available for download from the PharmacoGx server. The table includes the names of the PharamcoSet, the types of data available in the object, and the date of last update.

Usage

```
availableRSets(
  saveDir = tempdir(),
  fileName = "availableRadioSets.csv",
  verbose = TRUE
)
```

Arguments

`saveDir`	`character` Directory to save the table of rSets
`fileName`	`character` The filename for the table of rSets
`verbose`	`bool` Should status messages be printed during download.

Value

A data.frame with details about the available RadioSet objects

Examples

```
if (interactive()){
availableRSets()
}
```

checkRSetStructure *A function to verify the structure of a RadioSet*

Description

This function checks the structure of a PharamcoSet, ensuring that the correct annotations are in place and all the required slots are filled so that matching of cells and radiations can be properly done across different types of data and with other studies.

Usage

```
checkRSetStructure(object, plotDist = FALSE, result.dir = tempdir())
```

Arguments

object	A RadioSet object
plotDist	Should the function also plot the distribution of molecular data?
result.dir	The path to the directory for saving the plots as a string, defaults to 'tempdir()"

Value

Prints out messages whenever describing the errors found in the structure of the pset object passed in.

Examples

```
checkRSetStructure(clevelandSmall)
```

clevelandSmall *Cleaveland_mut RadioSet subsetted*

Description

Documentation for this dataset will be added at a later date. For now I just need this package to pass the CRAN checks! This dataset powers the exampe usage in the roxygen2 documentation for CoreGx.

Usage

```
data(clevelandSmall)
```

Format

RadioSet object

References

Lamb et al. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. Science, 2006.

`computeAUC` *computeAUC: computes AUC*

Description

This function computes the area under a dose-response curve of the form survival fraction SF = exp(-alpha * D - beta * D ^ 2).

Usage

```
computeAUC(
  D,
  SF,
  pars,
  lower,
  upper,
  trunc = TRUE,
  SF_as_log = FALSE,
  area.type = c("Fitted", "Actual"),
  verbose = TRUE
)
```

Arguments

`D`	vector of dosages
`SF`	vector of survival fractions
`pars`	parameters (alpha, beta) in equation y = exp(-alpha * x - beta * x ^ 2)
`lower`	lower bound of dose region to compute AUC over
`upper`	upper bound of dose region to compute AUC over
`trunc`	should survival fractions be truncated downward to 1 if they exceed 1?
`SF_as_log`	A boolean indicating whether survival fraction is displayed on a log axis. Defaults to FALSE
`area.type`	should the AUC of the raw (D, SF) points be returned, or should the AUC of a curve fit to said points be returned instead?
`verbose`	how detailed should error and warning messages be? See details.

Details

If lower and/or upper are missing, the function assumes their values to be the minimum and maximum D-values, respectively. For all warnings to be silent, set trunc = FALSE. For warnings to be output, set trunc = TRUE. For warnings to be output along with the arguments that triggered them, set trunc = 2.

Value

numeric The area under the ROC curve

Examples

```
computeAUC(D=c(0.1, 0.5, 0.7, 0.9), pars=c(0.2, 0.1), lower = 0,
 upper = 1) # Returns 0.7039296
```

---

computeD10 *Compute D10*

Description

This function computes the radiation dose at which only 10 cancer cells survive under the exponential model SF = exp(-alpha * D - beta * D ^ 2) given alpha and beta, where D is the radiation dose given and SF is the fraction of cells surviving

Usage

```
computeD10(pars)
```

Arguments

pars	parameters (alpha, beta) in equation y = exp(-alpha * x - beta * x ^ 2)

Details

The units of the returned dose are the inverses of the units of the alpha and beta passed in.

Value

numeric The D10 value

Examples

```
computeD10(c(0.2, 0.1))
```

computeSF2 *Compute SF2*

Description

This function computes the survival fraction after administering 2 units of radiation, given alpha and beta in the equation SF = exp(-alpha * D - beta * D ^ 2).

Usage

```
computeSF2(pars)
```

Arguments

pars	parameters (alpha, beta) in equation y = exp(-alpha * x - beta * x ^ 2)

Value

numeric The survival fraction

Examples

```
computeSF2(c(0.2, 0.1))
```

dim,RadioSet-method *Get the dimensions of a RadioSet*

Description

Get the dimensions of a RadioSet

Usage

```
## S4 method for signature 'RadioSet'
dim(x)
```

Arguments

x	RadioSet

Value

A named vector with the number of Cells and Drugs in the RadioSet

Examples

```
data(clevelandSmall)
dim(clevelandSmall)
```

`doseResponseCurve` *Plot drug response curve of a given drug and a given cell for a list of rSets (objects of the RadioSet class).*

Description

Given a list of RadioSets, the function will plot the drug_response curve, for a given drug/cell pair. The y axis of the plot is the viability percentage and x axis is the log transformed Ds. If more than one rSet is provided, a light gray area would show the common concentration range between rSets. User can ask for type of sensitivity measurment to be shown in the plot legend. The user can also provide a list of their own Ds and viability values, as in the examples below, and it will be treated as experiments equivalent to values coming from a pset. The names of the concentration list determine the legend labels.

Usage

```
doseResponseCurve(
  rad.type = "radiation",
  cellline,
  rSets = list(),
  Ds = list(),
  SFs = list(),
  trunc = TRUE,
  legends.label = c("alpha", "beta", "rsquared"),
  ylim = c(0, 100),
  xlim,
  mycol,
  title,
  plot.type = c("Fitted", "Actual", "Both"),
  summarize.replicates = TRUE,
  lwd = 1,
  cex = 0.7,
  cex.main = 0.9,
  legend.loc = "topright",
  verbose = TRUE
)
```

Arguments

`rad.type`	‘character(1)‘ The type(s) of radiation dosage to be plotted. If the plot is desirable for more than one radioset, A unique drug id should be provided.

cellline	'character(1)' A cell line name for which the radiation response curve should be plotted. If the plot is desirable for more than one radioset, a unique cell id should be provided.
rSets	'list' a list of RadioSet objects, for which the function should plot the curves.
Ds, SFs	'list' A list of Doses and SFs to plot, the function assumes that Ds[[i]] is plotted against SFs[[i]]. The names of the D list are used to create the legend labels
trunc	'logical(1)' Should the viability values be truncated to lie in [0-100] before doing the fitting
legends.label	'numeric' A vector of sensitivity measurment types which could be any combination of ic50_published, auc_published, auc_recomputed and auc_recomputed_star. A legend will be displayed on the top right of the plot which each line of the legend is the values of requested sensitivity measerments for one of the requested rSets. If this parameter is missed no legend would be provided for the plot.
ylim	'numeric' A vector of two numerical values to be used as ylim of the plot. If this parameter would be missed c(0,100) would be used as the ylim of the plot.
xlim	'numeric' A vector of two numerical values to be used as xlim of the plot. If this parameter would be missed the minimum and maximum concentrations between all the rSets would be used as plot xlim.
mycol	'numeric' A vector with the same lenght of the rSets parameter which will determine the color of the curve for the pharmaco sets. If this parameter is missed default colors from Rcolorbrewer package will be used as curves color.
title	'character' The title of the graph. If no title is provided, then it defaults to Drug':'Cell Line'.
plot.type	'character' Plot type which can be the actual one ("Actual") or the one fitted by logl logistic regression ("Fitted") or both of them ("Both"). If this parameter is missed by default actual curve is plotted.
summarize.replicates	'character' If this parameter is set to true replicates are summarized and replicates are plotted individually otherwise
lwd	'numeric' The line width to plot with
cex	'numeric' The cex parameter passed to plot
cex.main	'numeric' The cex.main parameter passed to plot, controls the size of the titles
legend.loc	And argument passable to xy.coords for the position to place the legend.
verbose	'logical(1)' Should warning messages about the data passed in be printed?

Value

Plots to the active graphics device and returns and invisible NULL.

Examples

```
doseResponseCurve(Ds=list("Experiment 1" = c(0, 2, 4, 6)),
  SFs=list("Experiment 1" = c(1,.6,.4,.2)), plot.type="Both")
```

`downloadRSet` *Download a RadioSet object*

Description

This function allows you to download a `RadioSet` object for use with this package. The `RadioSets` have been extensively curated and organized within a PharacoSet class, enabling use with all the analysis tools provided in `PharmacoGx`.

Usage

```
downloadRSet(
  name,
  saveDir = tempdir(),
  rSetFileName = NULL,
  verbose = TRUE,
  timeout = 600
)
```

Arguments

`name`	`Character` string, the name of the PhamracoSet to download.
`saveDir`	`Character` string with the folder path where the RadioSet should be saved. Defaults to `'./rSets/'`. Will create directory if it does not exist.
`rSetFileName`	`character` string, the file name to save the dataset under
`verbose`	`bool` Should status messages be printed during download. Defaults to TRUE.
`timeout`	'numeric(1)' How long to wait before erroring due to timeout, in seconds. Default it 600s (10 minutes).

Value

A rSet object with the dataset, downloaded from our server

Examples

```
if (interactive()) {
cleveland <- downloadRSet("Cleveland")
}
```

`linearQuadraticModel` *Fit linear-quadratic curves to dose-response data*

Description

This function fits a linear-quadratic curve to dose-response data.

Usage

```
linearQuadraticModel(
  D,
  SF,
  lower_bounds = c(0, 0),
  upper_bounds = c(1, 1),
  scale = 5,
  family = c("normal", "Cauchy"),
  median_n = 1,
  trunc = FALSE,
  verbose = FALSE
)
```

Arguments

`D`	vector of radiation doses
`SF`	vector of survival fractions corresponding to the doses
`lower_bounds`	vector of length 2 containing minimum allowed values of fitted alpha and beta, respectively
`upper_bounds`	vector of length 2 containing maximum allowed values of fitted alpha and beta, respectively
`scale`	parameter of the assumed error distribution of the data; see sdetails
`family`	family of distributions of the error terms in the data; currently supported options are "normal" and "cauchy"
`median_n`	see details
`trunc`	should survival fractions be truncated downward to 1? Defaults to FALSE.
`verbose`	'verbose' outputs warnings that are otherwised suppressed when the function sanity-checks user inputs. 'median_n' denotes the number of distributions from family 'family' that are medianned. (Note that setting n = 1 (the default) is equivalent to using a simple normal or cauchy distribution without taking any medians.)

Value

`numeric` The estimated alpha and beta values

Examples

```
linearQuadraticModel(c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
 c(1.1, 0.8, 0.7, 0.45, 0.15, -0.1, -0.1, -0.4, -0.65, -0.75, -1.1))
```

---

plotCurve *Plot radiation dose-response curve*

Description

This function plots doses of radiation against the cancer cell survival fractions thereby observed.

Usage

```
plotCurve(
  D,
  SF,
  pars,
  filename = "dose_response_plot.pdf",
  fit_curve = TRUE,
  SF_as_log = TRUE
)
```

Arguments

D	vector of radiation doses
SF	vector of survival fractions corresponding to the doses
pars	parameters (alpha, beta) in the equation SF = exp(-alpha * D - beta * D ^ 2)
filename	name of PDF which will be created by the function
fit_curve	should the graph include a linear-quadratic curve of best fit? Defaults to TRUE
SF_as_log	should SF be expressed in log10 on the graph? Defaults to TRUE

Value

nothing Function works by side effects only

Examples

```
plotCurve(c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
  c(1.1, 0.8, 0.7, 0.45, 0.15, -0.1, -0.1, -0.4, -0.65, -0.75, -1.1),
  filename = NULL)
```

RadioSet *RadioSet constructor*

Description

A constructor that simplifies the process of creating RadioSets, as well as creates empty objects for data not provided to the constructor. Only objects returned by this constructor are expected to work with the RadioSet methods. For a much more detailed instruction on creating RadioSets, please see the "CreatingRadioSet" vignette.

Usage

```
RadioSet(
  name,
  molecularProfiles = list(),
  sample = data.frame(),
  treatment = data.frame(),
  sensitivityInfo = data.frame(),
  sensitivityRaw = array(dim = c(0, 0, 0)),
  sensitivityProfiles = matrix(),
  sensitivityN = matrix(nrow = 0, ncol = 0),
  perturbationN = array(NA, dim = c(0, 0, 0)),
  curationSample = data.frame(),
  curationTissue = data.frame(),
  curationTreatment = data.frame(),
  datasetType = c("sensitivity", "perturbation", "both"),
  verify = TRUE
)
```

Arguments

name	A character string detailing the name of the dataset
molecularProfiles	A list of SummarizedExperiment objects containing molecular profiles for each molecular data type.
sample	A data.frame containing the annotations for all the sample profiled in the data set, across all data types. Must contain the mandatory sampleid column which uniquely identifies each sample in the object.
treatment	A data.frame containing annotations for all treatments profiled in the dataset. Must contain the mandatory treatmentid column which uniquely identifies each treatment in the object.
sensitivityInfo	A data.frame containing the information for the sensitivity experiments. Must contain a 'sampleid' column with unique identifiers to each sample, matching the sample object and a 'treatmentid' columns with unique indenifiers for each treatment, matching the treatment object.

`sensitivityRaw` A 3 Dimensional `array` contaning the raw drug dose response data for the sensitivity experiments

`sensitivityProfiles`
`data.frame` containing drug sensitivity profile statistics such as IC50 and AUC

`sensitivityN`, `perturbationN`
A `data.frame` summarizing the available sensitivity/perturbation data

`curationSample`, `curationTissue`, `curationTreatment`
A `data.frame` mapping the names for samples, tissues and treatments used in the data set to universal identifiers used between different CoreSet objects

`datasetType` A `character(1)` string of 'sensitivity', 'preturbation', or 'both' detailing what type of data can be found in the `CoreSet`, for proper processing of the data

`verify` `logical(1)`Should the function verify the CoreSet and print out any errors it finds after construction?

Value

An object of class 'RadioSet"

`RadioSet-accessors` *Accessing and modifying information in a 'RadioSet'*

Description

Documentation for the various setters and getters which allow manipulation of data in the slots of a 'RadioSet' object.

Usage

```
radiationInfo(...)

radiationInfo(...) <- value

radiationTypes(...)

radiationTypes(...) <- value

## S4 method for signature 'RadioSet'
annotation(object)

## S4 replacement method for signature 'RadioSet,list'
annotation(object) <- value

## S4 method for signature 'RadioSet'
dateCreated(object)

## S4 replacement method for signature 'RadioSet,character'
```

```
dateCreated(object) <- value

## S4 method for signature 'RadioSet'
name(object)

## S4 replacement method for signature 'RadioSet'
name(object) <- value

## S4 method for signature 'RadioSet'
sampleInfo(object)

## S4 replacement method for signature 'RadioSet,data.frame'
sampleInfo(object) <- value

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

## S4 replacement method for signature 'RadioSet,character'
sampleNames(object) <- value

## S4 method for signature 'RadioSet'
curation(object)

## S4 replacement method for signature 'RadioSet,list'
curation(object) <- value

## S4 method for signature 'RadioSet'
datasetType(object)

## S4 replacement method for signature 'RadioSet,character'
datasetType(object) <- value

## S4 method for signature 'RadioSet'
molecularProfiles(object, mDataType, assay)

## S4 replacement method for signature 'RadioSet,character,character,matrix'
molecularProfiles(object, mDataType, assay) <- value

## S4 method for signature 'RadioSet'
featureInfo(object, mDataType)

## S4 replacement method for signature 'RadioSet,character,data.frame'
featureInfo(object, mDataType) <- value

## S4 method for signature 'RadioSet,character'
phenoInfo(object, mDataType)

## S4 replacement method for signature 'RadioSet,character,data.frame'
```

```
phenoInfo(object, mDataType) <- value

## S4 method for signature 'RadioSet,character'
fNames(object, mDataType)

## S4 replacement method for signature 'RadioSet,character,character'
fNames(object, mDataType) <- value

## S4 method for signature 'RadioSet'
mDataNames(object)

## S4 replacement method for signature 'RadioSet'
mDataNames(object) <- value

## S4 method for signature 'RadioSet'
molecularProfilesSlot(object)

## S4 replacement method for signature 'RadioSet,list_OR_MAE'
molecularProfilesSlot(object) <- value

## S4 method for signature 'RadioSet'
sensitivityInfo(object, dimension, ...)

## S4 replacement method for signature 'RadioSet,data.frame'
sensitivityInfo(object, dimension, ...) <- value

## S4 method for signature 'RadioSet'
sensitivityMeasures(object)

## S4 replacement method for signature 'RadioSet,character'
sensitivityMeasures(object) <- value

## S4 method for signature 'RadioSet'
sensitivityProfiles(object)

## S4 replacement method for signature 'RadioSet,data.frame'
sensitivityProfiles(object) <- value

## S4 method for signature 'RadioSet'
sensitivityRaw(object)

## S4 replacement method for signature 'RadioSet,array'
sensitivityRaw(object) <- value

## S4 method for signature 'RadioSet'
treatmentResponse(object)

## S4 replacement method for signature 'RadioSet,list_OR_LongTable'
```

```
treatmentResponse(object) <- value

## S4 method for signature 'RadioSet'
sensNumber(object)

## S4 replacement method for signature 'RadioSet,matrix'
sensNumber(object) <- value

## S4 method for signature 'RadioSet'
pertNumber(object)

## S4 replacement method for signature 'RadioSet,array'
pertNumber(object) <- value
```

Arguments

...	See details.
value	See details.
object	A 'RadioSet' object.
mDataType	'character(1)' The name of a molecular datatype to access from the 'molecularProfiles' of a 'RadioSet' object.
assay	'character(1)' A valid assay name in the 'SummarizedExperiment' of '@molecularProfiles' of a RadioSet object for data type 'mDataType'.
dimension	See details.

Details

treatmentInfo: `data.frame` Metadata for all treatments in a `RadioSet` object. Arguments:

- object: `RadioSet` An object to retrieve treatment metadata from.

treatmentInfo<-: `RadioSet` object with updated treatment metadata. object. Arguments:

- object: `RadioSet` An object to set treatment metadata for.
- value: `data.frame` A new table of treatment metadata for `object`.

treatmentNames: `character` Names for all treatments in a `RadioSet` object. Arguments:

- object: `RadioSet` An object to retrieve treatment names from.

treatmentNames<-: `RadioSet` Object with updates treatment names. object. Arguments:

- object: `RadioSet` An object to set treatment names from.
- value: `character` A character vector of updated treatment names.

@annotation:

annotation: A `list` of RadioSet annotations with items: 'name', the name of the object; 'dateCreated', date the object was created; 'sessionInfo', the `sessionInfo()` when the object was created; 'call', the R constructor call; and 'version', the object version.

annotation<-: Setter method for the annotation slot. Arguments:

- value: a `list` of annotations to update the RadioSet with.

@dateCreated:
dateCreated: `character(1)` The date the `RadioSet` object was created, as returned by the `date()` function.

dateCreated<-: Update the 'dateCreated' item in the `annotation` slot of a `RadioSet` object. Arguments:

- value: A `character(1)` vector, as returned by the `date()` function.

name: `character(1)` The name of the `RadioSet`, retreived from the `@annotation` slot.

name<-: Update the `@annotation$name` value in a `RadioSet` object.

- value: `character(1)` The name of the `RadioSet` object.

cellInfo: `data.frame` Metadata for all sample in a `RadioSet` object.

sampleInfo<-: assign updated sample annotations to the `RadioSet` object. Arguments:

- value: a `data.frame` object.

sampleNames: `character` Retrieve the rownames of the `data.frame` in the `sample` slot from a RadioSet object.

sampleNames<-: assign new rownames to the sampleInfo `data.frame` for a RadioSet object. Arguments:

- value: `character` vector of rownames for the `sampleInfo(object)` `data.frame`.

@curation:
curation: A `list` of curated mappings between identifiers in the RadioSet object and the original data publication. Contains three `data.frames`, 'cell' with cell-line ids and 'tissue' with tissue ids and 'radiation' with radiation ids.

curation<-: Update the `curation` slot of a RadioSet object. Arugments:

- value: A `list` of `data.frames`, one for each type of curated identifier. For a `RadioSet` object the slot should contain tissue, cell-line and radiation id `data.frames`.

datasetType slot:
datasetType: `character(1)` The type treatment response in the `sensitivity` slot. Valid values are 'sensitivity', 'perturbation' or 'both'.

datasetType<-: Update the datasetType slot of a RadioSet object. Arguments:

- value: A `character(1)` vector with one of 'sensitivity', 'perturbation' or 'both'

@molecularProfiles:
molecularProfiles: `matrix()` Retrieve an assay in a `SummarizedExperiment` from the `molecularProfiles` slot of a `RadioSet` object with the specified `mDataType`. Valid `mDataType` arguments can be found with `mDataNames(object)`. Exclude `mDataType` and `assay` to access the entire slot. Arguments:

- assay: Optional `character(1)` vector specifying an assay in the `SummarizedExperiment` of the `molecularProfiles` slot of the `RadioSet` object for the specified `mDataType`. If excluded, defaults to modifying the first assay in the `SummarizedExperiment` for the given `mDataType`.

molecularProfiles<-: Update an assay in a `SummarizedExperiment` from the `molecularProfiles` slot of a RadioSet object with the specified `mDataType`. Valid `mDataType` arguments can be found with `mDataNames(object)`. Omit `mDataType` and `assay` to update the slot.

- assay: Optional `character(1)` vector specifying an assay in the `SummarizedExperiment` of the `molecularProfiles` slot of the `RadioSet` object for the specified `mDataType`. If excluded, defaults to modifying the first assay in the `SummarizedExperiment` for the given `mDataType`.
- value: A `matrix` of values to assign to the `assay` slot of the `SummarizedExperiment` for the selected `mDataType`. The rownames and column names must match the associated `SummarizedExperiment`.

featureInfo: Retrieve a `DataFrame` of feature metadata for the specified `mDataType` from the `molecularProfiles` slot of a `RadioSet` object. More specifically, retrieve the `@rowData` slot from the `SummarizedExperiment` from the `@molecularProfiles` of a `RadioSet` object with the name `mDataType`.

__featureInfo<-__: Update the ‘featureInfo(object, mDataType)‘ ‘DataFrame‘ with new feature metadata. Arguments: - value: A ‘data.frame‘ or ‘DataFrame‘ with updated feature metadata for the specified molecular profile in the ‘molecularProfiles‘ slot of a ‘RadioSet‘ object.

phenoInfo: Return the `@colData` slot from the `SummarizedExperiment` of `mDataType`, containing sample-level metadata, from a `RadioSet` object.

phenoInfo<-: Update the `@colData` slot of the `SummarizedExperiment` of `mDataType` in the `@molecularProfiles` slot of a `RadioSet` object. This updates the sample-level metadata in-place.

- value: A `data.frame` or `DataFrame` object where rows are samples and columns are sample metadata.

fNames: `character()` The features names from the `rowData` slot of a `SummarizedExperiment` of `mDataType` within a `RadioSet` object.

fNames: Updates the rownames of the feature metadata (i.e., `rowData`) for a `SummarizedExperiment` of `mDataType` within a `RadioSet` object.

- value: `character()` A character vector of new features names for the `rowData` of the `SummarizedExperiment` of `mDataType` in the `@molecularProfiles` slot of a `RadioSet` object. Must be the same length as `nrow(featureInfo(object, mDataType))`, the number of rows in the feature metadata.

mDataNames: `character` Retrieve the names of the molecular data types available in the `molecularProfiles` slot of a `RadioSet` object. These are the options which can be used in the `mDataType` parameter of various `molecularProfiles` slot accessors methods.

mDataNames: Update the molecular data type names of the `molecularProfiles` slot of a RadioSet object. Arguments:

- value: `character` vector of molecular datatype names, with length equal to `length(molecularProfilesSlot(object`

molecularProfilesSlot: Return the contents of the `@molecularProfiles` slot of a `RadioSet` object. This will either be a `list` or `MultiAssayExperiment` of `SummarizedExperiments`.

molecularProfilesSlot<-: Update the contents of the `@molecularProfiles` slot of a `RadioSet` object. Arguemnts:

- value: A `list` or `MultiAssayExperiment` of `SummarizedExperiments`. The `list` and `assays` should be named for the molecular datatype in each `SummarizedExperiment`.

@treatmentResponse:

Arguments::

- `dimension`: Optional `character(1)` One of 'treatment', 'sample' or 'assay' to retrieve `rowData`, `colData` or the 'assay_metadata' assay from the `RadioSet` `@sensitvity` `LongTable` object, respectively. Ignored with warning if `@treatmentResponse` is not a `LongTable` object.
- `...`: Additional arguments to the `rowData` or `colData`. `LongTable` methods. Only used if the sensitivity slot contains a `LongTable` object instead of a `list` and the `dimension` argument is specified.

Methods::
sensitivityInfo: `DataFrame` or `data.frame` of sensitivity treatment combo by sample metadata for the `RadioSet` object. When the `dimension` parameter is used, it allows retrieval of the dimension specific metadata from the `LongTable` object in `@treatmentResponse` of a RadioSet object.

sensitivityInfo<-: Update the `@treatmentResponse` slot metadata for a `RadioSet` object. When used without the `dimension` argument is behaves similar to the old RadioSet implementation, where the `@treatmentResponse` slot contained a list with a `$info` `data.frame` item. When the `dimension` arugment is used, more complicated assignments can occur where 'sample' modifies the `@sensitvity` `LongTable` colData, 'treatment' the rowData and 'assay' the 'assay_metadata' assay. Arguments:

- value: A `data.frame` of treatment response experiment metadata, documenting experiment level metadata (mapping to treatments and samples). If the `@treatmentResponse` slot doesn't contain a `LongTable` and `dimension` is not specified, you can only modify existing columns as returned by `sensitivityInfo(object)`.

sensitivityMeaures: Get the 'sensitivityMeasures' available in a `RadioSet` object. Each measure reprents some summary of sample sensitivity to a given treatment, such as ic50, ec50, AUC, AAC, etc. The results are returned as a `character` vector with all available metrics for the PSet object.

sensitivityMeaures: Update the sensitivity meaure in a `RadioSet` object. Thesee values are the column names of the 'profiles' assay and represent various compued sensitviity metrics such as ic50, ec50, AUC, AAC, etc.

- value: A `character` vector of new sensitivity measure names, the then length of the character vector must matcht he number of columns of the 'profiles' assay, excluding metadata and key columns.

sensitivityProfiles: Return the sensitivity profile summaries from the sensitivity slot. This data.frame cotanins vaarious sensitivity summary metrics, such as ic50, amax, EC50, aac, HS, etc as columns, with rows as treatment by sample experiments.

sensitivityProfiles<-: Update the sensitivity profile summaries the sensitivity slot. Arguments: - value: A `data.frame` the the same number of rows as as returned by `sensitivityProfiles(object)`, but potentially modified columns, such as the computation of additional summary metrics.

sensitivityRaw: Access the raw sensitiity measurents for a RadioSet object. A 3D `array` where rows are experiment_ids, columns are doses and the third dimension is metric, either 'Dose' for the doses used or 'Viability' for the sample viability at that dose.

sensitvityRaw<-: Update the raw dose and viability data in a `RadioSet`.

- value: A 3D `array` object where rows are experiment_ids, columns are replicates and pages are c('Dose', 'Viability'), with the corresponding dose or viability measurement for that experiment_id and replicate.

sensNumber: Return a count of viability observations in a `RadioSet` object for each treatment-combo by sample combination.

sensNumber<-: Update the 'n' item, which holds a matrix with a count of treatment by sample-line experiment counts, in the `list` in `@treatmentResponse` slot of a `RadioSet` object. Will error when `@sensitviity` contains a `LongTable` object, since the counts are computed on the fly. Arguments:

- value: A `matrix` where rows are samples and columns are treatments, with a count of the number of experiments for each combination as the values.

pertNumber: `array` Summary of available perturbation experiments from in a `RadioSet` object. Returns a 3D `array` with the number of perturbation experiments per treatment and sample, and data type.

pertNumber<-: Update the `@perturbation$n` value in a `RadioSet` object, which stores a summary of the available perturbation experiments. Arguments:

- value: A new 3D `array` with the number of perturbation experiments per treatment and sample, and data type

Value

Accessors: See details.

Setters: An updated 'RadioSet' object, returned invisibly.

Examples

```
data(clevelandSmall)
treatmentInfo(clevelandSmall)

treatmentInfo(clevelandSmall) <- treatmentInfo(clevelandSmall)

treatmentNames(clevelandSmall)

treatmentNames(clevelandSmall) <- treatmentNames(clevelandSmall)


## @annotation

annotation(clevelandSmall)

annotation(clevelandSmall) <- annotation(clevelandSmall)

dateCreated(clevelandSmall)
```

```
## dateCreated
dateCreated(clevelandSmall) <- date()

name(clevelandSmall)

name(clevelandSmall) <- 'new_name'

sampleInfo(clevelandSmall) <- sampleInfo(clevelandSmall)

sampleNames(clevelandSmall)

sampleNames(clevelandSmall) <- sampleNames(clevelandSmall)

## curation
curation(clevelandSmall)

curation(clevelandSmall) <- curation(clevelandSmall)

datasetType(clevelandSmall)

datasetType(clevelandSmall) <- 'both'

# No assay specified
molecularProfiles(clevelandSmall, 'rna') <- molecularProfiles(clevelandSmall, 'rna')

# Specific assay
molecularProfiles(clevelandSmall, 'rna', 'exprs') <-
    molecularProfiles(clevelandSmall, 'rna', 'exprs')

# Replace the whole slot
molecularProfiles(clevelandSmall) <- molecularProfiles(clevelandSmall)

featureInfo(clevelandSmall, 'rna')

featureInfo(clevelandSmall, 'rna') <- featureInfo(clevelandSmall, 'rna')

phenoInfo(clevelandSmall, 'rna')

phenoInfo(clevelandSmall, 'rna') <- phenoInfo(clevelandSmall, 'rna')

fNames(clevelandSmall, 'rna')

fNames(clevelandSmall, 'rna') <- fNames(clevelandSmall, 'rna')

mDataNames(clevelandSmall)

mDataNames(clevelandSmall) <- mDataNames(clevelandSmall)

molecularProfilesSlot(clevelandSmall)

molecularProfilesSlot(clevelandSmall) <- molecularProfilesSlot(clevelandSmall)
```

```
sensitivityInfo(clevelandSmall)

sensitivityInfo(clevelandSmall) <- sensitivityInfo(clevelandSmall)

sensitivityMeasures(clevelandSmall) <- sensitivityMeasures(clevelandSmall)

sensitivityMeasures(clevelandSmall) <- sensitivityMeasures(clevelandSmall)

sensitivityProfiles(clevelandSmall)

sensitivityProfiles(clevelandSmall) <- sensitivityProfiles(clevelandSmall)

head(sensitivityRaw(clevelandSmall))

sensitivityRaw(clevelandSmall) <- sensitivityRaw(clevelandSmall)

treatmentResponse(clevelandSmall)

treatmentResponse(clevelandSmall) <- treatmentResponse(clevelandSmall)

sensNumber(clevelandSmall)

sensNumber(clevelandSmall) <- sensNumber(clevelandSmall)

pertNumber(clevelandSmall)

pertNumber(clevelandSmall) <- pertNumber(clevelandSmall)
```

`RadioSet-class` *A Class to Contain RadioGenomic datasets together with their curations*

Description

The RadioSet (RSet) class was developed to contain and organise large RadioGenomic datasets, and aid in their metanalysis. It was designed primarily to allow bioinformaticians and biologists to work with data at the level of genes and cell lines, providing a more naturally intuitive interface and simplifying analyses between several datasets. As such, it was designed to be flexible enough to hold datasets of two different natures while providing a common interface. The class can accomidate datasets containing both radiation dose response data, as well as datasets contaning genetic profiles of cell lines pre and post treatement with compounds, known respecitively as sensitivity and perturbation datasets.

Value

An object of the RadioSet class

Slots

`annotation` A `list` of annotation data about the RadioSet, including the `$name` and the session information for how the object was creating, detailing the exact versions of R and all the packages used

`molecularProfiles` A `list` containing 4 `SummarizedExperiment` type object for holding data for RNA, DNA, SNP and Copy Number Variation measurements respectively, with associated `fData` and `pData` containing the row and column metadata

`sample` A `data.frame` containg the annotations for all the cell lines profiled in the data set, across all data types

`treatment` A `data.frame` containg the annotations for all the radiation treatment types used in the in the dataset, across all data types

`sensitivity` A `list` containing all the data for the sensitivity experiments, including `$info`, a `data.frame` containing the experimental info,`$raw` a 3D `array` containing raw data, `$profiles`, a `data.frame` containing sensitivity profiles statistics, and `$n`, a `data.frame` detailing the number of experiments for each cell-radiation type pair

`perturbation` A `list` containting `$n`, a `data.frame` summarizing the available perturbation data,

`curation` A `list` containing mappings for `cell` and `tissue` names used in the data set to universal identifiers used between different RadioSet objects

`datasetType` A `character` string of 'sensitivity', 'perturbation', or both detailing what type of data can be found in the RadioSet, for proper processing of the data

---

`RadioSig` *Radiation Signature Class Constructor*

Description

A documented constructor to provide user friendly interface to .RadioSig

Usage

```
RadioSig(
  Data = array(NA, dim = c(0, 0, 0)),
  RSetName = "",
  DateCreated = date(),
  SigType = "sensitivity",
  SessionInfo = sessionInfo(),
  Call = "No Call Recorded"
)
```

Arguments

`Data`	The data
`RSetName`	The name of the pSet
`DateCreated`	The date the object was created

SigType	The type of sensitivyt signature
SessionInfo	The package version used to generate the object
Call	The calls for sensitivity vs not

Value

A RadioSig object

---

radSensitivitySig *Creates a signature representing the association between gene expression (or other molecular profile) and radiation dose response, for use in radiation sensitivity analysis.*

Description

Given a RadioSet of the sensitivity experiment type, and a list of radiation types, the function will compute a signature for the effect of gene expression on the molecular profile of a cell. The function returns the estimated coefficient, the t-stat, the p-value and the false discovery rate associated with that coefficient, in a 3 dimensional array, with genes in the first direction, drugs in the second, and the selected return values in the third.

Usage

```
radSensitivitySig(
  rSet,
  mDataType,
  radiation.types,
  features,
  sensitivity.measure = "AUC_recomputed",
  molecular.summary.stat = c("mean", "median", "first", "last", "or", "and"),
  sensitivity.summary.stat = c("mean", "median", "first", "last"),
  returnValues = c("estimate", "pvalue", "fdr"),
  sensitivity.cutoff = NA,
  standardize = c("SD", "rescale", "none"),
  nthread = 1,
  verbose = TRUE,
  ...
)
```

Arguments

rSet	A RadioSet of the perturbation experiment type
mDataType	character which one of the molecular data types to use in the analysis, out of dna, rna, rnaseq, snp, cnv
radiation.types	character a vector of radiation.types for which to compute the signatures. Should match the names used in the PharmacoSet.

`features` `character` a vector of features for which to compute the signatures. Should match the names used in correspondant molecular data in PharmacoSet.

`sensitivity.measure`
`character` which measure of the radiation sensitivity should the function use for its computations? Use the sensitivityMeasures function to find out what measures are available for each PSet.

`molecular.summary.stat`
What summary statistic should be used to summarize duplicates for cell line molecular profile measurements?

`sensitivity.summary.stat`
What summary statistic should be used to summarize duplicates for cell line sensitivity measurements?

`returnValues` `character` Which of estimate, t-stat, p-value and fdr should the function return for each gene?

`sensitivity.cutoff`
Allows to provide upper and lower bounds to sensitivity measures in the cases where the values exceed physical values due to numerical or other errors.

`standardize` `character` One of "SD", "rescale", or "none", for the form of standardization of the data to use. If "SD", the the data is scaled so that SD = 1. If rescale, then the data is scaled so that the 95 interquantile range lies in [0,1]. If none no rescaling is done.

`nthread` `numeric` if multiple cores are available, how many cores should the computation be parallelized over?

`verbose` `boolean` 'TRUE' if the warnings and other infomrative message shoud be displayed

`...` additional arguments not currently fully supported by the function

Value

`list` a 3D array with genes in the first dimension, radiation.types in the second, and return values in the third.

Examples

```
data(clevelandSmall)
rad.sensitivity <- radSensitivitySig(clevelandSmall, mDataType="rna",
            nthread=1, features = fNames(clevelandSmall, "rna")[1],
            radiation.types=treatmentNames(clevelandSmall))
print(rad.sensitivity)
```

show,RadioSet-method *Show a RadioSet*

Description

Show a RadioSet

Usage

```
## S4 method for signature 'RadioSet'
show(object)
```

Arguments

object	A RadioSet object

Value

Prints the RadioSet object to the output stream, and returns invisible NULL.

Examples

```
data(clevelandSmall)
clevelandSmall
```

show,RadioSig-method *Show RadioGx Signatures*

Description

Show RadioGx Signatures

Usage

```
## S4 method for signature 'RadioSig'
show(object)
```

Arguments

object	RadioSig

Value

Prints the RadioGx Signatures object to the output stream, and returns invisible NULL.

Examples

```
data(clevelandSmall)
rad.sensitivity <- radSensitivitySig(clevelandSmall, mDataType="rna",
             nthread=1, features = fNames(clevelandSmall, "rna")[1])
rad.sensitivity
```

`showSigAnnot,RadioSig-method`

Show the Annotations of a signature object

Description

This funtion prints out the information about the call used to compute the rad signatures, and the session info for the session in which the computation was done. Useful for determining the exact conditions used to generate signatures.

Usage

```
## S4 method for signature 'RadioSig'
showSigAnnot(object)
```

Arguments

`object`	An object of the `RadioSig` Class, as returned by `radPerturbationSig` or `radSensitivitySig`

Value

Prints the RadioGx Signatures annotations to the output stream, and returns invisible NULL.

Examples

```
data(clevelandSmall)
rad.sensitivity <- radSensitivitySig(clevelandSmall, mDataType="rna",
             nthread=1, features = fNames(clevelandSmall, "rna")[1])
showSigAnnot(rad.sensitivity)
```

`subsetTo,RadioSet-method` *A function to subset a RadioSet to data containing only specified radiations, cells and genes*

Description

This is the prefered method of subsetting a RadioSet. This function allows abstraction of the data to the level of biologically relevant objects: radiations and cells. The function will automatically go through all of the combined data in the RadioSet and ensure only the requested radiations and cell lines are found in any of the slots. This allows quickly picking out all the experiments for a radiation or cell of interest, as well removes the need to keep track of all the metadata conventions between different datasets.

Usage

```
## S4 method for signature 'RadioSet'
subsetTo(
  object,
  cells = NULL,
  radiationTypes = NULL,
  molecular.data.cells = NULL,
  keep.controls = TRUE,
  ...
)
```

Arguments

`object`	A `RadioSet` to be subsetted
`cells`	A list or vector of cell names as used in the dataset to which the object will be subsetted. If left blank, then all cells will be left in the dataset.
`radiationTypes`	A list or vector of radiation names as used in the dataset to which the object will be subsetted. If left blank, then all radiationTypes will be left in the dataset.
`molecular.data.cells`	A list or vector of cell names to keep in the molecular data
`keep.controls`	If the dataset has perturbation type experiments, should the controls be kept in the dataset? Defaults to true.
`...`	Other arguments passed by other function within the package

Value

A RadioSet with only the selected radiation types and cells

Examples

```
clevelandRadiationTypes  <- treatmentNames(clevelandSmall)
clevelandCells <- sampleNames(clevelandSmall)
RSet <- subsetTo(clevelandSmall, radiationTypes = clevelandRadiationTypes[1],
  cells = clevelandCells[1])
RSet
```

`summarizeMolecularProfiles,RadioSet-method`
Takes molecular data from a RadioSet, and summarises them into one entry per drug

---

Description

Given a RadioSet with molecular data, this function will summarize the data into one profile per cell line, using the chosed summary.stat. Note that this does not really make sense with perturbation type data, and will combine experiments and controls when doing the summary if run on a perturbation dataset.

Usage

```
## S4 method for signature 'RadioSet'
summarizeMolecularProfiles(
  object,
  mDataType,
  cell.lines,
  features,
  summary.stat = c("mean", "median", "first", "last", "and", "or"),
  fill.missing = TRUE,
  summarize = TRUE,
  verbose = TRUE
)
```

Arguments

`object`	`RadioSet` The RadioSet to summarize
`mDataType`	`character` which one of the molecular data types to use in the analysis, out of all the molecular data types available for the rSet for example: rna, rnaseq, snp
`cell.lines`	`character` The cell lines to be summarized. If any cell.line has no data, missing values will be created
`features`	`caracter` A vector of the feature names to include in the summary
`summary.stat`	`character` which summary method to use if there are repeated cell.lines? Choices are "mean", "median", "first", or "last" In case molecular data type is mutation or fusion "and" and "or" choices are available

fill.missing	boolean should the missing cell lines not in the molecular data object be filled in with missing values?
summarize	A flag which when set to FALSE (defaults to TRUE) disables summarizing and returns the data unchanged as a ExpressionSet
verbose	boolean should messages be printed

Value

matrix An updated RadioSet with the molecular data summarized per cell line.

Examples

```
data(clevelandSmall)
clevelandSmall <- summarizeMolecularProfiles(clevelandSmall,
                    mDataType = "rna", cell.lines=sampleNames(clevelandSmall),
                    summary.stat = 'median', fill.missing = TRUE, verbose=TRUE)
clevelandSmall
```

---

summarizeSensitivityProfiles,RadioSet-method

Takes the sensitivity data from a RadioSet, and summarises them into a drug vs cell line table

---

Description

This function creates a table with cell lines as rows and radiation types as columns, summarising the drug senstitivity data of a RadioSet into drug-cell line pairs

Usage

```
## S4 method for signature 'RadioSet'
summarizeSensitivityProfiles(
  object,
  sensitivity.measure = "AUC_recomputed",
  cell.lines,
  radiation.types,
  summary.stat = c("mean", "median", "first", "last", "max", "min"),
  fill.missing = TRUE,
  verbose = TRUE
)
```

Arguments

object	'RadioSet' The RadioSet from which to extract the data
sensitivity.measure	'character' which sensitivity sensitivity.measure to use? Use the sensitivityMeasures function to find out what measures are available for each PSet.

cell.lines	‘character‘ The cell lines to be summarized. If any cell lines has no data, it will be filled with missing values
radiation.types	‘character‘ The radiation types to be summarized. If any radiation type has no data, it will be filled with missing values
summary.stat	‘character‘ which summary method to use if there are repeated cell line-drug experiments? Choices are "mean", "median", "first", or "last"
fill.missing	‘logical(1)‘ should the missing cell lines not in the molecular data object be filled in with missing values?
verbose	Should the function print progress messages?

Value

[matrix] A matrix with cell lines going down the rows, radiation types across the columns, with the selected sensitivity statistic for each pair.

Examples

```
data(clevelandSmall)
GDSCauc <- summarizeSensitivityProfiles(clevelandSmall, sensitivity.measure='AUC_published')
```

---

updateObject,RadioSet-method

Update the RadioSet class after changes in it struture or API

Description

Update the RadioSet class after changes in it struture or API

Usage

```
## S4 method for signature 'RadioSet'
updateObject(object)
```

Arguments

object	A RadioSet object to update the class structure for.

Value

RadioSet with update class structure.

`[,RadioSet,ANY,ANY,ANY-method`
'['

Description

'['

Usage

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

Arguments

x	object
i	Cell lines to keep in object
j	Drugs to keep in object
...	further arguments
drop	A boolean flag of whether to drop single dimensions or not

Value

Returns the subsetted object

Examples

```
data(clevelandSmall)
clevelandSmall[sampleNames(clevelandSmall)[1], treatmentNames(clevelandSmall)[1]]
```

Index

∗ **datasets**
    clevelandSmall, 4
.RadioSet *(*RadioSet-class*)*, 23
[,RadioSet,ANY,ANY,ANY-method, 33

annotation *(*RadioSet-accessors*)*, 14
annotation,RadioSet-method
        *(*RadioSet-accessors*)*, 14
annotation<- *(*RadioSet-accessors*)*, 14
annotation<-,RadioSet,list-method
        *(*RadioSet-accessors*)*, 14
availableRSets, 3

cellInfo *(*RadioSet-accessors*)*, 14
cellInfo,RadioSet-method
        *(*RadioSet-accessors*)*, 14
cellInfo<- *(*RadioSet-accessors*)*, 14
cellInfo<-,RadioSet,data.frame-method
        *(*RadioSet-accessors*)*, 14
cellName,RadioSet-method
        *(*RadioSet-accessors*)*, 14
cellNames *(*RadioSet-accessors*)*, 14
cellNames<- *(*RadioSet-accessors*)*, 14
cellNames<-,RadioSet,list-method
        *(*RadioSet-accessors*)*, 14
checkRSetStructure, 4
clevelandSmall, 4
computeAUC, 5
computeD10, 6
computeSF2, 7
curation *(*RadioSet-accessors*)*, 14
curation,RadioSet-method
        *(*RadioSet-accessors*)*, 14
curation<- *(*RadioSet-accessors*)*, 14
curation<-,RadioSet,list-method
        *(*RadioSet-accessors*)*, 14

datasetType *(*RadioSet-accessors*)*, 14
datasetType,RadioSet-method
        *(*RadioSet-accessors*)*, 14
datasetType<- *(*RadioSet-accessors*)*, 14
datasetType<-,RadioSet,character-method
        *(*RadioSet-accessors*)*, 14
dateCreated *(*RadioSet-accessors*)*, 14
dateCreated,RadioSet-method
        *(*RadioSet-accessors*)*, 14
dateCreated<- *(*RadioSet-accessors*)*, 14
dateCreated<-,RadioSet,character-method
        *(*RadioSet-accessors*)*, 14
dateCreated<-,RadioSet-method
        *(*RadioSet-accessors*)*, 14
dim,RadioSet-method, 7
doseResponseCurve, 8
downloadRSet, 10

featureInfo *(*RadioSet-accessors*)*, 14
featureInfo,RadioSet-method
        *(*RadioSet-accessors*)*, 14
featureInfo<- *(*RadioSet-accessors*)*, 14
featureInfo<-,RadioSet,character,data.frame-method
        *(*RadioSet-accessors*)*, 14
featureInfo<-,RadioSet,character,DataFrame-method
        *(*RadioSet-accessors*)*, 14
fNames *(*RadioSet-accessors*)*, 14
fNames,RadioSet,character-method
        *(*RadioSet-accessors*)*, 14
fNames<- *(*RadioSet-accessors*)*, 14
fNames<-,RadioSet,character,character-method
        *(*RadioSet-accessors*)*, 14

linearQuadraticModel, 11

mDataNames *(*RadioSet-accessors*)*, 14
mDataNames,RadioSet-method
        *(*RadioSet-accessors*)*, 14
mDataNames<- *(*RadioSet-accessors*)*, 14
mDataNames<-,RadioSet,ANY-method
        *(*RadioSet-accessors*)*, 14
mDataNames<-,RadioSet-method
        *(*RadioSet-accessors*)*, 14

molecularProfiles (RadioSet-accessors), 14
molecularProfiles,RadioSet-method (RadioSet-accessors), 14
molecularProfiles<- (RadioSet-accessors), 14
molecularProfiles<-,RadioSet,character,character,matrix-method (RadioSet-accessors), 14
molecularProfiles<-,RadioSet,character,missing,matrix-method (RadioSet-accessors), 14
molecularProfiles<-,RadioSet,missing,missing,list-method (RadioSet-accessors), 14
molecularProfiles<-,RadioSet,missing,missing,MultiAssayExperiment-method (RadioSet-accessors), 14
molecularProfilesSlot (RadioSet-accessors), 14
molecularProfilesSlot,RadioSet-method (RadioSet-accessors), 14
molecularProfilesSlot<- (RadioSet-accessors), 14
molecularProfilesSlot<-,RadioSet,list-method (RadioSet-accessors), 14
molecularProfilesSlot<-,RadioSet,list_OR_MAE-method (RadioSet-accessors), 14
molecularProfilesSlot<-RadioSet,MultiAssayExperiment-method (RadioSet-accessors), 14
moleculerProfilesSlot,RadioSet-method (RadioSet-accessors), 14

name (RadioSet-accessors), 14
name,RadioSet-method (RadioSet-accessors), 14
name<- (RadioSet-accessors), 14
name<-,RadioSet,character-method (RadioSet-accessors), 14
name<-,RadioSet-method (RadioSet-accessors), 14

pertNumber (RadioSet-accessors), 14
pertNumber,RadioSet-method (RadioSet-accessors), 14
pertNumber<- (RadioSet-accessors), 14
pertNumber<-,RadioSet,array-method (RadioSet-accessors), 14
phenoInfo (RadioSet-accessors), 14
phenoInfo,RadioSet,character-method (RadioSet-accessors), 14
phenoInfo<- (RadioSet-accessors), 14
phenoInfo<-,RadioSet,character,data.frame-method (RadioSet-accessors), 14
phenoInfo<-,RadioSet,character,DataFrame-method (RadioSet-accessors), 14
plotCurve, 12

radiationInfo (RadioSet-accessors), 14
radiationInfo<- (RadioSet-accessors), 14
radiationTypes (RadioSet-accessors), 14
radiationTypes<- (RadioSet-accessors), 14
RadioSet, 13
RadioSet-accessors, 14
RadioSet-class, 23
RadioSig, 24
radSensitivitySig, 25

sampleInfo (RadioSet-accessors), 14
sampleInfo,RadioSet-method (RadioSet-accessors), 14
sampleInfo<- (RadioSet-accessors), 14
sampleInfo<-,RadioSet,data.frame-method (RadioSet-accessors), 14
sampleName,RadioSet-method (RadioSet-accessors), 14
sampleNames (RadioSet-accessors), 14
sampleNames,RadioSet-method (RadioSet-accessors), 14
sampleNames<- (RadioSet-accessors), 14
sampleNames<-,RadioSet,character-method (RadioSet-accessors), 14
sampleNames<-,RadioSet,list-method (RadioSet-accessors), 14
sensitivityInfo,RadioSet,character-method (RadioSet-accessors), 14
sensitivityInfo,RadioSet,missing-method (RadioSet-accessors), 14
sensitivityInfo,RadioSet-method (RadioSet-accessors), 14
sensitivityInfo<-,RadioSet,data.frame-method (RadioSet-accessors), 14
sensitivityInfo<-,RadioSet,missing,data.frame-method (RadioSet-accessors), 14
sensitivityMeasures,RadioSet-method (RadioSet-accessors), 14
sensitivityMeasures<-,RadioSet,character-method (RadioSet-accessors), 14
sensitivityProfiles,RadioSet-method (RadioSet-accessors), 14

sensitivityProfiles<-,RadioSet,data.frame-method
(RadioSet-accessors), 14
sensitivityRaw,RadioSet-method
(RadioSet-accessors), 14
sensitivityRaw<-,RadioSet,array-method
(RadioSet-accessors), 14
sensitivitySlot (RadioSet-accessors), 14
sensitivitySlot<- (RadioSet-accessors),
14
sensitvityInfo<-,RadioSet,character,data.frame-method
(RadioSet-accessors), 14
sensNumber (RadioSet-accessors), 14
sensNumber,RadioSet-method
(RadioSet-accessors), 14
sensNumber<- (RadioSet-accessors), 14
sensNumber<-,RadioSet,matrix-method
(RadioSet-accessors), 14
show,RadioSet-method, 27
show,RadioSig-method, 27
showSigAnnot,RadioSig-method, 28
subsetTo,RadioSet-method, 29
summarizeMolecularProfiles,RadioSet-method,
30
summarizeSensitivityProfiles,RadioSet-method,
31

treamentResponse<-,RadioSet,list-method
(RadioSet-accessors), 14
treatmentInfo (RadioSet-accessors), 14
treatmentInfo,RadioSet-method
(RadioSet-accessors), 14
treatmentInfo<- (RadioSet-accessors), 14
treatmentInfo<-,RadioSet,data.frame-method
(RadioSet-accessors), 14
treatmentNames (RadioSet-accessors), 14
treatmentNames,RadioSet-method
(RadioSet-accessors), 14
treatmentNames<- (RadioSet-accessors),
14
treatmentNames<-,RadioSet,character-method
(RadioSet-accessors), 14
treatmentResponse (RadioSet-accessors),
14
treatmentResponse,RadioSet-method
(RadioSet-accessors), 14
treatmentResponse<-
(RadioSet-accessors), 14
treatmentResponse<-,RadioSet,list_OR_LongTable-method
(RadioSet-accessors), 14
treatmentResponse<-,RadioSet,LongTable-method
(RadioSet-accessors), 14

updateObject,RadioSet-method, 32