

# Package ‘RadioGx’

June 6, 2019

**Type** Package

**Title** Analysis of Large-Scale Radiogenomic Data

**Version** 0.0.0.1

**Date** 2019-05-31

**Author** BHKLab

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

**Description** Contains a set of functions to perform large-scale analysis of radiogenomic data.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Depends** R (*i*= 3.5.0)

**Imports** PharmacoGx, CoreGx, Biobase, RColorBrewer, caTools, magicaxis,  
methods, reshape2, KernSmooth, cluster, Matrix, scales

**RoxygenNote** 6.1.1

## R topics documented:

.linearQuadratic . . . . .	2
checkRSetStructure . . . . .	2
Cleveland_small . . . . .	3
computeAUC . . . . .	3
computeD10 . . . . .	4
computeSF2 . . . . .	5
dim,RadioSet-method . . . . .	5
doseResponseCurve . . . . .	6
linearQuadraticModel . . . . .	7
mDataNames,RadioSet-method . . . . .	8
plotCurve . . . . .	9
radiationInfo . . . . .	9
radiationInfoj- . . . . .	10
radiationTypes . . . . .	10
radiationTypesj- . . . . .	11
RadioSet . . . . .	11
RadioSet-class . . . . .	12
radSensitivitySig . . . . .	18
show,RadioSet-method . . . . .	20

show, RadioSig-method . . . . .	20
showSigAnnot . . . . .	21
subsetTo . . . . .	21
summarizeMolecularProfiles . . . . .	22
summarizeSensitivityProfiles . . . . .	23
[, RadioSet, ANY, ANY, ANY-method . . . . .	24

<b>Index</b>	<b>26</b>
--------------	-----------

---

<code>.linearQuadratic</code>	<i>Linear quadratic</i>
-------------------------------	-------------------------

---

## Description

Linear quadratic

## Usage

```
.linearQuadratic(D, pars, SF_as_log = TRUE)
```

## Arguments

<code>D</code>	A vector of drug concentrations
<code>pars</code>	Parameters (a, b) of the linear model
<code>SF_as_log</code>	Boolean indicating whether survival fraction is logged

---

<code>checkRSetStructure</code>	<i>A function to verify the structure of a RadioSet</i>
---------------------------------	---

---

## 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(rSet, plotDist = FALSE, result.dir = ".")
```

## Arguments

<code>rSet</code>	A RadioSet object
<code>plotDist</code>	Should the function also plot the distribution of molecular data?
<code>result.dir</code>	The path to the directory for saving the plots as a string

## Value

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

**Examples**

```
data(Cleveland_small)

checkRSetStructure(Cleveland_small)
```

---

Cleveland_small	<i>Cleveland_mut RadioSet subsetting</i>
-----------------	--

---

**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 example usage in the roxygen2 documentation for CoreGx.

**Usage**

```
data(Cleveland_small)
```

**Format**

RadioSet object

**References**

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

---

computeAUC	<i>computeAUC: computes AUC</i>
------------	---------------------------------

---

**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 = TRUE,
  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 TRUE
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.

## Examples

```
computeAUC(D=c(0.1, 0.5, 0.7, 0.9), pars=c(0.2, 0.1), lower = 0, upper = 4) # Returns 0.7039296
```

---

computeD10

*Compute D10*

---

## Description

This function computes the radiation dose at which only 10

## 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.

## 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  $y = \exp(-\alpha * x - \beta * x^2)$

### 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

---

doseResponseCurve	<i>Plot drug response curve of a given drug and a given cell for a list of rSets (objects of the RadioSet class).</i>
-------------------	---

---

## 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 measurement 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	[string] 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	[string] 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	[bool] Should the viability values be truncated to lie in [0-100] before doing the fitting
legends.label	[vector] A vector of sensitivity measurement 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 measurements for one of the requested rSets. If this parameter is missed no legend would be provided for the plot.
ylim	[vector] 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	[vector] 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.

<code>mycol</code>	[vector] A vector with the same length of the <code>rSets</code> parameter which will determine the color of the curve for the pharmacokinetic sets. If this parameter is missed default colors from Rcolorbrewer package will be used as curves color.
<code>title</code>	[character] The title of the graph. If no title is provided, then it defaults to 'Drug': 'Cell Line'.
<code>plot.type</code>	[character] Plot type which can be the actual one ("Actual") or the one fitted by logit logistic regression ("Fitted") or both of them ("Both"). If this parameter is missed by default actual curve is plotted.
<code>summarize.replicates</code>	[character] If this parameter is set to true replicates are summarized and replicates are plotted individually otherwise
<code>lwd</code>	[numeric] The line width to plot with
<code>cex</code>	[numeric] The cex parameter passed to plot
<code>cex.main</code>	[numeric] The cex.main parameter passed to plot, controls the size of the titles
<code>legend.loc</code>	And argument passable to <code>xy.coords</code> for the position to place the legend.
<code>verbose</code>	[boolean] Should warning messages about the data passed in be printed?

## Value

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

## Examples

```
if (interactive()) {
  drugDoseResponseCurve(Ds=list("Experiment 1" = c(0, 2, 4, 6)),
    SFs=list("Experiment 1" = c(1,.6,.4,.2)), plot.type="Both")
}
```

---

`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 details
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	see details

**Details**

'verbose' outputs warnings that are otherwise suppressed when the function sanity-checks user inputs. 'median\_n' denotes the number of distributions from family 'family' that are medians. (Note that setting n = 1 (the default) is equivalent to using a simple normal or cauchy distribution without taking any medians.)

**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))
```

---

mDataNames, RadioSet-method

*mDataNames*


---

**Description**

Returns the molecular data names for the RadioSet.

**Usage**

```
## S4 method for signature 'RadioSet'
mDataNames(cSet = rSet)
```

**Arguments**

cSet	The parameter
------	---------------

**Value**

Vector of names of the molecular data types

**Examples**

```
data(Cleveland_small)
mDataNames(Cleveland_small)
```



---

plotCurve	<i>Plot radiation dose-response curve</i>
-----------	---

---

### 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

### 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))
```

---

radiationInfo	<i>radiationInfo Generic</i>
---------------	------------------------------

---

### Description

Generic for radiationInfo method

### Usage

```
radiationInfo(rSet)
```

### Arguments

rSet	A RadioSet object
------	-------------------

### Value

a data.frame with the radiation annotations

**Examples**

```
data(Cleveland_small)
radiationInfo(Cleveland_small)
```

---

<code>radiationInfo&lt;-</code>	<i>radiationInfoj- Generic</i>
---------------------------------	--------------------------------

---

**Description**

Generic for radiationInfo replace method

**Usage**

```
radiationInfo(object) <- value
```

**Arguments**

<code>object</code>	The RadioSet to replace radiation info in
<code>value</code>	A <code>data.frame</code> with the new radiation annotations

**Value**

Updated RadioSet

**Examples**

```
data(Cleveland_small)
radiationInfo(Cleveland_small) <- radiationInfo(Cleveland_small)
```

---

<code>radiationTypes</code>	<i>radiationTypes Generic</i>
-----------------------------	-------------------------------

---

**Description**

A generic for the radiationTypes method

**Usage**

```
radiationTypes(rSet)
```

**Arguments**

<code>rSet</code>	A RadioSet
-------------------	------------

**Value**

A vector of the radiation names used in the RadioSet

**Examples**

```
data(Cleveland_small)
radType <- radiationTypes(Cleveland_small)
radType[1:10]
```

---

radiationTypes<-	<i>radiationTypesj- Generic</i>
------------------	---------------------------------

---

**Description**

A generic for the radiationTypes replacement method

**Usage**

```
radiationTypes(object) <- value
```

**Arguments**

object	A RadioSet object to update
value	A character vector of the new radiation names

**Value**

Updated RadioSet

**Examples**

```
data(Cleveland_small)
radiationTypes(Cleveland_small) <- radiationTypes(Cleveland_small)
```

---

RadioSet	<i>RadioSet constructor</i>
----------	-----------------------------

---

**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(), cell = data.frame(),
  radiation = 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)),
  curationCell = data.frame(), curationTissue = data.frame(),
  datasetType = c("sensitivity", "perturbation", "both"),
  verify = TRUE)
```

**Arguments**

<code>name</code>	A character string detailing the name of the dataset
<code>molecularProfiles</code>	A list of ExpressionSet objects containing molecular profiles
<code>cell</code>	A data.frame containing the annotations for all the cell lines profiled in the data set, across all data types
<code>radiation</code>	A data.frame containing the annotations for all the radiations profiled in the data set, across all data types
<code>sensitivityInfo</code>	A data.frame containing the information for the sensitivity experiments
<code>sensitivityRaw</code>	A 3 Dimensional array containing the raw radiation dose – response data for the sensitivity experiments
<code>sensitivityProfiles</code>	data.frame containing radiation sensitivity profile statistics such as IC50 and AUC
<code>sensitivityN, perturbationN</code>	A data.frame summarizing the available sensitivity/perturbation data
<code>curationCell, curationTissue</code>	A data.frame mapping the names for radiations, cells and tissues used in the data set to universal identifiers used between different RadioSet objects
<code>datasetType</code>	A character string of 'sensitivity', 'preturbation', or both detailing what type of data can be found in the RadioSet, for proper processing of the data
<code>verify</code>	boolean Should the function verify the RadioSet and print out any errors it finds after construction?

**Value**

An object of class RadioSet

**Examples**

```
## For help creating a RadioSet object, please see the following vignette:
browseVignettes("PharmacGx")
```

---

RadioSet-class	<i>A Class to Contain RadioGenomic datasets together with their curations</i>
----------------	---

---

**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 accomodate datasets containing both radiation dose

response data, as well as datasets containing genetic profiles of cell lines pre and post treatment with compounds, known respectively as sensitivity and perturbation datasets.

Generic for phenoInfo method

Generic for phenoInfo replace method

Generic for molecularProfiles method

Generic for molecularProfiles replace method

Generic for featureInfo replace method

Generic for sensitivityInfo method

A generic for the sensitivityInfo replacement method

Generic for sensitivityProfiles method

A generic for the sensitivityProfiles replacement method

A generic for the sensitivityMeasures method

A generic for the cellNames method

A generic for the cellNames replacement method

A generic for the fName method

A generic for the dateCreated method

A generic for the rSetName method

A generic for the pertNumber method

A generic for the sensNumber method

A generic for the pertNumber method

A generic for the sensNumber method

## Usage

```
## S4 method for signature 'RadioSet'
cellInfo(cSet = rSet)

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

## S4 method for signature 'RadioSet'
radiationInfo(rSet)

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

## S4 method for signature 'RadioSet,character'
phenoInfo(cSet = rSet, mDataType)

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

## S4 method for signature 'RadioSet,character'
molecularProfiles(cSet = rSet, mDataType)

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

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

## S4 method for signature 'RadioSet,character'
featureInfo(cSet = rSet, mDataType)

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

## S4 method for signature 'RadioSet'
sensitivityInfo(cSet = rSet)

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

## S4 method for signature 'RadioSet'
sensitivityProfiles(cSet = rSet)

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

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

## S4 method for signature 'RadioSet'
sensitivityMeasures(cSet = rSet)

## S4 method for signature 'RadioSet'
radiationTypes(rSet)

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

## S4 method for signature 'RadioSet'
cellNames(cSet = rSet)

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

## S4 method for signature 'RadioSet,character'
fNames(cSet = rSet, mDataType)

## S4 method for signature 'RadioSet'
dateCreated(cSet = rSet)

## S4 method for signature 'RadioSet'
cSetName(cSet = rSet)

## S4 method for signature 'RadioSet'
pertNumber(cSet = rSet)
```

```
## S4 method for signature 'RadioSet'
sensNumber(cSet = rSet)

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

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

### Arguments

<code>cSet</code>	Parameter name for parent method inherited from CoreGx
<code>object</code>	A RadioSet object
<code>value</code>	A replacement value
<code>rSet</code>	A RadioSet object
<code>mDataType</code>	A character with the type of molecular data to return/update

### Value

An object of the RadioSet class

a `data.frame` with the cell annotations

Updated RadioSet

a `data.frame` with the experiment info

The updated RadioSet

a `data.frame` with the experiment info

Updated RadioSet

Updated RadioSet

a `data.frame` with the experiment info

Updated RadioSet

a `data.frame` with the experiment info

Updated RadioSet

A character vector of all the available sensitivity measures

A vector of the cell names used in the RadioSet

Updated RadioSet

A character vector of the feature names

The date the RadioSet was created

The name of the RadioSet

A 3D array with the number of perturbation experiments per radiation type and cell line, and data type

A `data.frame` with the number of sensitivity experiments per radiation type and cell line

The updated RadioSet

The updated RadioSet

**Methods (by generic)**

- `cellInfo`: Returns the annotations for all the cell lines tested on in the `RadioSet`
- `cellInfo<-`: Update the cell line annotations
- `radiationInfo`: Returns the annotations for all the radiations tested in the `RadioSet`
- `radiationInfo<-`: Update the radiation annotations
- `phenoInfo`: Return the experiment info from the given type of molecular data in `RadioSet`
- `phenoInfo<-`: Update the the given type of molecular data experiment info in the `RadioSet`
- `molecularProfiles`: Return the given type of molecular data from the `RadioSet`
- `molecularProfiles<-`: Update the given type of molecular data from the `RadioSet`
- `featureInfo`: Return the feature info for the given molecular data
- `featureInfo<-`: Replace the gene info for the molecular data
- `sensitivityInfo`: Return the radiation dose sensitivity experiment info
- `sensitivityInfo<-`: Update the sensitivity experiment info
- `sensitivityProfiles`: Return the phenotypic data for the radiation dose sensitivity
- `sensitivityProfiles<-`: Update the phenotypic data for the radiation dose sensitivity
- `sensitivityProfiles<-`: Update the phenotypic data for the radiation dose sensitivity
- `sensitivityMeasures`: Returns the available sensitivity profile summaries, for example, whether there are IC50 values available
- `radiationTypes`: Return the names of the radiations used in the `RadioSet`
- `radiationTypes<-`: Update the radiation names used in the dataset
- `cellNames`: Return the cell names used in the dataset
- `cellNames<-`: Update the cell names used in the dataset
- `fNames`: Return the feature names used in the dataset
- `dateCreated`: Return the date the `RadioSet` was created
- `cSetName`: Return the name of the `RadioSet`
- `pertNumber`: Return the summary of available perturbation experiments
- `sensNumber`: Return the summary of available sensitivity experiments
- `pertNumber<-`: Update the summary of available perturbation experiments
- `sensNumber<-`: Update the summary of available sensitivity experiments

**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 `Biobase::ExpressionSet` 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
- `cell` A `data.frame` containing the annotations for all the cell lines profiled in the data set, across all data types
- `radiation` A `data.frame` containing 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 containing `$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

## Examples

```
data(Cleveland_small)
phenoInfo(Cleveland_small, mDataType="rna")

data(Cleveland_small)
phenoInfo(Cleveland_small, mDataType="rna") <- phenoInfo(Cleveland_small, mDataType="rna")

data(Cleveland_small)
Cleveland_mProf <- molecularProfiles(Cleveland_small, "rna")
Cleveland_mProf[1:10,]

data(Cleveland_small)
molecularProfiles(Cleveland_small, "rna") <- molecularProfiles(Cleveland_small, "rna")

data(Cleveland_small)
featureInfo(Cleveland_small, "rna")[1:10,]

data(Cleveland_small)
featureInfo(Cleveland_small, "rna") <- featureInfo(Cleveland_small, "rna")

data(Cleveland_small)
sensInf<- sensitivityInfo(Cleveland_small)
sensInf[1:10,]

data(Cleveland_small)
sensitivityInfo(Cleveland_small) <- sensitivityInfo(Cleveland_small)

data(Cleveland_small)
sensitivityProfiles(Cleveland_small)

data(Cleveland_small)
sensitivityProfiles(Cleveland_small) <- sensitivityProfiles(Cleveland_small)

data(Cleveland_small)
sensitivityMeasures(Cleveland_small)

data(Cleveland_small)
cellNames(Cleveland_small)

data(Cleveland_small)
cellNames(Cleveland_small) <- cellNames(Cleveland_small)
```

```

data(Cleveland_small)
fNames(Cleveland_small, "rna")[1:10]

data(Cleveland_small)
dateCreated(Cleveland_small)

data(Cleveland_small)
rSetName <- cSetName
rSetName(Cleveland_small)

data(Cleveland_small)
pertNumber(Cleveland_small)

data(Cleveland_small)
sensNumber(Cleveland_small)

data(Cleveland_small)
pertNumber(Cleveland_small) <- pertNumber(Cleveland_small)

data(Cleveland_small)
sensNumber(Cleveland_small) <- sensNumber(Cleveland_small)

```

---

<b>radSensitivitySig</b>	<i>Creates a signature representing the association between gene expression (or other molecular profile) and radiation dose response, for use in radiation sensitivity analysis.</i>
--------------------------	--

---

## Description

Given a RadioSet of the sensitivity experiment type, and a list of drugs, the function will compute a signature for the effect 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, standardize = c("SD", "rescale", "none"),
  nthread = 1, verbose = TRUE, ...)

```

## Arguments

<b>rSet</b>	[PharmacoSet] a PharmacoSet of the perturbation experiment type
<b>mDataType</b>	[character] which one of the molecular data types to use in the analysis, out of dna, rna, rnaseq, snp, cnv

<code>radiation.types</code>	[character] a vector of radiation.types for which to compute the signatures. Should match the names used in the PharmacoSets.
<code>features</code>	[character] a vector of features for which to compute the signatures. Should match the names used in correspondant molecular data in PharmacoSets.
<code>sensitivity.measure</code>	[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.
<code>molecular.summary.stat</code>	What summary statistic should be used to summarize duplicates for cell line molecular profile measurements?
<code>sensitivity.summary.stat</code>	What summary statistic should be used to summarize duplicates for cell line sensitivity measurements?
<code>returnValues</code>	[character] Which of estimate, t-stat, p-value and fdr should the function return for each gene?
<code>sensitivity.cutoff</code>	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.
<code>standardize</code>	[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.
<code>nthread</code>	[numeric] if multiple cores are available, how many cores should the computation be parallelized over?
<code>verbose</code>	[boolean] 'TRUE' if the warnings and other infomrative message should be displayed
<code>...</code>	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(Cleveland_small)
rad.sensitivity <- radSensitivitySig(Cleveland_small, mDataType="rna",
                                   nthread=1, features = fNames(Cleveland_small, "rna")[1],
                                   radiation.types=radiationTypes(Cleveland_small))
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(Cleveland_small)
Cleveland_small
```

---

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

---

showSigAnnot	<i>Show the Annotations of a signature object</i>
--------------	---

---

## Description

This function 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

```
showSigAnnot(Sigs)
```

## Arguments

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

---

subsetTo	<i>A function to subset a RadioSet to data containing only specified radiations, cells and genes</i>
----------	--

---

## Description

This is the preferred 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

```
subsetTo(rSet, cells = NULL, radiationTypes = NULL,
         molecular.data.cells = NULL, keep.controls = TRUE, ...)
```

**Arguments**

<code>rSet</code>	A <code>RadioSet</code> to be subsetted
<code>cells</code>	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.
<code>radiationTypes</code>	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.
<code>molecular.data.cells</code>	A list or vector of cell names to keep in the molecular data
<code>keep.controls</code>	If the dataset has perturbation type experiments, should the controls be kept in the dataset? Defaults to true.
<code>...</code>	Other arguments passed by other function within the package

**Value**

A `RadioSet` with only the selected radiation types and cells

**Examples**

```
data(Cleveland_small)
clevelandRadiationTypes <- radiationTypes(Cleveland_small)
clevelandCells <- cellNames(Cleveland_small)
RSet <- subsetTo(Cleveland_small, radiationTypes = clevelandRadiationTypes[1],
  cells = clevelandCells[1])
RSet
```

---

**summarizeMolecularProfiles**

*Takes molecular data from a `PharmacoSet`, and summarises them into one entry per drug*

---

**Description**

Given a `PharmacoSet` with molecular data, this function will summarize the data into one profile per cell line, using the chosen `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**

```
summarizeMolecularProfiles(rSet, mDataType, cell.lines, features,
  summary.stat = c("mean", "median", "first", "last", "and", "or"),
  fill.missing = TRUE, summarize = TRUE, verbose = TRUE)
```



**Arguments**

<code>rSet</code>	[RadioSet] The RadioSet from which to extract the data
<code>sensitivity.measure</code>	[character] which sensitivity sensitivity.measure to use? Use the <code>sensitivityMeasures</code> function to find out what measures are available for each PSet.
<code>cell.lines</code>	character The cell lines to be summarized. If any cell lines has no data, it will be filled with missing values
<code>radiation.types</code>	character The radiation types to be summarized. If any radiation type has no data, it will be filled with missing values
<code>summary.stat</code>	character which summary method to use if there are repeated cell line-drug experiments? Choices are "mean", "median", "first", or "last"
<code>fill.missing</code>	boolean should the missing cell lines not in the molecular data object be filled in with missing values?
<code>verbose</code>	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(Cleveland_small)
GDSCauc <- summarizeSensitivityProfiles(Cleveland_small, sensitivity.measure='AUC_published')
```

---

```
[,RadioSet,ANY,ANY,ANY-method
  ‘[‘
```

---

**Description**

```
‘[‘
```

**Usage**

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

**Arguments**

<code>x</code>	RSet
<code>i</code>	Cell lines to keep in RSet
<code>j</code>	Drugs to keep in RSet
<code>...</code>	further arguments
<code>drop</code>	A boolean flag of whether to drop single dimensions or not



*[,RadioSet,ANY,ANY,ANY-method*

25

**Value**

Returns the subsetted RSet

# Index

## \*Topic **datasets**

- Cleveland\_small, [3](#)
- .RadioSet (RadioSet-class), [12](#)
- .linearQuadratic, [2](#)
- [,RadioSet,ANY,ANY,ANY-method, [24](#)
- cellInfo, RadioSet-method  
(RadioSet-class), [12](#)
- cellInfo<-, RadioSet, data.frame-method  
(RadioSet-class), [12](#)
- cellNames, RadioSet-method  
(RadioSet-class), [12](#)
- cellNames<-, RadioSet, character-method  
(RadioSet-class), [12](#)
- checkRSetStructure, [2](#)
- Cleveland\_small, [3](#)
- computeAUC, [3](#)
- computeD10, [4](#)
- computeSF2, [5](#)
- cSetName, RadioSet-method  
(RadioSet-class), [12](#)
- dateCreated, RadioSet-method  
(RadioSet-class), [12](#)
- dim, RadioSet-method, [5](#)
- doseResponseCurve, [6](#)
- featureInfo, RadioSet, character-method  
(RadioSet-class), [12](#)
- featureInfo<-, RadioSet, character, data.frame-method  
(RadioSet-class), [12](#)
- fNames, RadioSet, character-method  
(RadioSet-class), [12](#)
- linearQuadraticModel, [7](#)
- mDataNames, RadioSet-method, [8](#)
- molecularProfiles, RadioSet, character-method  
(RadioSet-class), [12](#)
- molecularProfiles<-, RadioSet, character, matrix-method  
(RadioSet-class), [12](#)
- pertNumber, RadioSet-method  
(RadioSet-class), [12](#)
- pertNumber<-, RadioSet, array-method  
(RadioSet-class), [12](#)
- phenoInfo, RadioSet, character-method  
(RadioSet-class), [12](#)
- phenoInfo<-, RadioSet, character, data.frame-method  
(RadioSet-class), [12](#)
- plotCurve, [9](#)
- radiationInfo, [9](#)
- radiationInfo, RadioSet-method  
(RadioSet-class), [12](#)
- radiationInfo<-, [10](#)
- radiationInfo<-, RadioSet, data.frame-method  
(RadioSet-class), [12](#)
- radiationTypes, [10](#)
- radiationTypes, RadioSet-method  
(RadioSet-class), [12](#)
- radiationTypes<-, [11](#)
- radiationTypes<-, RadioSet, character-method  
(RadioSet-class), [12](#)
- RadioSet, [11](#)
- RadioSet-class, [12](#)
- radSensitivitySig, [18](#)
- sensitivityInfo, RadioSet-method  
(RadioSet-class), [12](#)
- sensitivityInfo<-, RadioSet, data.frame-method  
(RadioSet-class), [12](#)
- sensitivityMeasures, RadioSet-method  
(RadioSet-class), [12](#)
- sensitivityProfiles, RadioSet-method  
(RadioSet-class), [12](#)
- sensitivityProfiles<-, RadioSet, data.frame-method  
(RadioSet-class), [12](#)
- sensitivityProfiles<-, RadioSet, matrix-method  
(RadioSet-class), [12](#)
- sensNumber, RadioSet-method  
(RadioSet-class), [12](#)
- sensNumber<-, RadioSet, matrix-method  
(RadioSet-class), [12](#)
- show, RadioSet-method, [20](#)
- show, RadioSig-method, [20](#)
- showSigAnnot, [21](#)
- subsetTo, [21](#)

summarizeMolecularProfiles, [22](#)  
summarizeSensitivityProfiles, [23](#)