Package 'ToxicoGx'

March 30, 2021

Type Package

Title Analysis of Large-Scale Toxico-Genomic Data

Version 1.0.1

Date 2020-11-15

Description Contains a set of functions to perform large-scale analysis of toxicogenomic data, providing a standardized data structure to hold information relevant to annotation, visualization and statistical analysis of toxicogenomic data.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.0), CoreGx

Imports SummarizedExperiment, S4Vectors, Biobase, BiocParallel, ggplot2, tibble, dplyr, caTools, downloader, magrittr, methods, reshape2, tidyr, data.table, assertthat, scales, graphics, grDevices, parallel, stats, utils, limma

RoxygenNote 7.1.1

VignetteBuilder knitr

Suggests rmarkdown, testthat, BiocStyle, knitr, tinytex, devtools, PharmacoGx, xtable

biocViews GeneExpression, Pharmacogenetics, Pharmacogenomics, Software

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

git_branch RELEASE_3_12

git_last_commit e5d0924

git_last_commit_date 2020-11-16

Date/Publication 2021-03-29

Author Sisira Nair [aut],
Esther Yoo [aut],
Christopher Eeles [aut],
Amy Tang [aut],
Nehme El-Hachem [aut],
Petr Smirnov [aut],
Benjamin Haibe-Kains [aut, cre]

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

R topics documented:

`annotation` *annotation Slot Getter*

Description

annotation Slot Getter

Usage

```
annotation(object, ...)
```

Arguments

object	A ToxicoSet
...	A list to allow definition of new parameters on this generic

Value

A list of named annotaiton

Examples

```
data(TGGATESsmall)
annotation(TGGATESsmall)
```

---

annotation<- *annotation<- Slot Setter*

Description

annotation<- Slot Setter

Usage

```
annotation(object, ...) <- value
```

Arguments

object	A ToxicoSet
...	A list to allow definition of new parameters on this generic
value	A list of annotations to add to the annotatiosn slot of an tSet

Value

A copy of the ToxicoSet with the updated annotation slot

Examples

```
data(TGGATESsmall)
annotation(TGGATESsmall) <- annotation(TGGATESsmall)
```

`availableTSets` *Return a table of ToxicoSets available for download*

Description

The function fetches a table of all ToxicoSets 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

```
availableTSets(
  saveDir = tempdir(),
  myfn = "availableToxicoSets.csv",
  verbose = TRUE
)
```

Arguments

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

Value

A data.frame with details about the available ToxicoSet objects

Examples

```
availableTSets()
```

`checkTSetStructure` *A function to verify the structure of a ToxicoSet*

Description

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

Usage

```
checkTSetStructure(tSet, plotDist = FALSE, result.dir = ".")
```

Arguments

`tSet`	A `ToxicoSet` 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

```
checkTSetStructure(TGGATESsmall)
```

`computeAUC` *Computes the AUC for a Drug Dose Viability Curve*

Description

Returns the AUC (Area Under the drug response Curve) given concentration and viability as input, normalized by the concentration range of the experiment. The area returned is the response (1-Viablility) area, i.e. area under the curve when the response curve is plotted on a log10 concentration scale, with high AUC implying high sensitivity to the drug. The function can calculate both the area under a fitted Hill Curve to the data, and a trapz numeric integral of the actual data provided. Alternatively, the parameters of a Hill Slope returned by logLogisticRegression can be passed in if they already known.

Usage

```
computeAUC(
  concentration,
  viability,
  Hill_fit,
  conc_as_log = FALSE,
  viability_as_pct = TRUE,
  trunc = TRUE,
  area.type = c("Fitted", "Actual"),
  verbose = TRUE
)
```

Arguments

`concentration`	[vector] is a vector of drug concentrations.
`viability`	[vector] is a vector whose entries are the viability values observed in the presence of the drug concentrations whose logarithms are in the corresponding entries of conc, where viability 0 indicates that all cells died, and viability 1 indicates that the drug had no effect on the cells.
`Hill_fit`	[list or vector] In the order: c("Hill Slope", "E_inf", "EC50"), the parameters of a Hill Slope as returned by logLogisticRegression. If conc_as_log is set then the function assumes logEC50 is passed in, and if viability_as_pct flag is set, it assumes E_inf is passed in as a percent. Otherwise, E_inf is assumed to be a decimal, and EC50 as a concentration.
`conc_as_log`	[logical], if true, assumes that log10-concentration data has been given rather than concentration data.

viability_as_pct
: [logical], if false, assumes that viability is given as a decimal rather than a percentage, and returns AUC as a decimal. Otherwise, viability is interpreted as percent, and AUC is returned 0-100.

trunc
: [logical], if true, causes viability data to be truncated to lie between 0 and 1 before curve-fitting is performed.

area.type
: Should the area be computed using the actual data ("Actual"), or a fitted curve ("Fitted")

verbose
: [logical], if true, causes warnings thrown by the function to be printed.

Value

Numeric AUC value

Examples

```
dose <- c("0.0025","0.008","0.025","0.08","0.25","0.8","2.53","8")
viability <- c("108.67","111","102.16","100.27","90","87","74","57")
computeAUC(dose, viability)
```

---

computeIC50 *Computes the ICn for any n in 0-100 for a Drug Dose Viability Curve*

Description

Returns the ICn for any given nth percentile when given concentration and viability as input, normalized by the concentration range of the experiment. A Hill Slope is first fit to the data, and the ICn is inferred from the fitted curve. Alternatively, the parameters of a Hill Slope returned by logLogisticRegression can be passed in if they already known.

Usage

```
computeIC50(
  concentration,
  viability,
  Hill_fit,
  conc_as_log = FALSE,
  viability_as_pct = TRUE,
  verbose = TRUE,
  trunc = TRUE
)

computeICn(
  concentration,
  viability,
  Hill_fit,
  n,
  conc_as_log = FALSE,
  viability_as_pct = TRUE,
```

```
  verbose = TRUE,
  trunc = TRUE
)
```

Arguments

concentration	[vector] is a vector of drug concentrations.
viability	[vector] is a vector whose entries are the viability values observed in the presence of the drug concentrations whose logarithms are in the corresponding entries of conc, where viability 0 indicates that all cells died, and viability 1 indicates that the drug had no effect on the cells.
Hill_fit	[list or vector] In the order: c("Hill Slope", "E_inf", "EC50"), the parameters of a Hill Slope as returned by logLogisticRegression. If conc_as_log is set then the function assumes logEC50 is passed in, and if viability_as_pct flag is set, it assumes E_inf is passed in as a percent. Otherwise, E_inf is assumed to be a decimal, and EC50 as a concentration.
conc_as_log	[logical], if true, assumes that log10-concentration data has been given rather than concentration data, and that log10(ICn) should be returned instead of ICn.
viability_as_pct	[logical], if false, assumes that viability is given as a decimal rather than a percentage, and that E_inf passed in as decimal.
verbose	[logical], if true, causes warnings thrown by the function to be printed.
trunc	[logical], if true, causes viability data to be truncated to lie between 0 and 1 before curve-fitting is performed.
n	[numeric] The percentile concentration to compute. If viability_as_pct set, assumed to be percentage, otherwise assumed to be a decimal value.

Value

a numeric value for the concentration of the nth precentile viability reduction

Functions

- computeIC50: Returns the IC50 of a Drug Dose response curve

Examples

```
dose <- c("0.0025","0.008","0.025","0.08","0.25","0.8","2.53","8")
viability <- c("108.67","111","102.16","100.27","90","87","74","57")
computeIC50(dose, viability)
computeICn(dose, viability, n=10)
```

computeLimmaDiffExpr,ToxicoSet-method

Conduct differential expression analysis using the limma R pacakge

Description

WARNING: This function can take a very long time to compute!

Usage

```
## S4 method for signature 'ToxicoSet'
computeLimmaDiffExpr(object, buildTable = TRUE)
```

Arguments

object	A ['ToxicoSet'] object with a molecular profile named 'rna'
buildTable	['logical'] Should the result of the eBayes function from limma be assembled into a data.table containing the result along with the gene, compound and durations names. Default it TRUE, otherwise this function with return the object produced by eBayes.

Value

A ['data.table'] containing the results the limma differential expression analysis comparing control vs each dose level for each compound within each duration.

Examples

```
if (interactive()) {
data(TGGATESsmall)
analysis <- computeLimmaDiffExpr(TGGATESsmall)
}
```

curation

curation Slot Getter

Description

curation Slot Getter

Usage

```
curation(object, ...)
```

Arguments

object	A ToxicoSet
...	A list to allow definition of new parameters on this generic

Value

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

Examples

```
data(TGGATESsmall)
curation(TGGATESsmall)
```

---

curation<- *curation<- Slot Setter*

Description

curation<- Slot Setter

Usage

```
curation(object, ...) <- value
```

Arguments

`object`	A `ToxicoSet`
`...`	A `list` to allow definition of new parameters on this generic
`value`	A `list` of curations for the cell and tissues types in the tSet object

Value

A copy of the `ToxicoSet` with the updated curation slot

Examples

```
data(TGGATESsmall)
curation(TGGATESsmall) <- curation(TGGATESsmall)
```

---

datasetType *datasetType Generic*

Description

A generic for retrieving the dataset type of an tSet object

Usage

```
datasetType(object, ...)
```

Arguments

object	A ToxicoSet from which to retrieve the dataset type
...	A list containing fall through arguments; this allows addition of new parameters to methods for this generic

Value

A character vector containing the dataset type

---

datasetType<- *datasetType<- Replacement Generic*

Description

A generic for updating the dataset type of a ToxicoSet object

Usage

```
datasetType(object) <- value
```

Arguments

object	A ToxicoSet from which to retrieve the dataset type
value	A character vector containing the dataset type

Value

A ToxicoSet with the datasetType slot updated

Examples

```
data(TGGATESsmall)
datasetType(TGGATESsmall)
```

---

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

Description

Get the dimensions of a ToxicoSet

Usage

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

Arguments

x	ToxicoSet

Value

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

Examples

```
data(TGGATESsmall)
dim(TGGATESsmall)
```

`downloadTSet` *Download a ToxicoSet object*

Description

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

Usage

```
downloadTSet(name, saveDir = tempdir(), tSetFileName = NULL, verbose = TRUE)
```

Arguments

`name`	`Character` string, the name of the PhamracoSet to download.
`saveDir`	`Character` string with the folder path where the ToxicoSet should be saved. Defaults to `'./tSets/'`. Will create directory if it does not exist.
`tSetFileName`	`character` string, the file name to save the dataset under
`verbose`	`bool` Should status messages be printed during download. Defaults to TRUE.

Value

A tSet object with the dataset, downloaded from our server

Examples

```
if (interactive()) {
drugMatrix_rat <- downloadtSet("drugMatrix_rat")
}
```

drugGeneResponseCurve *Compares gene expression for a specificed set of features over specific drug dosages vs time*

Description

This function generates a plot visualizing the relationship between gene expression, time and dose level for the selected tSet. The plot is generated with ggplot2 and can be customized using ggplot plot + function() syntax.

Usage

```
drugGeneResponseCurve(
  tSet,
  duration = NULL,
  cell_lines = NULL,
  mDataTypes = NULL,
  features = NULL,
  dose = NULL,
  drug = NULL,
  summarize_replicates = TRUE,
  line_width = 1,
  point_size = 2.5,
  ggplot_args = NULL,
  verbose = TRUE
)
```

Arguments

tSet	ToxicoSet A ToxicoSet to be plotted in this graph. Currently only a single tSet is supported.
duration	character A vector of durations to include in the plot.
cell_lines	character A vector of cell lines to include in the plot.
mDataTypes	vector A vector specifying the molecular data types to include in this plot. Defaults to the first mDataType if not specified.ex This release version only accepts one mDataType, more to be added in forthcoming releases.
features	character A vector of feature names to include in the plot. If you specify more than two dose levels, you may only pass in up to two features.
dose	character A vector of dose levels to be included in the plot. Default to include all dose levels available for a drug. If you specify more than two features you may only pass in up to two dose levels.
drug	character A drug name to include in this plot. See drugNames(tSet) for a list of options.
summarize_replicates	logical If TRUE will average viability across replicates for each unique drug-dose-duration combination.
line_width	numeric A number specifying the thickness of lines in the plot, as passed to size in geom_line(). Defaults to 1.

point_size	numeric A number specifying how large points should be in the plot, as passed to size in geom_point(). Defaults to 2.5.
ggplot_args	list A list of ggplot2 functions which can be called using the plot + function() syntax. This allows arbitrary customization of the plot including changing the title, axis labels, colours, etc. Please see the included examples for basic usage or ggplot2 documentation for advanced customization.
verbose	boolean Should warning messages about the data passed in be printed?

Value

Plot of the viabilities for each drug vs time of exposure

Examples

```
if (interactive()) {
  drugGeneResponseCurve(TGGATESsmall, dose = c("Control", "Low", "Middle"),
  mDataTypes="rna", drug = drugNames(TGGATESsmall)[1],
  duration = c("2", "8", "24"), features = "ENSG00000002726_at")
}
```

---

drugInfo *drugInfo Getter*

Description

Get the drug annotations in a ToxicoSet object

Usage

```
drugInfo(object)
```

Arguments

object	A ToxicoSet object

Value

a data.frame with the drug annotations

Examples

```
data(TGGATESsmall)
drugInfo <- drugInfo(TGGATESsmall)
```

drugInfo<- *drugInfo<- Setter method*

Description

Set the drug annotations in a ToxicoSet object

Usage

```
drugInfo(object) <- value
```

Arguments

object	A ToxicoSet object.
value	A data.frame of replacement values.

Value

Updated ToxicoSet

Examples

```
data(TGGATESsmall)
drugInfo(TGGATESsmall) <- drugInfo(TGGATESsmall)
```

drugNames *drugNames Generic*

Description

A generic for the drugNames method

Usage

```
drugNames(object)
```

Arguments

object	A ToxicoSet object from which to retrieve the included drug names

Value

A vector of the drug names used in the ToxicoSet

Examples

```
data(TGGATESsmall)
drugName <- drugNames(TGGATESsmall)[seq_len(10)]
```

drugNames<- *drugNames<- Generic*

Description

A generic for the drugNames replacement method

Usage

```
drugNames(object) <- value
```

Arguments

object	A ToxicoSet object to modify
value	A character vector of replacement drug names

Value

Updated ToxicoSet

Examples

```
data(TGGATESsmall)
drugNames(TGGATESsmall) <- drugNames(TGGATESsmall)
```

drugPerturbationSig *Drug perturbation analysis*

Description

Creates a signature representing gene expression (or other molecular profile) change induced by administrating a drug, for use in drug effect analysis.

Usage

```
drugPerturbationSig(
  tSet,
  mDataType,
  drugs = NULL,
  cell_lines = NULL,
  features = NULL,
  duration = NULL,
  dose = NULL,
  nthread = 1,
  returnValues = c("estimate", "tstat", "pvalue", "fdr"),
  verbose = FALSE
)
```

Arguments

tSet	ToxicoSet a ToxicoSet 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 (only rna currently supported)
drugs	character a vector of drug names for which to compute the signatures. Should match the names used in the ToxicoSet.
cell_lines	character a vector of cell names to use in computing the signatures. Should match the names used in the ToxicoSet.
features	character a vector of features for which to compute the signatures. Should match the names used in correspondant molecular data in ToxicoSet.
duration	character a vector of experiment durations for which to inlcude in the computed the signatures.
dose	character a vector of dose levels to include in the results
nthread	numeric if multiple cores are available, how many cores should the computation be parallelized over?
returnValues	character Which of estimate, t-stat, p-value and fdr should the function return for each gene drug pair
verbose	bool Should diagnostive messages be printed? (default false)

Details

Given a Toxicoset of the perturbation experiment type, and a character vector of drugs, the function will compute a signature for the effect of drug concentration on the molecular profile of a cell. The algorithm uses a regression model which corrects for experimental batch effects, cell specific differences, and duration of experiment to isolate the effect of the concentration of the drug applied. The function returns the estimated coefficient for concentration, 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.

Value

ToxicoSig An object composed of a 3D array with genes in the first dimension, drugs in the second, and return values in the third.

Examples

```
if (interactive()) {
data(TGGATESsmall)
drug.perturbation <- drugPerturbationSig(TGGATESsmall, mDataType="rna", features = head(fNames(TGGATESsmall,
}
```

`drugTimeResponseCurve` *Compares viabilities at a given dose over different experimental durations*

Description

This function generates a plot visualizing the relationship between gene expression, time and dose level for the selected tSet. The plot is generated with ggplot2 and can be customized using ggplot plot + function() syntax.

Usage

```
drugTimeResponseCurve(
  tSet,
  duration = NULL,
  cell_lines = NULL,
  dose = NULL,
  drugs = NULL,
  summarize_replicates = TRUE,
  line_width = 1,
  point_size = 2.5,
  verbose = TRUE,
  ggplot_args = NULL
)
```

Arguments

`tSet`	`ToxicoSet` A ToxicoSet to be plotted in this figure
`duration`	`character` A vector of durations to include in the plot.
`cell_lines`	`character` A vector of cell lines to include in the plot.
`dose`	`character` A vector of dose levels to be included in the plot. Default to include all dose levels available for a drug. Must include at minimum two dose levels, one of witch is "Control".
`drugs`	`character` A drugs or pair of drugs to be plotted.
`summarize_replicates`	`logical` If TRUE will average viability across replicates for each unique drug-dose-duration combination.
`line_width`	`numeric` A number specifying the thickness of lines in the plot, as passed to size in geom_line(). Defaults to 1.
`point_size`	`numeric` A number specifying how large points should be in the plot, as passed to size in geom_point(). Defaults to 2.5.
`verbose`	`boolean` Should warning messages about the data passed in be printed?
`ggplot_args`	`list` A list of ggplot2 functions which can be called using the plot + function() syntax. This allows arbitrary customization of the plot including changing the title, axis labels, colours, etc. Please see the included examples for basic usage or ggplot2 documentation for advanced customization. Alternatively, you could assign the return value to a variable and add the customization yourself using plot + function().

Value

Plot of the viabilities for each drugs vs time of exposure

Examples

```
library(ggplot2)

# Default settings
plot <- drugTimeResponseCurve(TGGATESsmall, cell_lines = "Hepatocyte",
dose = c("Control", "Low", "Middle"), drugs = drugNames(TGGATESsmall)[6],
duration = c("2", "8", "24"))

# Customize title, x/y labels, x/y limits, colour palette and define
# custom ticks for x axis using the function argument ggplot2_args
customizations <- list(labs(title= 'My Custom Title', ylab = 'The y-axis'),
                       xlim(c(2, 24)), ylim(c(99,105)),
                       scale_color_brewer(palette="Set1"),
                       scale_x_continuous(breaks=c(2, 8, 24),
                         labels = c("Two", "Eight", "Twenty-Four"))
                       )

if(interactive()) {
   drugTimeResponseCurve(TGGATESsmall, cell_lines = "Hepatocyte",
     dose = c("Control", "Low", "Middle"),
     drugs = drugNames(TGGATESsmall)[6], duration = c("2", "8", "24"),
     ggplot_args = customizations)
}

# Customize the plot using standard ggplot2 syntax
if(interactive()) {
   plot + labs(title= 'My Custom Title', ylab = 'The y-axis') +
     xlim(c(2, 24)) + ylim(c(99,105)) + scale_color_brewer(palette="Set1")
}
```

HCC_sig *HCC_sig dataset*

Description

A dataset cotaining the gene names associated with the HCC geneset signature

Usage

```
data(HCC_sig)
```

Format

character

`logLogisticRegression` *Fits curves of the form E = E_inf + (1 - E_inf)/(1 + (c/EC50)^HS) to dose-response data points (c, E) given by the user and returns a vector containing estimates for HS, E_inf, and EC50.*

Description

By default, logLogisticRegression uses an L-BFGS algorithm to generate the fit. However, if this fails to converge to solution, logLogisticRegression samples lattice points throughout the parameter space. It then uses the lattice point with minimal least-squares residual as an initial guess for the optimal parameters, passes this guess to drm, and re-attempts the optimization. If this still fails, logLogisticRegression uses the PatternSearch algorithm to fit a log-logistic curve to the data.

Usage

```
logLogisticRegression(
  conc,
  viability,
  density = c(2, 10, 2),
  step = 0.5/density,
  precision = 0.05,
  lower_bounds = c(0, 0, -6),
  upper_bounds = c(4, 1, 6),
  scale = 0.07,
  family = c("normal", "Cauchy"),
  median_n = 1,
  conc_as_log = FALSE,
  viability_as_pct = TRUE,
  trunc = TRUE,
  verbose = FALSE
)
```

Arguments

`conc`	[vector] is a vector of drug concentrations.
`viability`	[vector] is a vector whose entries are the viability values observed in the presence of the drug concentrations whose logarithms are in the corresponding entries of the log_conc, where viability 0 indicates that all cells died, and viability 1 indicates that the drug had no effect on the cells.
`density`	[vector] is a vector of length 3 whose components are the numbers of lattice points per unit length along the HS-, E_inf-, and base-10 logarithm of the EC50-dimensions of the parameter space, respectively.
`step`	[vector] is a vector of length 3 whose entries are the initial step sizes in the HS, E_inf, and base-10 logarithm of the EC50 dimensions, respectively, for the PatternSearch algorithm.
`precision`	is a positive real number such that when the ratio of current step size to initial step size falls below it, the PatternSearch algorithm terminates. A smaller value will cause LogisticPatternSearch to take longer to complete optimization, but will produce a more accurate estimate for the fitted parameters.

lower_bounds	[vector] is a vector of length 3 whose entries are the lower bounds on the HS, E_inf, and base-10 logarithm of the EC50 parameters, respectively.
upper_bounds	[vector] is a vector of length 3 whose entries are the upper bounds on the HS, E_inf, and base-10 logarithm of the EC50 parameters, respectively.
scale	is a positive real number specifying the shape parameter of the Cauchy distribution.
family	[character], if "cauchy", uses MLE under an assumption of Cauchy-distributed errors instead of sum-of-squared-residuals as the objective function for assessing goodness-of-fit of dose-response curves to the data. Otherwise, if "normal", uses MLE with a gaussian assumption of errors
median_n	If the viability points being fit were medians of measurements, they are expected to follow a median of family distribution, which is in general quite different from the case of one measurement. Median_n is the number of measurements the median was taken of. If the measurements are means of values, then both the Normal and the Cauchy distributions are stable, so means of Cauchy or Normal distributed variables are still Cauchy and normal respectively.
conc_as_log	[logical], if true, assumes that log10-concentration data has been given rather than concentration data, and that log10(EC50) should be returned instead of EC50.
viability_as_pct	[logical], if false, assumes that viability is given as a decimal rather than a percentage, and that E_inf should be returned as a decimal rather than a percentage.
trunc	[logical], if true, causes viability data to be truncated to lie between 0 and 1 before curve-fitting is performed.
verbose	[logical], if true, causes warnings thrown by the function to be printed.

Value

A vector containing estimates for HS, E_inf, and EC50

Examples

```
dose <- c("0.0025","0.008","0.025","0.08","0.25","0.8","2.53","8")
viability <- c("108.67","111","102.16","100.27","90","87","74","57")
computeAUC(dose, viability)
```

---

mDataNames,ToxicoSet-method

mDataNames

Description

Returns the names of the molecular data types available in a ToxicoSet object

Usage

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

Arguments

object	A ToxicoSet object

Value

Vector of names of the molecular data types

Examples

```
mDataNames(TGGATESsmall)
```

---

name,ToxicoSet-method *name Getter method*

Description

Retrieves the name of a tSet

Usage

```
## S4 method for signature 'ToxicoSet'
name(object)
```

Arguments

object	ToxicoSet A ToxicoSet object

Value

character A string of the tSet's name

Examples

```
name(TGGATESsmall)
```

---

sensitivityRaw *sensitivityRaw Generic*

Description

sensitivityRaw Generic

Usage

```
sensitivityRaw(object, ...)
```

Arguments

object	A ToxicoSet to extract the raw sensitivity data from
...	A list to allow new parameters in specific methods

Value

A array containing the raw sensitivity data

Examples

```
data(TGGATESsmall)
sensitivityRaw(TGGATESsmall)
```

---

sensitivityRaw<- *sensitivityRaw<- Replacement Generic*

Description

sensitivityRaw<- Replacement Generic

Usage

```
sensitivityRaw(object, ...) <- value
```

Arguments

object	A ToxicoSet to extract the raw sensitivity data from
...	A list to allow new parameters in specific methods
value	A array containing the raw dose and viability data for the tSet

Value

A copy of the ToxicoSet containing the updated sensitivty data

Examples

```
data(TGATESsmall)
sensitivityRaw(TGGATESsmall) <- sensitivityRaw(TGGATESsmall)
```

sensitivitySlot *sensitivitySlot Generic*

Description

sensitivitySlot Generic

Usage

```
sensitivitySlot(object, ...)
```

Arguments

object	A ToxicoSet to extract the raw sensitivity data from
...	Allow new parameters to be defined for this generic

Value

A list of the sensitivity slot contents

sensitivitySlot<- *sensitivitySlot<- Replacement Generic*

Description

sensitivitySlot<- Replacement Generic

Usage

```
sensitivitySlot(object, ...) <- value
```

Arguments

object	A ToxicoSet to extract the raw sensitivity data from
...	Allow new parameters to be defined for this generic
value	A list of new sensitivity slot data for the tSet

Value

A copy of the ToxicoSet containing the updated sensitivty slot

show,ToxicoSet-method *Show a ToxicoSet*

Description

Show a ToxicoSet

Usage

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

Arguments

object	A ToxicoSet object to print a summary for

Value

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

Examples

```
TGGATESsmall
```

show,ToxicoSig-method *Show ToxicoGx Signatures*

Description

Show ToxicoGx Signatures

Usage

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

Arguments

object	ToxicoSig

Value

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

Examples

```
data(TGGATESsmall)
drug.perturbation <- drugPerturbationSig(TGGATESsmall, mDataType="rna", nthread = 1, duration = "2",
   drugs = head(drugNames(TGGATESsmall)), features = fNames(TGGATESsmall, "rna")[seq_len(2)])
drug.perturbation
```

showSigAnnot *Show the Annotations of a signature object*

Description

This funtion prints out the information about the call used to compute the drug 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 ToxicoSig Class, as returned by drugPerturbationSig

Value

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

Examples

```
data(TGGATESsmall)
drug.perturbation <- drugPerturbationSig(TGGATESsmall, mDataType="rna", nthread=1, duration = "2",
   drugs = head(drugNames(TGGATESsmall)), features = fNames(TGGATESsmall, "rna")[seq_len(2)])
showSigAnnot(drug.perturbation)
```

subsetTo *A function to subset a ToxicoSet to data containing only specified drugs, cells and genes*

Description

This is the prefered method of subsetting a ToxicoSet. This function allows abstraction of the data to the level of biologically relevant objects: drugs and cells. The function will automatically go through all of the combined data in the ToxicoSet 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(
  object,
  cell_lines = NULL,
  drugs = NULL,
  molecular.data.cells = NULL,
  duration = NULL,
```

```
  features = NULL,
  ...
)
```

Arguments

object	A ToxicoSet to be subsetted
cell_lines	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.
drugs	A list or vector of drug names as used in the dataset to which the object will be subsetted. If left blank, then all drugs will be left in the dataset.
molecular.data.cells	A list or vector of cell names to keep in the molecular data
duration	A list or vector of the experimental durations to include in the subset as strings. Defaults to all durations if parameter is not specified.
features	A list or vector of feature names as used in the dataset from which the object will be subsetted. If left blank that all features will be left in.
...	Other arguments passed to other functions within the package

Value

A ToxicoSet with only the selected drugs and cells

Examples

```
TGGATESDrugNames  <- drugNames(TGGATESsmall)
TGGATESCells <- cellNames(TGGATESsmall)
tSet <- subsetTo(TGGATESsmall,drugs = TGGATESDrugNames[1],
  cells = TGGATESCells[1], duration = "2")
```

---

summarizeMolecularProfiles

Takes molecular data from a ToxicoSet, and summarises them into one entry per drug and experimental condition.

Description

Given a ToxicoSet with molecular data, this function will summarize the data into one profile per experimental condition (duration, dose level) using the chosen summary.stat and return a SummarizedExperiment object, with one Assay corresponding to a requested drug.

Usage

```
summarizeMolecularProfiles(
  tSet,
  mDataType,
  cell_lines = NULL,
  drugs = NULL,
  features = NULL,
```

```
  duration = NULL,
  dose = c("Control", "Low", "Middle", "High"),
  summary.stat = c("mean", "median", "first", "last"),
  fill.missing = TRUE,
  summarize = TRUE,
  verbose = TRUE
)
```

Arguments

tSet	ToxicoSet The ToxicoSet 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 tSet for example: rna
cell_lines	character The cell lines to be summarized. If any cell.line has no data, missing values will be created
drugs	character The drugs to be summarized
features	character A vector of the feature names to include in the summary
duration	character A vector of durations to summarize across
dose	character The dose level to summarize replicates across
summary.stat	character which summary method to use if there are repeated cell_lines? Choices are "mean", "median", "first", or "last"
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

SummarizedExperiment A SummarizedExperiment object with the molecular data summarized per cell line.

Examples

```
data(TGGATESsmall)
summMP <- ToxicoGx::summarizeMolecularProfiles(
  tSet = TGGATESsmall, mDataType = "rna",
  cell_lines=cellNames(TGGATESsmall), drugs = head(drugNames(TGGATESsmall)),
  features = fNames(TGGATESsmall,"rna")[seq_len(100)], duration = "8",
  dose = c("Control", "High"), summary.stat = "median",
  fill.missing = TRUE, verbose=TRUE
  )

#subset into expression matrix for a requested drug
assays <- SummarizedExperiment::assays(summMP)[[drugNames(TGGATESsmall)[1]]]
#summarization of phenoData for requested experiments
phenoData <- SummarizedExperiment::colData(summMP)
#summarization of phenoData for requested experiments
featureData <- SummarizedExperiment::rowData(summMP) #featureData for requested experiments
```

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

Description

This function creates a table with drug as rows and cell lines as columns, summarising the drug senstitivity data of a ToxicoSet into drug-cell line pairs for a specified experiment duration.

Usage

```
summarizeSensitivityProfiles(
  tSet,
  duration = NULL,
  cell_lines = NULL,
  drugs = NULL,
  sensitivity.measure = "auc_recomputed",
  summary.stat = c("mean", "median", "first", "last", "max", "min"),
  fill.missing = TRUE,
  verbose = TRUE
)
```

Arguments

tSet	ToxicoSet The ToxicoSet from which to extract the data
duration	numeric The duration at which to summarize the drug-cell combo. This is a required parameter.
cell_lines	character The cell lines to be summarized. If any cell lines has no data, it will be filled with missing values
drugs	character The drugs to be summarized. If any drugs has no data, it will be filled with missing values. Defaults to include all drugs in the given tSet.
sensitivity.measure	character which sensitivity sensitivity.measure to use? Use the sensitivityMeasures function to find out what measures are available for each TSet.
summary.stat	character which summary method to use if there are repeated cell line-drug experiments? Choices are "mean", "median", "first", "last", "max", or "min"
fill.missing	boolean 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 drugs going down the rows, cell lines across the columns, with the selected sensitivity statistic for each pair.

Examples

```
data(TGGATESsmall)
TGGATESauc <- summarizeSensitivityProfiles(TGGATESsmall, sensitivity.measure='auc_recomputed')
```

TGGATESsmall *TGGATESsmall dataset*

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 ToxicoGx.

Usage

```
data(TGGATESsmall)
```

Format

ToxicoSet object

References

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

ToxicoSet *ToxicoSet constructor*

Description

A constructor that simplifies the process of creating ToxicoSets, 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 ToxicoSet methods. For a much more detailed instruction on creating ToxicoSets, please see the "CreatingToxicoSet" vignette.

Usage

```
ToxicoSet(
  name,
  molecularProfiles = list(),
  cell = data.frame(),
  drug = 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)),
  curationDrug = data.frame(),
  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 containing the annotations for all the cell lines profiled in the data set, across all data types
drug	A data.frame containing the annotations for all the drugs 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 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
curationCell, curationDrug, curationTissue	A data.frame mapping the names for cells, drugs, and tissues used in the data set to universal identifiers used between different ToxicoSet objects
datasetType	A character string of "sensitivity", "perturbation", or both detailing what type of data can be found in the ToxicoSet, for proper processing of the data
verify	boolean Should the function verify the ToxicoSet and print out any errors it finds after construction?

Value

An object of class ToxicoSet

ToxicoSet-class *Class to contain Toxico-genomic Data*

Description

The ToxicoSet (tSet) class was development to contain and organise large ToxicGenomic datasets as well as provide useful tools for interacting with this data. Functions are included for exploring the relationship between survival fraction and gene expression in cultured human and rat tissues during exposure to a wide ranges of compounds. Features include plotting dose and exposure time curves, calculating AUC, fitting linear models and computing sensitivity signatures.

Get the cell line annotations in a ToxicoSet

Set cell line annotations for a ToxicoSet object

Get names of cell lines in a ToxicoSet object

Set the cell line names in a TocicoSet object

Get the date a ToxicoSet object was created

Get the feature names in a ToxicoSet object for the specified molecular data type

Set the feature names in a ToxicoSet object for the specified molecular data type

Set the feature annotations for a specficied molecular data type

Get the molecular profile data associated with the specific molecular data

Set the molecular profile data assciated with the specificied molecular data type

Get an array of the number of pertubration experiments per drug and cell line in a ToxicoSet object

Set the number of perturbation experiments per drug and cell line and molecular data type in a ToxicoSet object

Get the phenotype annotations for cell lines with the specificed molecular data type

Set the phenotype annotations for cell lines with the selected molecular data type.

Get the number of sensitivity experiments per drug and cell line in a ToxicoSet

Set the number of sensitivity experiments per drug and cell line in a ToxicoSet object

Get the annotations for the sensitivity experiments in the ToxicoSet

Set the annotations for sensitivity experiments in this ToxicSet

Get the avilable measurments for sensitivity experiments in a ToxicoSet

Get the data for sensitivty experiments on cell lines in a ToxicoSet

Set the data for sensitivity experiments on cell lines in a ToxicoSet

Usage

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

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

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

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

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

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

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

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

## S4 replacement method for signature 'ToxicoSet'
datasetType(object) <- value

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

## S4 method for signature 'ToxicoSet'
```

```
datasetType(object)

## S4 method for signature 'ToxicoSet'
drugInfo(object)

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

## S4 method for signature 'ToxicoSet'
drugNames(object)

## S4 replacement method for signature 'ToxicoSet,character'
drugNames(object) <- value

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

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

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

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

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

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

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

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

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

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

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

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

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

```
## S4 method for signature 'ToxicoSet'
sensNumber(object)

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

## S4 method for signature 'ToxicoSet'
sensitivityInfo(object)

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

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

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

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

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

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

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

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

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

Arguments

object	A ToxicoSet to extract the raw sensitivity data from
value	A list of new sensitivity slot data for the tSet
mDataType	A character with the type of molecular data to return/update
assay	character Name or index of the assay data to return
...	Allow new parameters to be defined for this generic

Value

An object of the ToxicoSet class

a data.frame with the cell annotations

Updated ToxicoSet

A vector of the cell names used in the ToxicoSet

Updated ToxicoSet

The date the ToxicoSet was created

A character vector of the feature names

Updated ToxicoSet

Updated ToxicoSet

Updated ToxicoSet

A list containing the molecularProfiles from a tSet

A copy of the ToxicoSet with the molecularProfiles slot updated

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

The updated ToxicoSet

a Dframe with the experiment info

The updated ToxicoSet

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

The updated ToxicoSet

a data.frame with the experiment info

Updated ToxicoSet

A character vector of all the available sensitivity measures

a data.frame with the experiment info

Updated ToxicoSet

Methods (by generic)

- annotation: Retrieve the annotations slot form an tSet
- annotation<-: Update the annotation slot of a tSet
- cellInfo: Returns the annotations for all the cell lines tested on in the ToxicoSet
- cellInfo<-: Returns the annotations for all the cell lines tested on in the ToxicoSet
- 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 tSet
- curation<-: Update the annotation slot of a tSet
- datasetType<-: Update the dataset type of an tSet and return a copy of the updated object
- dateCreated: Return the date the ToxicoSet was created
- datasetType: Update the dataset type of an tSet and return a copy of the updated object
- drugInfo: Returns the annotations for all the drugs tested in the ToxicoSet
- drugInfo<-: Update the drug annotations
- drugNames: Return the names of the drugs used in the ToxicoSet
- drugNames<-: Update the drug names used in the dataset
- fNames: Return the feature names used in the dataset
- fNames<-: Update the feature names used in the dataset
- 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 ToxicoSet
- molecularProfiles<-: Update the given type of molecular data from the ToxicoSet
- molecularProfiles<-: Update the given type of molecular data from the ToxicoSet
- molecularProfilesSlot: Get contents of molecularProfiles slot
- molecularProfilesSlot<-: Update the molecular profiles slot of a ToxicoSet and returns the updated copy
- 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 ToxicoSet
- phenoInfo<-: Update the the given type of molecular data experiment info in the ToxicoSet
- sensNumber: Return the summary of available sensitivity experiments
- sensNumber<-: Update the summary of available sensitivity experiments
- sensitivityInfo: Return the drug 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 drug dose sensitivity
- sensitivityProfiles<-: Update the phenotypic data for the drug dose sensitivity
- sensitivityProfiles<-: Update the phenotypic data for the drug dose sensitivity
- sensitivityRaw: Retrive the raw dose and viability data from an tSet
- sensitivityRaw<-: Set the raw dose and viability data for a tSet and return and updated copty
- sensitivitySlot: Retrieves the contents of the sensitivity slot
- sensitivitySlot<-: Set the raw dose and viability data for an tSet and return and updated copy

Slots

annotation A list of annotation data about the ToxicoSet, 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 SummarizedExperiment type object for holding data for RNA, DNA, SNP and CNV measurements, 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

drug A data.frame containg the annotations for all the drugs profiled in the data set, 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-drug pair

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

curation A list containing mappings for $drug, cell, tissue names used in the data set to universal identifiers used between different ToxicoSet objects

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

Examples

```
data(TGGATESsmall)
cellInfo <- cellInfo(TGGATESsmall)

data(TGGATESsmall)
cellInfo(TGGATESsmall) <- cellInfo(TGGATESsmall)

cellNames(TGGATESsmall)

data(TGGATESsmall)
cellNames(TGGATESsmall) <- cellNames(TGGATESsmall)

dateCreated(TGGATESsmall)

data(TGGATESsmall)
datasetType(TGGATESsmall)

fNames(TGGATESsmall, "rna")[seq_len(10)]

data(TGGATESsmall)
cellNames(TGGATESsmall) <- cellNames(TGGATESsmall)

data(TGGATESsmall)
featureInfo <- featureInfo(TGGATESsmall, "rna")[seq_len(10),]

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

data(TGGATESsmall)
TGGATES_mProf <- molecularProfiles(TGGATESsmall, "rna")[seq_len(10),]

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

data(TGGATESsmall)
molecularProfilesSlot(TGGATESsmall)

data(TGGATESsmall)
molecularProfilesSlot(TGGATESsmall) <- molecularProfilesSlot(TGGATESsmall)

pertNumber(TGGATESsmall)

pertNumber(TGGATESsmall) <- pertNumber(TGGATESsmall)

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

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

sensNumber(TGGATESsmall)

sensNumber(TGGATESsmall) <- sensNumber(TGGATESsmall)

data(TGGATESsmall)
```

```
sensInf<- sensitivityInfo(TGGATESsmall)[seq_len(10),]

data(TGGATESsmall)
sensitivityInfo(TGGATESsmall) <- sensitivityInfo(TGGATESsmall)

sensitivityMeasures(TGGATESsmall)

data(TGGATESsmall)
sensProf <- sensitivityProfiles(TGGATESsmall)

sensitivityProfiles(TGGATESsmall) <- sensitivityProfiles(TGGATESsmall)

data(TGGATESsmall)
sensitivitySlot(TGGATESsmall)

data(TGGATESsmall)
sensitivitySlot(TGGATESsmall) <- sensitivitySlot(TGGATESsmall)
```

`[,ToxicoSet,ANY,ANY,ANY-method`
'['

Description

'['

Usage

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

Arguments

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

Value

Returns the subsetted tSet

Examples

```
tSet <- TGGATESsmall[cellNames(TGGATESsmall), drugNames(TGGATESsmall)[seq_len(3)]]
```

Index