Package 'ToxicoGx'

October 15, 2019

Type Package

Title Analysis of Large-Scale Toxicogenomic Data

Version 0.0.1

Date 2019-10-14

Description Contains a set of functions to perform large-scale analysis of toxicogenomic data. A member of the BHK lab's Gx suite of software, this package provides a standardized data structure to hold information relevant to annotation, visualization and statistical analysis of toxicogenomic data. This package is in early release, and more features will be added over the next few months.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0), CoreGx

Imports SummarizedExperiment, Biobase, RColorBrewer, caTools, magicaxis, methods, reshape2, KernSmooth, cluster, Matrix, scales

RoxygenNote 6.1.1

VignetteBuilder knitr

Suggests testthat (>= 2.1.0), knitr

NeedsCompilation no

Author Sisira Nair [aut],
Esther Yoo [aut],
Christopher Eeles [aut],
Amy Tang [aut],
Nehme El-Hachem1 [aut],
Petr Smirnov [aut],
Benjamin Haibe-Kains [aut, cre]

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

R topics documented:

`.saveTSetAsGroundTruth` *Generate data to test a tSet object and class accessor methods*

Description

Unit tests in this package will assume that this data is the ground truth and any deviation from these files represents a breakdown in the curation and/or accessor methods of the tSet object

Usage

```
.saveTSetAsGroundTruth(tSet)
```

Arguments

tSet	A ToxicoSet object to be used as ground truth for tests

---

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)
```

---

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)
```

drugDoseResponseCurve *Plot drug response curve of a given drug and a given cell for a list of tSets (objects of the ToxicoSet class).*

Description

Given a list of ToxicoSets, 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 concentrations. If more than one tSet is provided, a light gray area would show the common concentration range between tSets. 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 concentrations and viability values, as in the examples below, and it will be treated as experiments equivalent to values coming from a tSet. The names of the concentration list determine the legend labels.

Usage

```
drugDoseResponseCurve(drug, cellline, durations, tSets = list(),
  concentrations = list(), viabilities = list(), conc_as_log = FALSE,
  viability_as_pct = TRUE, trunc = TRUE,
  legends.label = c("ic50_published", "gi50_published", "auc_published",
  "auc_recomputed", "ic50_recomputed"), ylim = c(0, 100), xlim, mycol,
  title, plot.type = c("Fitted", "Actual", "Both"),
  summarize.replicates = TRUE, lwd = 0.5, cex = 0.5,
  cex.main = 0.9, legend.loc = "topright", verbose = TRUE)
```

Arguments

drug	[string] A drug name for which the drug response curve should be plotted. If the plot is desirable for more than one toxico set, A unique drug id should be provided.
cellline	[string] A cell line name for which the drug response curve should be plotted. If the plot is desirable for more than one toxico set, A unique cell id should be provided.
durations	[numeric] A duration for which the drug response curve should be plotted.
tSets	[list] a list of ToxicoSet objects, for which the function should plot the curves.
concentrations, viabilities	[list] A list of concentrations and viabilities to plot, the function assumes that concentrations[[i]] is plotted against viabilities[[i]]. The names of the concentration list are used to create the legend labels
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. Applies only to the concentrations parameter.
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. Applies only to the viabilities parameter.

trunc	[bool] Should the viability values be truncated to lie in [0-100] before doing the fitting
legends.label	[vector] A vector of sensitivity measurment types. A legend will be displayed on the top right of the plot which each line of the legend is the values of requested sensitivity measurements for one of the requested tSets. If this parameter is missed no legend would be provided for the plot.
ylim	[vector] A vector of two numerical values to be used as ylim of the plot. If this parameter would be missed c(0,100) would be used as the ylim of the plot.
xlim	[vector] A vector of two numerical values to be used as xlim of the plot. If this parameter would be missed the minimum and maximum comncentrations between all the tSets would be used as plot xlim.
mycol	[vector] A vector with the same length of the tSets parameter which will determine the color of the curve for the toxico 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	[boolean] Should warning messages about the data passed in be printed?

Value

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

Examples

```
if (interactive()) {
drugDoseResponseCurve(concentrations=list("Experiment 1"=c(.008, .04, .2, 1)),
 viabilities=list(c(100,50,30,1)), plot.type="Both")
}
```

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.

Usage

```
drugGeneResponseCurve(tSets, duration, cellline, mDataTypes,
  features = NULL, dose, drug, plot.type = "Actual",
  summarize.replicates = FALSE, xlim = c(0, 24), ylim = c(0, 15),
  mycol, x.custom.ticks = NULL, title, lwd = 1.5, cex = 0.7,
  cex.main = 1, legend.loc = "topright", verbose = TRUE)
```

Arguments

tSets	[ToxicoSet] A ToxicoSet to be plotted in this graph. Currently only a single tSet is supported, passing more may results in errors.
duration	[character] A vector of durations to include in the plot.
cellline	[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. 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. Please note that using too many features will have a significant computational cost and will likely result in a over crowded 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 which must be "Control".
drug	[character] A vector of drugs to be included in this plot. In this release, only one drug is supported.
plot.type	[character] The type of plot which you would like returned. Options are 'Actual' for unfitted curve, 'Fitted' for the fitted curve and 'Both' to display 'Actual and 'Fitted' in the sample plot. Currently this function only supports 'Actual'.
summarize.replicates	[logical] If true will take the average of all replicates at each time point per gene and duration. This release has not yet implemented this feature.
xlim	[numeric] A vector of minimum and maximum values for the x-axis of the returned plot.
ylim	[numeric] A vector of minimum and miximum values for the y-axis of the returned plot.

mycol	[vector] A vector of length equal to the lenth of the tSets argument specifying which RColorBrewer colour to use per tSet. Default colours will be used if this parameter is excluded.
x.custom.ticks	[vector] A numeric vector of the distance between major and minor ticks on the x-axis. If excluded ticks appear only where duration values are specified.
title	[character] A string containing the desired plot name. If excluded a title wil be generated automatically.
lwd	[numeric] The line width to plot width
cex	[numeric] The cex parameter passed to plot. Controls the size of plot points and the font size of the legend and defaults to 0.7.
cex.main	[numeric] The cex.main parameter passed to plot, controls the size of the titles and defaults to 1.0.
legend.loc	[character] The location of the legend as passed to the plot() function from base graphics. Suggested values are either "topleft" or "topright", with the latter as the default.
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 = "naphthyl isothiocyanate",
  duration = c("2", "8", "24"), features = "ENSG00000000003_at")
}
```

---

drugInfo *drugInfo Generic*

Description

The generic for drugInfo method

Usage

```
drugInfo(tSet)
```

Arguments

tSet	A ToxicoSet object

Value

a data.frame with the drug annotations

Examples

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

---

drugInfo<- *drugInfo<- Generic*

Description

Generic for drugInfo replace method

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(tSet)
```

Arguments

tSet	A ToxicoSet object

Value

A vector of the drug names used in the ToxicoSet

Examples

```
data(TGGATESsmall)
drugName <- drugNames(TGGATESsmall)[1:10]
```

drugNames<- *drugNames<- Generic*

Description

A generic for the drugNames replacement method

Usage

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

Arguments

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

Value

Updated ToxicoSet

Examples

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

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

Description

Given a Toxicoset of the perturbation experiment type, and a list 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.

Usage

```
drugPerturbationSig(tSet, mDataType, drugs, cells, features, duration,
  dose, 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.
cells	[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)

Value

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

Examples

```
#data(TGGATES_small)
#drug.perturbation <- drugPerturbationSig(TGGATES_small, mDataType="rna", nthread=1)
#print(drug.perturbation)
```

---

drugResponseCurve *Plot drug response curve of a given drug and a given cell for a list of tSets (objects of the ToxicoSet class).*

Description

Given a list of ToxicoSets, 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 concentrations. If more than one tSet is provided, a light gray area would show the common concentration range between tSets. 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 concentrations and viability values, as in the examples below, and it will be treated as experiments equivalent to values coming from a tSet. The names of the concentration list determine the legend labels.

Usage

```
drugResponseCurve(drug = NULL, cellline = NULL, durations = NULL,
  tSets = NULL, concentrations = NULL, viabilities = NULL,
  conc_as_log = FALSE, viability_as_pct = TRUE, trunc = TRUE,
  legends.label = c("ic50_published", "gi50_published", "auc_published",
  "auc_recomputed", "ic50_recomputed"), ylim = c(0, 100), xlim, mycol,
  title, plot.type = c("Fitted", "Actual", "Both"),
  summarize.replicates = TRUE, lwd = 0.5, cex = 0.5,
  cex.main = 0.9, legend.loc = "topright", verbose = TRUE)
```

Arguments

drug	[string] A drug name for which the drug response curve should be plotted. If the plot is desirable for more than one toxico set, A unique drug id should be provided.
cellline	[string] A cell line name for which the drug response curve should be plotted. If the plot is desirable for more than one toxico set, A unique cell id should be provided.
durations	[numeric] A duration for which the drug response curve should be plotted.
tSets	[list] a list of ToxicoSet objects, for which the function should plot the curves.
concentrations, viabilities	[list] A list of concentrations and viabilities to plot, the function assumes that concentrations[[i]] is plotted against viabilities[[i]]. The names of the concentration list are used to create the legend labels

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. Applies only to the concentrations parameter.
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. Applies only to the viabilities parameter.
trunc	[bool] Should the viability values be truncated to lie in [0-100] before doing the fitting
legends.label	[vector] A vector of sensitivity measurment types. A legend will be displayed on the top right of the plot which each line of the legend is the values of requested sensitivity measurements for one of the requested tSets. If this parameter is missed no legend would be provided for the plot.
ylim	[vector] A vector of two numerical values to be used as ylim of the plot. If this parameter would be missed c(0,100) would be used as the ylim of the plot.
xlim	[vector] A vector of two numerical values to be used as xlim of the plot. If this parameter would be missed the minimum and maximum comncentrations between all the tSets would be used as plot xlim.
mycol	[vector] A vector with the same length of the tSets parameter which will determine the color of the curve for the toxico 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	[boolean] Should warning messages about the data passed in be printed?

Value

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

Examples

```
if (interactive()) {
drugResponseCurve(concentrations=list("Experiment 1"=c(.008, .04, .2, 1)),
 viabilities=list(c(100,50,30,1)), plot.type="Both")
}
```

drugTimeResponseCurve *Compares viabilities at a given dose over different experimental duration*

Description

Description of this function

Usage

```
drugTimeResponseCurve(tSets, duration, cellline, dose, drug,
  plot.type = "Actual", summarize.replicates = TRUE,
  viability_as_pct = TRUE, xlim = c(0, 24), ylim = c(0, 100), mycol,
  x.custom.ticks = NULL, title, lwd = 1, cex = 0.5, cex.main = 0.9,
  legend.loc = "topleft", verbose = TRUE)
```

Arguments

tSets	[ToxicoSet] A ToxicoSet or list of ToxicoSets to be plotted in this graph.
duration	[character] A vector of durations to include in the plot.
cellline	[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".
drug	[character] A vector of drugs to be included in this plot.
plot.type	[character] The type of plot which you would like returned. Options are 'Actual' for unfitted curve, 'Fitted' for the fitted curve and 'Both' to display 'Actual and 'Fitted' in the sample plot.
summarize.replicates	[logical] If true will take the average of all replicates at each time point per gene and duration. This release has not yet implemented this feature.
viability_as_pct	[logical] A vector specifying if viabilities should be plotted as a percentage. Defaults to TRUE.
xlim	[numeric] A vector of minimum and maximum values for the x-axis of the returned plot.
ylim	[numeric] A vector of minimum and miximum values for the y-axis of the returned plot.
mycol	[vector] A vector of length equal to the lenth of the tSets argument specifying which RColorBrewer colour to use per tSet. Default colours will be used if this parameter is excluded.
x.custom.ticks	[vector] A numeric vector of the distance between major and minor ticks on the x-axis. If excluded ticks appear only where duration values are specified.

title	[character] A string containing the desired plot name. If excluded a title wil be generated automatically.
lwd	[numeric] The line width to plot width
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	[character] The location of the legend as passed to the legends function from base graphics. Recommended values are 'topright' or 'topleft'. Default is 'topleft'.
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()) {
  ToxicoGx::drugTimeResponseCurve(TGGATESsmall, cellline = "Hepatocyte",
    dose = c("Control", "Low", "Middle"),
    drug = "naphthyl isothiocyanate", duration = c("2", "8", "24"))
}
```

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 molecular data names for the ToxicoSet.

Usage

```
## S4 method for signature 'ToxicoSet'
mDataNames(cSet = tSet)
```

Arguments

cSet	The CoreSet to retrieve cell info from

Value

Vector of names of the molecular data types

Examples

```
mDataNames(TGGATESsmall)
```

`names,ToxicoSet-method`
tSet Name

Description

Retrieves the name of a tSet

Usage

```
## S4 method for signature 'ToxicoSet'
names(x = tSet)
```

Arguments

`x`	[param] The named parameter from the base R names function. For internal use only.
`tSet`	[ToxicoSet] A ToxcioSet object

Value

character A string of the tSet's name

Examples

```
names(TGGATESsmall)
```

pharmacoSettoToxicoSet *Coerce pSet to tSet*

Description

Forces a PharmacoSet class objects to be a ToxicoSet object.

Usage

```
pharmacoSettoToxicoSet(pSet)
```

Arguments

pSet	A PharmacoSet class object.

Value

A ToxicoSet class object containing all data from the pSet.

show,ToxicoSet-method *Show a ToxicoSet*

Description

Show a ToxicoSet

Usage

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

Arguments

object	A ToxicoSet object.

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(tSet, cells = NULL, drugs = NULL,
  molecular.data.cells = NULL, duration = NULL, features = NULL, ...)
```

Arguments

tSet	A ToxicoSet 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.
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
features	A list or vector of feature names as used int he dataset from which the object will be subsetted. If left blank that all features will be left in.
...	Other arguments passed by other function 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, drugs, features,
  duration, 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"), 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,
  sensitivity.measure = "auc_recomputed", cell.lines, drugs, duration,
  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
sensitivity.measure	[character] which sensitivity sensitivity.measure to use? Use the sensitivityMeasures function to find out what measures are available for each TSet.
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
duration	numeric The duration at which to summarize the drug-cell combo.
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 Toxicogenomic Data*

Description

A description which has yet to be added to this class. This is just a place holder.

A generic for cellInfo method

Generic for cellInfo replace method

Generic for phenoInfo method

Generic for phenoInfo replace method

Generic for molecularProfiles method

Generic for molecularProfiles replace method

Generic for featureInfo replace method

Generic for sensitivityInfo method

A generic for the sensitivityInfo replacement method

Generic for sensitivityProfiles method

A generic for the sensitivityProfiles replacement method

A generic for the sensitivityMeasures method

A generic for the cellNames method

A generic for the cellNames replacement method

A generic for the fNames method

A generic for the dateCreated method

A generic for the tSetName method

A generic for the pertNumber method

A generic for the sensNumber method

A generic for the pertNumber method

A generic for the sensNumber method

Usage

```
## S4 method for signature 'ToxicoSet'
cellInfo(cSet = tSet)

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

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

```
drugInfo(tSet)

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

## S4 method for signature 'ToxicoSet,character'
phenoInfo(cSet = tSet, mDataType)

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

## S4 method for signature 'ToxicoSet,character'
molecularProfiles(cSet = tSet, mDataType)

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

## S4 method for signature 'ToxicoSet,character'
featureInfo(cSet = tSet, mDataType)

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

## S4 method for signature 'ToxicoSet'
sensitivityInfo(cSet = tSet)

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

## S4 method for signature 'ToxicoSet'
sensitivityProfiles(cSet = tSet)

## 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'
sensitivityMeasures(cSet = tSet)

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

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

```
drugNames(object) <- value

## S4 method for signature 'ToxicoSet'
cellNames(cSet = tSet)

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

## S4 method for signature 'ToxicoSet,character'
fNames(cSet = tSet, mDataType)

## S4 method for signature 'ToxicoSet'
dateCreated(cSet = tSet)

## S4 method for signature 'ToxicoSet'
cSetName(cSet = tSet)

## S4 method for signature 'ToxicoSet'
pertNumber(cSet = tSet)

## S4 method for signature 'ToxicoSet'
sensNumber(cSet = tSet)

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

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

Arguments

cSet	Parameter name for parent method inherited from CoreGx
object	A ToxicoSet object
value	A data.frame of replacement values
tSet	A ToxicoSet object
mDataType	A character with the type of molecular data to return/update

Value

An object of the ToxicoSet class

a data.frame with the cell annotations

Updated ToxicoSet

a data.frame with the experiment info

The updated ToxicoSet

Updated ToxicoSet

Updated ToxicoSet

a data.frame with the experiment info

Updated ToxicoSet setGeneric("sensitivityInfo<-", function(object, value) standardGeneric("sensitivityInfo<-"))

a data.frame with the experiment info setGeneric("sensitivityProfiles", function(tSet) standardGeneric("sensitivityProfiles"))

Updated ToxicoSet setGeneric("sensitivityProfiles<-", function(object, value) standardGeneric("sensitivityProfiles<-"))

A character vector of all the available sensitivity measures

A vector of the cell names used in the ToxicoSet

Updated ToxicoSet

A character vector of the feature names

The date the ToxicoSet was created

The name of the ToxicoSet

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

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

The updated ToxicoSet

The updated ToxicoSet

Methods (by generic)

- cellInfo: Returns the annotations for all the cell lines tested on in the ToxicoSet
- cellInfo<-: Update the cell line annotations
- drugInfo: Returns the annotations for all the drugs tested in the ToxicoSet
- drugInfo<-: Update the drug annotations
- 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
- molecularProfiles: Return the given type of molecular data from the ToxicoSet
- molecularProfiles<-: Update the given type of molecular data from the ToxicoSet
- featureInfo: Return the feature info for the given molecular data
- featureInfo<-: Replace the gene info for the molecular data
- sensitivityInfo: Return the drug dose sensitivity experiment info
- sensitivityInfo<-: Update the sensitivity experiment info
- 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
- sensitivityMeasures: Returns the available sensitivity profile summaries, for example, whether there are IC50 values available
- drugNames: Return the names of the drugs used in the ToxicoSet

- drugNames<-: Update the drug names used in the dataset
- cellNames: Return the cell names used in the dataset
- cellNames<-: Update the cell names used in the dataset
- fNames: Return the feature names used in the dataset
- dateCreated: Return the date the ToxicoSet was created
- cSetName: Return the name of the ToxicoSet
- pertNumber: Return the summary of available perturbation experiments
- sensNumber: Return the summary of available sensitivity experiments
- pertNumber<-: Update the summary of available perturbation experiments
- sensNumber<-: Update the summary of available sensitivity experiments

Slots

drug A data.frame containg the annotations for all the drugs profiled in the in the dataset, across all data types

Examples

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

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

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

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

data(TGGATESsmall)
TGGATES_mProf <- molecularProfiles(TGGATESsmall, "rna")[1:10,]

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

data(TGGATESsmall)
featureInfo <- featureInfo(TGGATESsmall, "rna")[1:10,]

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

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

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

data(TGGATESsmall)
sensProf <- sensitivityProfiles(TGGATESsmall)
```

```
sensitivityProfiles(TGGATESsmall) <- sensitivityProfiles(TGGATESsmall)

sensitivityMeasures(TGGATESsmall)

cellNames(TGGATESsmall)

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

fNames(TGGATESsmall, "rna")[1:10]

dateCreated(TGGATESsmall)

tSetName <- cSetName
tSetName(TGGATESsmall)

pertNumber(TGGATESsmall)

sensNumber(TGGATESsmall)

pertNumber(TGGATESsmall) <- pertNumber(TGGATESsmall)

sensNumber(TGGATESsmall) <- sensNumber(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)[1:3]]
```

Index