Package 'RadioGx'

October 14, 2021

Type Package

Title Analysis of Large-Scale Radio-Genomic Data

Version 1.2.0

Date 2021-05-13

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, 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 7.1.1

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

git_branch RELEASE_3_13

git_last_commit fdb4a45

git_last_commit_date 2021-05-19

Date/Publication 2021-10-14

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

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

R topics documented:

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)
```

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.

Value

A rSet object with the dataset, downloaded from our server

Examples

```
if (interactive()) {
drugMatrix_rat <- 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))
```

mDataNames,RadioSet-method *mDataNames*

Description

Returns the molecular data names for the RadioSet.

Usage

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

Arguments

object	The parameter

Value

Vector of names of the molecular data types

Examples

```
mDataNames(clevelandSmall)
```

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)
```

---

radiationInfo *radiationInfo Generic*

Description

Generic for radiationInfo method

Usage

```
radiationInfo(object)
```

Arguments

object	A RadioSet object

Value

a data.frame with the radiation annotations

Examples

```
radiationInfo(clevelandSmall)
```

radiationInfo<- *radiationInfo<- Generic*

Description

Generic for radiationInfo replace method

Usage

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

Arguments

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

Value

Updated RadioSet

Examples

```
radiationInfo(clevelandSmall) <- radiationInfo(clevelandSmall)
```

radiationTypes *radiationTypes Generic*

Description

A generic for the radiationTypes method

Usage

```
radiationTypes(object)
```

Arguments

object	A RadioSet

Value

A vector of the radiation names used in the RadioSet

Examples

```
data(clevelandSmall)
radTypes <- radiationTypes(clevelandSmall)
```

`radiationTypes<-` *radiationTypes<- Generic*

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

```
radiationTypes(clevelandSmall) <- radiationTypes(clevelandSmall)
```

`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(),
  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

`name`	A `character` string detailing the name of the dataset
`molecularProfiles`	A `list` of ExpressionSet objects containing molecular profiles
`cell`	A `data.frame` containg the annotations for all the cell lines profiled in the data set, across all data types
`radiation`	A `data.frame` containg the annotations for all the radiations profiled in the data set, across all data types
`sensitivityInfo`	A `data.frame` containing the information for the sensitivity experiments
`sensitivityRaw`	A 3 Dimensional `array` contaning the raw radiation dose – response data for the sensitivity experiments
`sensitivityProfiles`	`data.frame` containing radiation sensitivity profile statistics such as IC50 and AUC
`sensitivityN, perturbationN`	A `data.frame` summarizing the available sensitivity/perturbation data
`curationCell, curationTissue`	A `data.frame` mapping the names for radiations, cells and tissues 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
`verify`	`boolean` Should the function verify the RadioSet and print out any errors it finds after construction?

Value

An object of class RadioSet

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.

Get cell line information from a RadioSet object

Set cell line annotations for a RadioSet object

Get the names of all cell lines in a RadioSet object

Set the names of all cell lines in a RadioSet object

A generic for retrieving the dataset type of an rSet object

A generic for updating the dataset type of a RadioSet object

Get the date a RadioSet object was created

Return the feature names for the specified molecular data type

Setter for the feature names of a `SummarizedExperiment` in the molecularProfiles slot

Getter method for the feature data of the given molecular data type

Setter method for the feature data of the given molecular data type

Getter method for the molecular profile data of the given molecular data type

Setter method for the molecular profile data of the given molecular data type

Get the name of a RadioSet

Get a summary of available perturbation experiments

Get the summary of the available sensitivity experiments in the RadioSet

Getter method for the phenotype information of a given molecular data type

Setter method for the phenotype information of a given molecular data type

Get a summary of the available sensitivity experiments in the RadioSet

Set the summary of the available sensitivity experiments in the RadioSet

Get the sensitivity information for the cell lines in the RadioSet

Set the sensitivity information for the cell lines in the RadioSet

Get the types of sensitivity measurements available in a RadioSet

Get the sensitivity values for the cell lines in the RadioSet

Set the sensitivity information for the cell lines in the RadioSet

Usage

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

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

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

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

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

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

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

## 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,character'
featureInfo(object, mDataType)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 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'
sensitivityRaw(object)

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

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

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

Arguments

object	A RadioSet to extract the raw sensitivity data from
value	A list of new sensitivity slot data for the rSet
mDataType	the type of molecular data
assay	character Name or index of the assay data to return

Value

An object of the RadioSet class

A list of named annotaiton

A copy of the RadioSet with the updated annotation slot

a data.frame with the cell annotations

Updated RadioSet

A vector of the cell line names in the RadioSet

Updated RadioSet

A list of unique cell and tissue identifiers to check validity of an rSet

A copy of the RadioSet with the updated curation slot

A character vector containing the dataset tpye

A character vector containing the dataset tpye

The date the RadioSet was created

A character vector of the feature names

Updated RadioSet

A DataFrame containing the feature information

Updated RadioSet

a data.frame with the experiment info

Updated RadioSet

A list containing the molecularProfiles from a cSet

A copy of the RadioSet with the molecularProfiles slot updated

The name of the RadioSet

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

The updated RadioSet

a data.frame with the experiment info

The updated RadioSet

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

The updated RadioSet

a DataFrame with the experiment info

Updated RadioSet

A character vector of all the available sensitivity measures

a data.frame with the experiment info

Updated RadioSet

A array containing the raw sensitivity data

A copy of the RadioSet containing the updated sensitivty data

A list of the sensitivity slot contents

A copy of the RadioSet containing the updated sensitivty slot

Methods (by generic)

- annotation: Retrieve the annotations slot form an rSet
- annotation<-: Update the annotation slot of a tSet
- cellInfo: Returns the annotations for all the cell lines tested on in the RadioSet
- cellInfo<-: Update the cell line annotations
- cellNames: Return the cell names used in the dataset
- cellNames<-: Update the cell names used in the dataset
- curation: Retrieve the curation slot form an rSet

- curation<-: Update the curations for cell and tissue types in an rSet object
- datasetType: Return the dataset type of an rSet object
- datasetType<-: Update the dataset type of an rSet and return a copy of the updated object
- dateCreated: Return the date the RadioSet was created
- fNames: Return the feature names used in the dataset
- fNames<-: Set the feature names for a given molecular data type
- featureInfo: Return the feature info for the given molecular data
- featureInfo<-: Replace the gene info for the molecular data
- molecularProfiles: Return the given type of molecular data from the RadioSet
- molecularProfiles<-: Update the given type of molecular data from the RadioSet
- molecularProfiles<-: Update the given type of molecular data from the RadioSet
- molecularProfilesSlot: Get contents of molecularProfiles slot
- molecularProfilesSlot<-: Update the molecular profiles slot of a RadioSet and returns the updated copy
- name: Return the name of the RadioSet
- pertNumber: Return the summary of available perturbation experiments
- pertNumber<-: Update the summary of available perturbation experiments
- 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
- radiationInfo: Returns the annotations for all the radiations tested in the RadioSet
- radiationInfo<-: Update the radiation annotations
- radiationTypes: Return the names of the radiations used in the RadioSet
- radiationTypes<-: Update the radiation names used in the dataset
- sensNumber: Return a summary of available sensitivity experiments
- sensNumber<-: Update the summary of available sensitivity experiments
- sensitivityInfo: Return the radiation dose sensitivity experiment info
- sensitivityInfo<-: Update the sensitivity experiment info
- sensitivityMeasures: Returns the available sensitivity profile summaries, for example, whether there are IC50 values available
- 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
- sensitivityRaw: Retrive the raw dose and viability data from an rSet
- sensitivityRaw<-: Update the raw dose and viability data in an rSet object
- sensitivitySlot: Retrieve the contents of the sensitivity slot
- sensitivitySlot<-: Set the raw dose and viability data for an rSet and return and updated copty

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

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

`radiation` 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

Examples

```
data(clevelandSmall)
annotation(clevelandSmall)

data(clevelandSmall)
annotation(clevelandSmall) <- annotation(clevelandSmall)

data(clevelandSmall)
cellInf <- cellInfo(clevelandSmall)

data(clevelandSmall)
cellInfo(clevelandSmall) <- cellInfo(clevelandSmall)

data(clevelandSmall)
cellNames(clevelandSmall)

data(clevelandSmall)
cellNames(clevelandSmall) <- cellNames(clevelandSmall)

data(clevelandSmall)
curation(clevelandSmall)

data(clevelandSmall)
curation(clevelandSmall) <- curation(clevelandSmall)

data(clevelandSmall)
```

```
datasetType(clevelandSmall)

data(clevelandSmall)
datasetType(clevelandSmall)

dateCreated(clevelandSmall)

data(clevelandSmall)
fNames(clevelandSmall, "rna")

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

featInf <- featureInfo(clevelandSmall, "rna")

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

data(clevelandSmall)
Cleveland_mProf <- molecularProfiles(clevelandSmall, "rna")

molecularProfiles(clevelandSmall, "rna") <-
  molecularProfiles(clevelandSmall, "rna")

data(clevelandSmall)
molecularProfilesSlot(clevelandSmall)

data(clevelandSmall_cSet)
molecularProfilesSlot(clevelandSmall_cSet) <- molecularProfilesSlot(clevelandSmall_cSet)

name(clevelandSmall)

data(clevelandSmall)
pertNumber(clevelandSmall)

pertNumber(clevelandSmall) <- pertNumber(clevelandSmall)

data(clevelandSmall)
phenoInf <- phenoInfo(clevelandSmall, mDataType="rna")

phenoInfo(clevelandSmall, mDataType="rna") <- phenoInfo(clevelandSmall,
  mDataType="rna")

data(clevelandSmall)
sensNumber(clevelandSmall)

sensNumber(clevelandSmall) <- sensNumber(clevelandSmall)

sensInf<- sensitivityInfo(clevelandSmall)

sensitivityInfo(clevelandSmall) <- sensitivityInfo(clevelandSmall)

data(clevelandSmall)
sensMeas <- sensitivityMeasures(clevelandSmall)
```

```
sensProf <- sensitivityProfiles(clevelandSmall)

sensitivityProfiles(clevelandSmall) <- sensitivityProfiles(clevelandSmall)

data(clevelandSmall)
sensitivityRaw(clevelandSmall)

data(clevelandSmall)
sensitivityRaw(clevelandSmall) <- sensitivityRaw(clevelandSmall)

data(clevelandSmall)
sensitivitySlot(clevelandSmall)

data(clevelandSmall)
sensitivitySlot(clevelandSmall) <- sensitivitySlot(clevelandSmall)
```

`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=radiationTypes(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  <- radiationTypes(clevelandSmall)
clevelandCells <- cellNames(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=cellNames(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')
```

[,RadioSet,ANY,ANY,ANY-method

'['

Description

'['

Usage

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

Arguments

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

Value

Returns the subsetted RSet

Index