

# Package ‘ToxicoGx’

September 30, 2024

**Type** Package

**Title** Analysis of Large-Scale Toxico-Genomic Data

**Version** 2.8.0

**Date** 2023-03-08

**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 (>= 4.1), CoreGx

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

**RoxygenNote** 7.2.3

**Roxygen** list(markdown=TRUE, r6=FALSE)

**VignetteBuilder** knitr

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

**biocViews** GeneExpression, Pharmacogenetics, Pharmacogenomics, Software

**Collate** 'LogLogisticRegression.R' 'SanityCheck.R' 'allGenerics.R' 'ToxicoSet-class.R' 'ToxicoSet-accessors.R' 'class-signature.R' 'columnsAsCharacter.R' 'computeAUC.R' 'computeDrugSensitivity.R' 'computeIC50.R' 'computeICn.R' 'dataset.R' 'downloadTSet.R' 'drugGeneResponseCurve.R' 'drugPerturbationSig.R' 'drugTimeResponseCurve.R' 'geneDrugPerturbation.R' 'geneDrugSensitivity.R' 'globals.R' 'methods-computeLimmaDiffExpr.R' 'methods-subsetTo.R'

'paramErrorChecker.R' 'paramMissingHandler.R'  
 'paramMissingHandlerTests.R' 'rankGeneDrugPerturbation.R'  
 'rankGeneDrugSensitivity.R' 'summarizeMolecularProfiles.R'  
 'summarizeSensitivityProfiles.R' 'updateObject-methods.R'  
 'utilities.R' 'zzz.R'

**git\_url** <https://git.bioconductor.org/packages/ToxicoGx>

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 6018f21

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-09-29

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

**Maintainer** Benjamin Haibe-Kains <[benjamin.haibe.kains@utoronto.ca](mailto:benjamin.haibe.kains@utoronto.ca)>

## Contents

availableTsets . . . . .	3
checkTsetStructure . . . . .	4
computeAUC . . . . .	4
computeIC50 . . . . .	6
computeLimmaDiffExpr . . . . .	7
computeLimmaDiffExpr,ToxicoSet-method . . . . .	8
dim,ToxicoSet-method . . . . .	8
downloadTset . . . . .	9
drugGeneResponseCurve . . . . .	10
drugPerturbationSig . . . . .	11
drugTimeResponseCurve . . . . .	13
geneDrugPerturbation . . . . .	15
HCC_sig . . . . .	16
logLogisticRegression . . . . .	16
show,ToxicoSet-method . . . . .	18
show,ToxicoSig-method . . . . .	18
showSigAnnot . . . . .	19
subsetTo . . . . .	20
summarizeMolecularProfiles . . . . .	21
summarizeSensitivityProfiles . . . . .	22
TGGATESsmall . . . . .	23
ToxicoSet . . . . .	24

<i>availableTsets</i>	3
ToxicoSet-accessors . . . . .	25
ToxicoSet-class . . . . .	34
ToxicoSig . . . . .	35
updateObject,ToxicoSet-method . . . . .	36
[,ToxicoSet,ANY,ANY,ANY-method . . . . .	37
<b>Index</b>	<b>38</b>

---

<code>availableTsets</code>	<i>Return a table of ToxicoSets available for download</i>
-----------------------------	--

---

### Description

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

### Usage

```
availableTsets(canonical = TRUE)
```

### Arguments

`canonical`      **logical** Should available Tsets show only official Tsets, or should user generated Tsets be included?

### Details

Much more information on the processing of the data and data provenance can be found at: [www.orchestra.ca](http://www.orchestra.ca)

### Value

A data.frame with details about the available ToxicoSet objects

### Examples

```
if (interactive()){
  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-Viability) 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 IC<sub>n</sub> for any n in 0-100 for a Drug Dose Viability Curve*


---

### Description

Returns the IC<sub>n</sub> 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 IC<sub>n</sub> 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(IC <sub>n</sub> ) should be returned instead of IC <sub>n</sub> .

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 percentile 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` *Generic method for performing differential expression analysis on an S4 object using the limma package*

---

**Description**

Generic method for performing differential expression analysis on an S4 object using the limma package

**Usage**

```
computeLimmaDiffExpr(object, ...)
```

**Arguments**

`object` [S4](#) An S4 object to conduct differential expression analysis on.  
`...` Allow new parameters to be added to this generic.

**Value**

To be defined by the method implementation.

---

computeLimmaDiffExpr,ToxicoSet-method

*Conduct differential expression analysis using the limma R package*

---

### 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 <a href="#">ToxicoSet</a> object with a molecular profile named 'rna'
buildTable	<a href="#">logical</a> 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 will 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)  
}
```

---

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	<i>Download a ToxicoSet object</i>
--------------	------------------------------------

---

**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 ToxicoSet class, enabling use with all the analysis tools provided in ToxicoGx.

**Usage**

```
downloadTSet(
  name,
  saveDir = tempdir(),
  tSetFileName = NULL,
  verbose = TRUE,
  timeout = 600
)
```

**Arguments**

name	Character string, the name of the PhamracoSet to download.
saveDir	Character string with the folder path where the 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.
timeout	numeric(1) How long to wait before the download times out, in seconds. Default is 600 seconds (10 minutes).

**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 specified 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 <code>treatmentNames(tSet)</code> 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 <code>size</code> in <code>geom_line()</code> . Defaults to 1.
point_size	numeric A number specifying how large points should be in the plot, as passed to <code>size</code> in <code>geom_point()</code> . Defaults to 2.5.
ggplot_args	list A list of ggplot2 functions which can be called using the <code>plot + function()</code> 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 = treatmentNames(TGGATESsmall)[1],
    duration = c("2", "8", "24"), features = "ENSG00000002726_at")
}
```

---

drugPerturbationSig     *Drug perturbation analysis*

---

**Description**

Creates a signature representing gene expression (or other molecular profile) change induced by administering 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 include 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, "rna")
}

```

---

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 which 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.

<code>point_size</code>	numeric A number specifying how large points should be in the plot, as passed to <code>size</code> in <code>geom_point()</code> . Defaults to 2.5.
<code>verbose</code>	boolean Should warning messages about the data passed in be printed?
<code>ggplot_args</code>	list A list of <code>ggplot2</code> functions which can be called using the <code>plot + function()</code> 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 <code>ggplot2</code> documentation for advanced customization. Alternatively, you could assign the return value to a variable and add the customization yourself using <code>plot + function()</code> .

**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 = treatmentNames(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 = treatmentNames(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")
}
```

---

geneDrugPerturbation *Compute gene-drug associations*

---

## Description

Function computing gene-drug associations from perturbation data

## Usage

```
geneDrugPerturbation(x, concentration, type, batch, duration, model = FALSE)
```

## Arguments

**x** [numeric](#) Vector of gene expression values

**concentration** [numeric](#) Vector with drug concentrations/doses

**type** [factor](#) Vector of factors specifying the cell lines or type types

**batch** [factor](#) Vector of factors specifying the batch

**duration** [character](#) Vector of measurement times (in hours)

**model** [logical](#) Should the full linear model be returned? Default set to FALSE

## Value

[numeric](#) Vector reporting the effect size (estimate of the coefficient of drug concentration), standard error (se), sample size (n), t statistic, and F statistics and its corresponding p-value

## Examples

```
ToxicoGx::drugPerturbationSig(tSet = TGGATESsmall,  
  mDataType="rna",  
  cell_lines="Hepatocyte",  
  duration="24",  
  dose=c("Control", "Low"),  
  drugs=c("Omeprazole", "Isoniazid"),  
  returnValues=c("estimate", "tstat", "pvalue", "fdr"),  
  verbose=FALSE)
```

---

HCC_sig	<i>HCC_sig dataset</i>
---------	------------------------

---

**Description**

A dataset containing 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	<b>vector</b> is a vector of drug concentrations.
viability	<b>vector</b> 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 <code>log_conc</code> , where viability 0 indicates that all cells died, and viability 1 indicates that the drug had no effect on the cells.
density	<b>vector</b> is a vector of length 3 whose components are the numbers of lattice points per unit length along the HS-, $E_{\infty}$ -, and base-10 logarithm of the EC50-dimensions of the parameter space, respectively.
step	<b>vector</b> is a vector of length 3 whose entries are the initial step sizes in the HS, $E_{\infty}$ , 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	<b>vector</b> is a vector of length 3 whose entries are the lower bounds on the HS, $E_{\infty}$ , and base-10 logarithm of the EC50 parameters, respectively.
upper_bounds	<b>vector</b> is a vector of length 3 whose entries are the upper bounds on the HS, $E_{\infty}$ , and base-10 logarithm of the EC50 parameters, respectively.
scale	is a positive real number specifying the shape parameter of the Cauchy distribution.
family	<b>character</b> , 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	<b>logical</b> , 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	<b>logical</b> , if false, assumes that viability is given as a decimal rather than a percentage, and that $E_{\infty}$ should be returned as a decimal rather than a percentage.
trunc	<b>logical</b> , if true, causes viability data to be truncated to lie between 0 and 1 before curve-fitting is performed.
verbose	<b>logical</b> , if true, causes warnings thrown by the function to be printed.

**Value**

A vector containing estimates for HS,  $E_{\infty}$ , 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)
```

---

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 ToxicGx 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(treatmentNames(TGGATESsmall)), features = fName(TGGATESsmall, "rna")[seq_len(2)])
drug.perturbation
```

---

showSigAnnot

*Show the Annotations of a signature object*

---

**Description**

This function 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 ToxicSig Class, as returned by drugPerturbationSig

**Value**

Prints the ToxicGx 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(treatmentNames(TGGATESsmall)), features = fName(TGGATESsmall, "rna")[seq_len(2)])
showSigAnnot(drug.perturbation)
```

---

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

---

### Description

This is the preferred 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 <- treatmentNames(TGGATESsmall)
TGGATESCells <- sampleNames(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=sampleNames(TGGATESsmall), drugs = head(treatmentNames(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)[[treatmentNames(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 sensitivity 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	<i>ToxicoSet constructor</i>
-----------	------------------------------

---

**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(),
  sample = data.frame(),
  treatment = data.frame(),
  sensitivityInfo = data.frame(),
  sensitivityRaw = array(dim = c(0, 0, 0)),
  sensitivityProfiles = matrix(),
  sensitivityN = matrix(nrow = 0, ncol = 0),
  perturbationN = array(NA, dim = c(0, 0, 0)),
  curationTreatment = data.frame(),
  curationSample = 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 SummarizedExperiment objects containing molecular profiles for each molecular data type.



sample	A data.frame containing the annotations for all the sample profiled in the data set, across all data types. Must contain the mandatory <code>sampleid</code> column which uniquely identifies each sample in the object.
treatment	A data.frame containing annotations for all treatments profiled in the dataset. Must contain the mandatory <code>treatmentid</code> column which uniquely identifies each treatment in the object.
sensitivityInfo	A data.frame containing the information for the sensitivity experiments. Must contain a <code>'sampleid'</code> column with unique identifiers to each sample, matching the sample object and a <code>'treatmentid'</code> columns with unique identifiers for each treatment, matching the treatment object.
sensitivityRaw	A 3 Dimensional array containing the raw drug dose response data for the sensitivity experiments
sensitivityProfiles	data.frame containing drug sensitivity profile statistics such as IC50 and AUC
sensitivityN, perturbationN	A data.frame summarizing the available sensitivity/perturbation data
curationSample, curationTissue, curationTreatment	A data.frame mapping the names for samples, tissues and treatments used in the data set to universal identifiers used between different CoreSet objects
datasetType	A character(1) string of <code>'sensitivity'</code> , <code>'perturbation'</code> , or <code>'both'</code> detailing what type of data can be found in the CoreSet, for proper processing of the data
verify	logical(1) Should the function verify the CoreSet and print out any errors it finds after construction?

**Value**

An object of class `ToxicoSet`

---

`ToxicoSet-accessors`    *Accessing and modifying information in a CoreSet*

---

**Description**

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

**Usage**

```
drugInfo(...)
```

```
drugInfo(...) <- value
```

```
drugNames(...)
```

```
drugNames(...) <- value

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

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

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

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

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

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

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

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

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

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

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

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

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

## S4 replacement method for signature 'ToxicoSet,character'
datasetType(object) <- 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 method for signature 'ToxicoSet'
featureInfo(object, mDataType)

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

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

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

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

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

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

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

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

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

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

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

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

```

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'
treatmentResponse(object)

## S4 replacement method for signature 'ToxicoSet,list_OR_LongTable'
treatmentResponse(object) <- value

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

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

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

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

```

## Arguments

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

## Details

**treatmentInfo:** data.frame Metadata for all treatments in a ToxicoSet object. Arguments:

- object: ToxicoSet An object to retrieve treatment metadata from.

**treatmentInfo<-:** ToxicoSet object with updated treatment metadata. object. Arguments:

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

**treatmentNames:** character Names for all treatments in a ToxicoSet object. Arguments:

- **object:** ToxicoSet An object to retrieve treatment names from.

**treatmentNames<-:** ToxicoSet Object with updates treatment names. **object.** Arguments:

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

**@annotation:**

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

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

- **value:** a list of annotations to update the ToxicoSet with.

**@dateCreated:**

**dateCreated:** character(1) The date the ToxicoSet object was created, as returned by the date() function.

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

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

**name:** character(1) The name of the ToxicoSet, retrieved from the @annotation slot.

**name<-:** Update the @annotation\$name value in a ToxicoSet object.

- **value:** character(1) The name of the ToxicoSet object.

**cellInfo:** data.frame Metadata for all sample in a ToxicoSet object.

**sampleInfo<-:** assign updated sample annotations to the ToxicoSet object. Arguments:

- **value:** a data.frame object.

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

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

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

**@curation:**

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

**curation<-:** Update the curation slot of a ToxicoSet object. Arguments:

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

**datasetType slot:**

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

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

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

**@molecularProfiles:**

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

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

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

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

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

**featureInfo<-:** Update the featureInfo(object, mDataType) DataFrame with new feature metadata. Arguments:

- value: A data.frame or DataFrame with updated feature metadata for the specified molecular profile in the molecularProfiles slot of a ToxicoSet object.

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

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

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

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

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

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

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

**mDataNames:** Update the molecular data type names of the molecularProfiles slot of a ToxicoSet object. Arguments:

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

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

**molecularProfilesSlot<-:** Update the contents of the @molecularProfiles slot of a ToxicoSet object. Arguments:

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

#### @treatmentResponse:

*Arguments::*

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

*Methods::*

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

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

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

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

**sensitivityMeasures:** Update the sensitivity measure in a ToxicoSet object. These values are the column names of the 'profiles' assay and represent various computed sensitivity metrics such as ic50, ec50, AUC, AAC, etc.

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

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

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

**sensitivityRaw:** Access the raw sensitivity measurements for a ToxicoSet object. A 3D array where rows are experiment\_ids, columns are doses and the third dimension is metric, either 'Dose' for the doses used or 'Viability' for the sample viability at that dose.

**sensitivityRaw<-:** Update the raw dose and viability data in a ToxicoSet.

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

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

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

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

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

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

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

## Value

Accessors: See details.

Setters: An updated CoreSet object, returned invisibly.



**Examples**

```
data(TGGATESsmall)
treatmentInfo(TGGATESsmall)

treatmentInfo(TGGATESsmall) <- treatmentInfo(TGGATESsmall)

treatmentNames(TGGATESsmall)

treatmentNames(TGGATESsmall) <- treatmentNames(TGGATESsmall)

## @annotation
annotation(TGGATESsmall)

annotation(TGGATESsmall) <- annotation(TGGATESsmall)

dateCreated(TGGATESsmall)

## dateCreated
dateCreated(TGGATESsmall) <- date()

name(TGGATESsmall)

name(TGGATESsmall) <- 'new_name'

sampleInfo(TGGATESsmall) <- sampleInfo(TGGATESsmall)

sampleNames(TGGATESsmall)

sampleNames(TGGATESsmall) <- sampleNames(TGGATESsmall)

## curation
curation(TGGATESsmall)

curation(TGGATESsmall) <- curation(TGGATESsmall)

datasetType(TGGATESsmall)

datasetType(TGGATESsmall) <- 'both'

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

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

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

featureInfo(TGGATESsmall, 'rna')
```

```
featureInfo(TGGATESSsmall, 'rna') <- featureInfo(TGGATESSsmall, 'rna')
phenoInfo(TGGATESSsmall, 'rna')
phenoInfo(TGGATESSsmall, 'rna') <- phenoInfo(TGGATESSsmall, 'rna')
fNames(TGGATESSsmall, 'rna')
fNames(TGGATESSsmall, 'rna') <- fNames(TGGATESSsmall, 'rna')
mDataNames(TGGATESSsmall)
mDataNames(TGGATESSsmall) <- mDataNames(TGGATESSsmall)
molecularProfilesSlot(TGGATESSsmall)
molecularProfilesSlot(TGGATESSsmall) <- molecularProfilesSlot(TGGATESSsmall)
sensitivityInfo(TGGATESSsmall)
sensitivityInfo(TGGATESSsmall) <- sensitivityInfo(TGGATESSsmall)
sensitivityMeasures(TGGATESSsmall) <- sensitivityMeasures(TGGATESSsmall)
sensitivityMeasures(TGGATESSsmall) <- sensitivityMeasures(TGGATESSsmall)
sensitivityProfiles(TGGATESSsmall)
sensitivityProfiles(TGGATESSsmall) <- sensitivityProfiles(TGGATESSsmall)
head(sensitivityRaw(TGGATESSsmall))
sensitivityRaw(TGGATESSsmall) <- sensitivityRaw(TGGATESSsmall)
treatmentResponse(TGGATESSsmall)
treatmentResponse(TGGATESSsmall) <- treatmentResponse(TGGATESSsmall)
sensNumber(TGGATESSsmall)
sensNumber(TGGATESSsmall) <- sensNumber(TGGATESSsmall)
pertNumber(TGGATESSsmall)
pertNumber(TGGATESSsmall) <- pertNumber(TGGATESSsmall)
```

**Description**

The ToxicoSet (tSet) class was developed 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 range of compounds. Features include plotting dose and exposure time curves, calculating AUC, fitting linear models and computing sensitivity signatures.

**Value**

An object of the ToxicoSet class

**Slots**

`annotation` A list of annotation data about the ToxicoSet, including the `$name` and the session information for how the object was created, 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

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

`treatment` A `data.frame` containing the annotations for all the drugs profiled in the data set, across all data types

`treatmentResponse` 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 containing `$n`, a `data.frame` summarizing the available perturbation data,

`curation` A list containing mappings for `$treatment`, `sample`, `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

---

ToxicoSig

*ToxicoSig Constructor*

---

**Description**

A user friendly constructor to create ToxicoSig class objects. This function is implemented as an internal and should only be called for development purposes

**Usage**

```

ToxicoSig(
  Data = array(NA, dim = c(0, 0, 0)),
  tSetName = "",
  DateCreated = date(),
  SigType = "sensitivity",
  SessionInfo = sessionInfo(),
  Call = "No Call Recorded",
  Arguments = list()
)

```

**Arguments**

Data	‘array‘ An array containing the data for constructing the ToxicoSig object
tSetName	character(1) The name of the tSet used in the constructor
DateCreated	date The data at time of running the constructor
SigType	character A string of the experiment type
SessionInfo	sessionInfo The current session info
Call	character(1) A string
Arguments	list A list of arguments passed to the constructor

**Value**

object A new ToxicoSig object

---

updateObject,ToxicoSet-method

*Update the ToxicoSet class after changes in it struture or API*

---

**Description**

Update the ToxicoSet class after changes in it struture or API

**Usage**

```

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

```

**Arguments**

object	A ToxicoSet object to update the class structure for.
--------	---

**Value**

ToxicoSet with update class structure.

---

[,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[sampleNames(TGGATESsmall), treatmentNames(TGGATESsmall)[seq_len(3)]]
```

# Index

- \* **datasets**
  - HCC\_sig, [16](#)
  - TGGATESsmall, [23](#)
- \* **internal**
  - geneDrugPerturbation, [15](#)
  - ToxicoSig, [35](#)
- .ToxicoSet (ToxicoSet-class), [34](#)
- [,ToxicoSet,ANY,ANY,ANY-method, [37](#)
  
- annotation (ToxicoSet-accessors), [25](#)
- annotation,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- annotation<- (ToxicoSet-accessors), [25](#)
- annotation<- ,ToxicoSet,list-method  
(ToxicoSet-accessors), [25](#)
- availableTSets, [3](#)
  
- cellInfo (ToxicoSet-accessors), [25](#)
- cellInfo,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- cellInfo<- (ToxicoSet-accessors), [25](#)
- cellInfo<- ,ToxicoSet,data.frame-method  
(ToxicoSet-accessors), [25](#)
- cellName,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- cellNames (ToxicoSet-accessors), [25](#)
- cellNames<- (ToxicoSet-accessors), [25](#)
- cellNames<- ,ToxicoSet,list-method  
(ToxicoSet-accessors), [25](#)
- character, [15](#), [17](#)
- checkTSetStructure, [4](#)
- computeAUC, [4](#)
- computeIC50, [6](#)
- computeICn (computeIC50), [6](#)
- computeLimmaDiffExpr, [7](#)
- computeLimmaDiffExpr,ToxicoSet-method,  
[8](#)
- curation (ToxicoSet-accessors), [25](#)
- curation,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
  
- curation<- (ToxicoSet-accessors), [25](#)
- curation<- ,ToxicoSet,list-method  
(ToxicoSet-accessors), [25](#)
  
- data.table, [8](#)
- datasetType (ToxicoSet-accessors), [25](#)
- datasetType,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- datasetType<- (ToxicoSet-accessors), [25](#)
- datasetType<- ,ToxicoSet,character-method  
(ToxicoSet-accessors), [25](#)
- dateCreated (ToxicoSet-accessors), [25](#)
- dateCreated,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- dateCreated<- (ToxicoSet-accessors), [25](#)
- dateCreated<- ,ToxicoSet,character-method  
(ToxicoSet-accessors), [25](#)
- dateCreated<- ,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- dim,ToxicoSet-method, [8](#)
- downloadTSet, [9](#)
- drugGeneResponseCurve, [10](#)
- drugInfo (ToxicoSet-accessors), [25](#)
- drugInfo<- (ToxicoSet-accessors), [25](#)
- drugNames (ToxicoSet-accessors), [25](#)
- drugNames<- (ToxicoSet-accessors), [25](#)
- drugPerturbationSig, [11](#)
- drugTimeResponseCurve, [13](#)
  
- factor, [15](#)
- featureInfo (ToxicoSet-accessors), [25](#)
- featureInfo,ToxicoSet-method  
(ToxicoSet-accessors), [25](#)
- featureInfo<- (ToxicoSet-accessors), [25](#)
- featureInfo<- ,ToxicoSet,character,data.frame-method  
(ToxicoSet-accessors), [25](#)
- featureInfo<- ,ToxicoSet,character,DataFrame-method  
(ToxicoSet-accessors), [25](#)
- fNames (ToxicoSet-accessors), [25](#)

- fNames, ToxicoSet, character-method  
 (ToxicoSet-accessors), 25  
 fNames<- (ToxicoSet-accessors), 25  
 fNames<- , ToxicoSet, character, character-method  
 (ToxicoSet-accessors), 25  
 geneDrugPerturbation, 15  
 HCC\_sig, 16  
 logical, 3, 8, 15, 17  
 logLogisticRegression, 16  
 mDataNames (ToxicoSet-accessors), 25  
 mDataNames, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 mDataNames<- (ToxicoSet-accessors), 25  
 mDataNames<- , ToxicoSet, ANY-method  
 (ToxicoSet-accessors), 25  
 mDataNames<- , ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 molecularProfiles  
 (ToxicoSet-accessors), 25  
 molecularProfiles, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 molecularProfiles<-  
 (ToxicoSet-accessors), 25  
 molecularProfiles<- , ToxicoSet, character, character-method  
 (ToxicoSet-accessors), 25  
 molecularProfiles<- , ToxicoSet, character, missing, matrix-method  
 (ToxicoSet-accessors), 25  
 molecularProfiles<- , ToxicoSet, missing, missing, matrix-method  
 (ToxicoSet-accessors), 25  
 molecularProfiles<- , ToxicoSet, missing, missing, multiAssayExperiment-method  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot<-  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot<- , ToxicoSet, list-method  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot<- , ToxicoSet, list\_OR\_MAEs-method  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot<- ToxicoSet, MultiAssayExperiment-method  
 (ToxicoSet-accessors), 25  
 molecularProfilesSlot, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 name (ToxicoSet-accessors), 25  
 name, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 name<- (ToxicoSet-accessors), 25  
 name<- , ToxicoSet, character-method  
 (ToxicoSet-accessors), 25  
 name<- , ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 numeric, 15  
 pertNumber (ToxicoSet-accessors), 25  
 pertNumber, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 pertNumber<- (ToxicoSet-accessors), 25  
 pertNumber<- , ToxicoSet, array-method  
 (ToxicoSet-accessors), 25  
 phenoInfo (ToxicoSet-accessors), 25  
 phenoInfo, ToxicoSet, character-method  
 (ToxicoSet-accessors), 25  
 phenoInfo<- (ToxicoSet-accessors), 25  
 phenoInfo<- , ToxicoSet, character, data.frame-method  
 (ToxicoSet-accessors), 25  
 phenoInfo<- , ToxicoSet, character, DataFrame-method  
 (ToxicoSet-accessors), 25  
 S4, 7  
 sampleInfo (ToxicoSet-accessors), 25  
 sampleInfo, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 sampleInfo<- (ToxicoSet-accessors), 25  
 sampleInfo<- , ToxicoSet, data.frame-method  
 (ToxicoSet-accessors), 25  
 sampleInfo<- , ToxicoSet, multiAssayExperiment-method  
 (ToxicoSet-accessors), 25  
 sampleNames (ToxicoSet-accessors), 25  
 sampleNames, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 sampleNames<- (ToxicoSet-accessors), 25  
 sampleNames<- , ToxicoSet, character-method  
 (ToxicoSet-accessors), 25  
 sampleNames<- , ToxicoSet, list-method  
 (ToxicoSet-accessors), 25  
 sampleNames<- , ToxicoSet, list\_OR\_MAEs-method  
 (ToxicoSet-accessors), 25  
 sampleNames<- ToxicoSet, MultiAssayExperiment-method  
 (ToxicoSet-accessors), 25  
 sampleNames, ToxicoSet-method  
 (ToxicoSet-accessors), 25  
 sensitivityInfo, ToxicoSet, character-method  
 (ToxicoSet-accessors), 25  
 sensitivityInfo, ToxicoSet, missing-method  
 (ToxicoSet-accessors), 25  
 sensitivityInfo, ToxicoSet-method  
 (ToxicoSet-accessors), 25

- sensitivityInfo<- ,ToxicoSet,data.frame-method [treatmentNames \(ToxicoSet-accessors\), 25](#)  
[\(ToxicoSet-accessors\), 25](#)      [treatmentNames,ToxicoSet-method](#)
- sensitivityInfo<- ,ToxicoSet,missing,data.frame-method [\(ToxicoSet-accessors\), 25](#)  
[\(ToxicoSet-accessors\), 25](#)      [treatmentNames<- \(ToxicoSet-accessors\),](#)  
[25](#)
- sensitivityMeasures,ToxicoSet-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentNames<- ,ToxicoSet,character-method](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivityMeasures<- ,ToxicoSet,character-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentResponse](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivityProfiles,ToxicoSet-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentResponse,ToxicoSet-method](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivityProfiles<- ,ToxicoSet,data.frame-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentResponse<-](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivityRaw,ToxicoSet-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentResponse<- ,ToxicoSet,list\\_OR\\_LongTable-method](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivityRaw<- ,ToxicoSet,array-method  
[\(ToxicoSet-accessors\), 25](#)      [treatmentResponse<- ,ToxicoSet,LongTable-method](#)  
[\(ToxicoSet-accessors\), 25](#)
- sensitivitySlot (ToxicoSet-accessors),  
[25](#)      [updateObject,ToxicoSet-method, 36](#)
- sensitivitySlot<-  
[\(ToxicoSet-accessors\), 25](#)      [vector, 17](#)
- sensitivityInfo<- ,ToxicoSet,character,data.frame-method  
[\(ToxicoSet-accessors\), 25](#)
- sensNumber (ToxicoSet-accessors), [25](#)
- sensNumber ,ToxicoSet-method  
[\(ToxicoSet-accessors\), 25](#)
- sensNumber<- (ToxicoSet-accessors), [25](#)
- sensNumber<- ,ToxicoSet,matrix-method  
[\(ToxicoSet-accessors\), 25](#)
- show,ToxicoSet-method, [18](#)
- show,ToxicoSig-method, [18](#)
- showSigAnnot, [19](#)
- subsetTo, [20](#)
- summarizeMolecularProfiles, [21](#)
- summarizeSensitivityProfiles, [22](#)
  
- TGGATESsmall, [23](#)
- ToxicoSet, [8, 24](#)
- ToxicoSet-accessors, [25](#)
- ToxicoSet-class, [34](#)
- ToxicoSig, [35](#)
- treatmentResponse<- ,ToxicoSet,list-method  
[\(ToxicoSet-accessors\), 25](#)
- treatmentInfo (ToxicoSet-accessors), [25](#)
- treatmentInfo,ToxicoSet-method  
[\(ToxicoSet-accessors\), 25](#)
- treatmentInfo<- (ToxicoSet-accessors),  
[25](#)
- treatmentInfo<- ,ToxicoSet,data.frame-method  
[\(ToxicoSet-accessors\), 25](#)