

Package ‘mixlow’

January 2, 2012

Version 1.0.0

Date 2010-04-16

Author John Boik <john.boik@newearthbiomed.org>

Maintainer John Boik <john.boik@newearthbiomed.org>

Depends R (>= 2.5), nlme (>= 3.1.4)

Description mixlow software for assessing drug synergism/antagonism

License GPL (>= 2)

Title Mixlow Method for Assessing Drug Synergism/Antagonism

Repository CRAN

Date/Publication 2010-04-17 05:13:22

R topics documented:

mixlow-package	2
doLoewe	3
doNlme	5
doNls	8
getCellLines	9
getDrugs	10
getNlsParameterDefaults	11
getTrays	12
loeweData	13
mixlowData	14
nlmeData	15
NlmePrintVarFunctions	15
nlsData	16
plot.loeweData	17
plot.mixlowData	18
plot.nlmeData	19

plot.nlsData	20
prepareData	21
print.loeweData	22
print.mixlowData	23
print.nlmeData	24
print.nlsData	25
readDataFile	26
summary.loeweData	27
summary.mixlowData	28
summary.nlmeData	29
summary.nlsData	30
summary.trayData	31
trayData	32

Index 33

mixlow-package	<i>Calculate synergism/antagonism indices based on concentration- response curve parameters</i>
----------------	---

Description

Uses a mixed-effects model to calculate parameters of concentration- response curves. These parameters are used in calculating Loewe interaction indices to quantify drug synergism/antagonism.

Details

Package:	mixlow
Type:	Package
Version:	0.02
Date:	2010-4-8
License:	GPL-2

Functions are generally called in this order: readDataFile, prepareData, doNls, doNlme, and doLoewe. Plot functions are available to view results graphically.

Author(s)

John Boik <john.boik@newearthbiomed.org>

References

- Boik J.C., Newman R.A., Boik R.J. (2008). Quantifying Synergism/Antagonism Using Nonlinear Mixed-Effects Modeling: A Simulation Study. *Statistics in Medicine*, **27**(7), 1040–1061
- Boik J.C., Narasimhan N. (2010). An R Package for Assessing Drug Synergism/Antagonism. *Journal of Statistical Software*, **34**(6), 1–18.

Examples

```
# trayData data object is obtained using the readDataFile function
data(trayData)
mixlowData <- prepareData(trayData)
trays <- getTrays(trayData)
parameterDefaults <- getNlsParameterDefaults(trays[1:9])
parameterDefaults["vin_tr1", "param.lambda"] = .2
parameterDefaults["vin_tr2", "param.lambda"] = .2
parameterDefaults["vin_tr3", "param.lambda"] = .2
nlsData = doNls(mixlowData, parameterDefaults)
nlmeData = doNlme(mixlowData, nlsData)
loeweData = doLoewe(mixlowData, nlmeData)
```

doLoewe

Estimate Loewe indices and confidence intervals

Description

Uses parameter and covariance estimates produced by doNlme to estimate Loewe indices and confidence intervals at different fraction affected values.

Usage

```
doLoewe(mixlowData, nlmeData, verbose)
```

Arguments

mixlowData	A list obtained from the output of prepareData
nlmeData	A list obtained from the output of doNlme
verbose	An optional logical value. If TRUE, information from intermediate steps in the Loewe analysis is printed. Default is FALSE.

Details

Loewe indices are estimated based on parameter and covariance values generated by use of the doNlme function. In addition to returning interaction indices, the doLoewe function returns a vector indicating the degree of statistically significant synergism at each fraction affected value. The degree of statistically significant synergism at each fraction affected value is zero if the upper confidence limit is above 1.0, and is the difference between the upper confidence limit and 1.0 otherwise. A vector of statistically significant antagonism is calculated in an analogous manner. These vectors summarize the degree of synergism/antagonism occurring over a range of fraction affected values.

The width of the index confidence intervals is dependent in part on the critical value taken from the t-distribution, which in turn is dependent on the degrees of freedom. For multiple drug analysis with doNlme, the degrees of freedom is taken directly from the nlme results. For single drug analysis, the degrees of freedom is taken as the minimum degrees of freedom over all drugs.

Value

A list with the following named components:

drugs	The drugs analyzed
mix	The mixture analyzed
ciL	A data frame containing Loewe index estimates and confidence intervals based on the log of the index, for different fraction affected values
covv	Covariance matrix of the parameter estimates
g0	gamma, shape parameters
p0	psi, IC50 parameters
uu	u parameter
dat0	Data frame containing concentration-response values
score.interval	Interval of fraction affected values over which synergism/antagonism scores were computed
syner	Synergism scores for fraction affected values in interval
syner.total	Sum of synergism scores over interval
antag	Antagonism scores for fraction affected values in interval
antag.total	Sum of antagonism scores over interval

See doNlme for a description of gamma, psi, and u parameters.

Author(s)

John Boik <john.boik@newearthbiomed.org>

References

- Boik J.C., Newman R.A., Boik R.J. (2008). Quantifying Synergism/Antagonism Using Nonlinear Mixed-Effects Modeling: A Simulation Study. *Statistics in Medicine*, **27**(7), 1040–1061
- Boik J.C., Narasimhan N. (2010). An R Package for Assessing Drug Synergism/Antagonism. *Journal of Statistical Software*, **34**(6), 1–18.

See Also

[doNlme](#), [plot.loeweData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
# nlmeData data object is obtained using the doNlme function
data(nlmeData)
loeweData <- doLoewe(mixlowData, nlmeData)
```

doNlme	<i>Conduct nonlinear mixed-effects analysis of sigmoidal concentration-response data</i>
--------	--

Description

Organizes input and calls the `nlme` function to obtain parameter estimates for sigmoidal concentration-response curves. Starting values for the fixed effects are obtained from the output of the `doNls` function.

Usage

```
doNlme(mixlowData, nlsData, drugs=getDrugs(mixlowData), analysis="multiple",
       varFunction= 1, method="ML", verbose=FALSE)
```

Arguments

<code>mixlowData</code>	A list obtained from the output of <code>prepareData</code> . It contains adjusted concentration-response data.
<code>nlsData</code>	A list obtained from the output of <code>doNls</code> . It contains parameter estimates for the concentration-effect curves, which are used as fixed-effects starting values by the <code>nlme</code> function.
<code>drugs</code>	A vector of drug names
<code>analysis</code>	An optional character string either "single" or "multiple" where "single" indicates that each drug is to be analyzed separately and "multiple" indicates that all drugs should be analyzed together. The default is "multiple".
<code>varFunction</code>	If <code>analysis</code> is "multiple", an optional numerical vector from the set (1,2,3,4) that specifies the variance function(s) to be used for each analysis. The default is 1. If <code>analysis</code> is "single" a list, named by drug, must be provided that specifies a vector of variance functions to be used for each drug.
<code>method</code>	An optional character string. If "REML" the model is fit by maximizing the restricted log-likelihood. If "ML" the log-likelihood is maximized. Default is "ML".
<code>verbose</code>	An optional logical value. If TRUE information from intermediate steps in the <code>doNlme</code> analysis is printed. The default is FALSE.

Details

This function uses output from `doNls` as starting values for fixed effects. It organizes input and sets up calls to the `nlme` function in order to estimate parameters of sigmoidal concentration-response curves.

The variance functions for analysis of single drugs are:

- 1 `sigma`
- 2 `sigma*E[response]`

- 3 $\sigma * E[\text{response}]^{\beta}$
- 4 $\sigma * (\beta_1 + E[\text{response}])$

The variance functions for analysis of multiple drugs are:

- 1 σ
- 2 $\sigma * \alpha$, where α is drug-dependent
- 3 $\sigma * E[\text{response}]$
- 4 $\sigma * E[\text{response}]^{\beta}$, where β is drug-dependent

$E[\text{response}]$ is used above to designate the expected response in a given well of a tray. For heteroscedastic errors, `varFunction=2` (for single) and `varFunction=3` (for multiple) analysis may be appropriate in many cases. The simpler error functions could be tried if the `nlme` function does not converge.

See Boik et al., 2008 for details of the model. The parameters estimated by the model are:

- g** γ , designates the steepness of the concentration-effect curve at the IC50
- p** ψ , designates the IC50
- u** $E[\exp(u) + b_t]$, the expected value of all control wells across trays, where b is a tray-dependent random variable
- lambda** Optional. The non-zero asymptote of a “modified” sigmoidal concentration-response curve. The value of λ represents the fraction of $E[\exp(u) + b_t]$ that is associated with the non-zero asymptote.

Parameters u , g , and p are in log scale.

Use of `analysis=“multiple”` for mixtures that contain more than two drugs can sometimes be problematic in that the estimation procedure may not converge or may take a long time to converge. If a mixture contains more than a few drugs, one alternative is to estimate concentration-response curve parameters for each drug/mixture separately (i.e., use `analysis=“single”`). The disadvantage with this approach is that for analysis of any given drug, control-well data from the trays of other drugs are not used. Control-well responses are then estimated based on only a few (replicate) trays.

Value

Returns a list of class `nlmeData` for each analysis with the following components:

- `nlmeResults` A list of results from the `doNlme` analysis:
 - nam** Name of the best model, chosen according to BIC score
 - method** Method used to estimate the model
 - drug** A string containing drug names
 - setNum** The sequential number of the analysis
 - cell** String indicating the cell line
 - lik** Log likelihood
 - bic** BIC score
 - sig** σ
 - modelstruct** A list containing $\log(\text{stdev}(b_t)/\sigma)$ and other variance structures

	rc Goodness-of-fit statistic
	r2 r^2 goodness-of-fit statistic
	se Standard errors
	coeff Fixed effects
	df Degrees of freedom
	covv Parameter covariance values
nlmeGraph	A list of data used for graphing nlme results:
	pred0 Predicted values
	dat1 Data values
	ord A string containing drug names
	residu Residuals from the best model
	best A string denoting the name of the best model
nlmeModels	A list of the best model objects

Author(s)

John Boik <john.boik@newearthbiomed.org>

References

Boik J.C., Newman R.A., Boik R.J. (2008). Quantifying Synergism/Antagonism Using Nonlinear Mixed-Effects Modeling: A Simulation Study. *Statistics in Medicine*, **27**(7), 1040–1061

Boik J.C., Narasimhan N. (2010). An R Package for Assessing Drug Synergism/Antagonism. *Journal of Statistical Software*, **34**(6), 1–18.

See Also

[doNls](#), [plot.nlmeData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
# nlsData data object is obtained using the doNls function
data(nlsData)
drugs = getDrugs(mixlowData)
nlmeData = doNlme(mixlowData, nlsData, drugs=drugs, varFunction= c(1,2))
```

doNls	<i>Conduct nonlinear least-squares analysis of sigmoidal concentration-response data</i>
-------	--

Description

Organizes input and calls the `nls` function to estimate the parameters of sigmoidal concentration-response curves. Users can supply starting values for parameters or can use default values that are based on the data.

Usage

```
doNls(mixlowData, parameterDefaults, lambdaThreshold=0.05, numRandomTries=10,
      verbose=FALSE)
```

Arguments

<code>mixlowData</code>	A list obtained from <code>prepareData</code> .
<code>parameterDefaults</code>	A list obtained from <code>getNlsParameterDefaults</code> . The default values can be edited as desired before sending to the <code>doNls</code> function.
<code>lambdaThreshold</code>	A scalar designating any lower boundary for <code>lambda</code> . If a model estimates a <code>lambda</code> value less than the threshold, that model will be discarded and a model that does not use a <code>lambda</code> parameter will be used instead.
<code>numRandomTries</code>	A scalar designating the number of attempts that should be made to reach convergence for each model. For each attempt, starting values are randomly selected from a normal distribution with mean equal to the given parameter starting value and a standard deviation equal to the given standard deviation.
<code>verbose</code>	An optional logical value. If <code>TRUE</code> , information from intermediate steps in the <code>doNls</code> analysis is printed. Default is <code>FALSE</code> .

Details

See `doNlsme` for a description of the parameters estimated. See `getNlsParameterDefaults` for a description of default parameter values. Parameter values of `NaN` indicate the following default values will be used: `param.g=0`, `param.g.std=1`, `param.p` = the closest tested log concentration corresponding to 1/2 the mean response of control wells, and `param.p.std` = the absolute value of `param.p/2`. The default for `param.lambda` is `NaN`, which indicates that a `lambda` term will not be used. If a `lambda` term is used, its value should be between zero and 0.5.

Value

A list of type `nlsData` of length equal to the number of trays with the following components:

<code>nlsEstimates</code>	Data frame containing parameter estimates
---------------------------	---

nlsGraphing List used by plot.nlsData for making graphs:
tray String tray name
drug String drug name
xx Concentration values for graphing predictions
pred Predicted values
y Adjusted observed responses
x2 Adjusted observed concentrations

Adjusted responses take into account responses in optical control wells (see prepareData). In the adjusted concentrations, values of zero are replaced by concentrations of 1/1000th of the lowest nonzero concentration (for log-plot graphing purposes only).

Author(s)

John Boik <john.boik@newearthbiomed.org>

References

Boik J.C., Newman R.A., Boik R.J. (2008). Quantifying Synergism/Antagonism Using Nonlinear Mixed-Effects Modeling: A Simulation Study. *Statistics in Medicine*, **27**(7), 1040–1061
Boik J.C., Narasimhan N. (2010). An R Package for Assessing Drug Synergism/Antagonism. *Journal of Statistical Software*, **34**(6), 1–18.

See Also

[prepareData](#), [plot.nlsData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
trays <- getTrays(mixlowData)
parameterDefaults <- getNlsParameterDefaults(trays=trays[1:9])
nlsData <- doNls(mixlowData=mixlowData, parameterDefaults)
```

getCellLines

Retrieve a vector of cell line names from data

Description

Convenience function called on output from prepareData or readDataFile to obtain a vector of cell line names.

Usage

```
getCellLines(data, drugs=NULL, trays=NULL)
```

Arguments

data	A list obtained from prepareData or readDataFile functions
drugs	A vector of drug names. If NULL, cell lines will not be subset by drugs.
trays	A vector of tray names. If NULL, cell lines will not be subset by trays.

Details

A data file may contain experimental results for multiple cell lines. If so, the prepareData function requires that one of these be specified for further analysis. Slices of the vector returned from getCellLines can be used as an argument to the prepareData function to subset the data.

Value

A vector of strings indicating which cell lines are contained in the input data object.

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[getTrays](#), [getDrugs](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data("mixlowData")
cellLines <- getCellLines(mixlowData)
```

getDrugs

Retrieve a vector of drug names from data

Description

Convenience function called on output from prepareData or readDataFile to obtain a vector of drug names.

Usage

```
getDrugs(data, trays=NULL, cellLines=NULL)
```

Arguments

data	A list obtained from prepareData or readDataFile functions.
trays	A vector of tray names. If NULL, drugs will not be subset by trays.
cellLines	A vector of cell line names. If NULL, drugs will not be subset by cell lines.

Details

A data file may contain experimental results for multiple drugs. If so, slices of the vector returned from `getDrugs` can be used as an argument to the `prepareData` function in order to select a subset of the data.

Value

A vector of strings indicating which drugs are contained in the input data object.

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[getTrays](#), [getCellLines](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data("mixlowData")
drugs <- getDrugs(mixlowData)
```

`getNlsParameterDefaults`

Generate a data frame of default values for nonlinear least-squares analysis

Description

Convenience function to generate default values for input to the `doNls` function. Defaults are indicated by NaN values. Users can manually change these values once the data frame is generated.

Usage

```
getNlsParameterDefaults(trays)
```

Arguments

`trays` A vector of tray names that can be generated by the `getTrays` function.

Details

See `doNls` for a description of the parameters. The data frame contains names:

param.g gamma

param.g.std Standard deviation of gamma

param.p psi

param.p.std Standard deviation of psi

param.lambda lambda

See help for functions `doNls` and `doNlsme` for information on parameters. `getNlsParameterDefaults` sets all parameter values to NaN, instructing the `doNls` function to use default values. After generating the data frame, NaNs can be set to numerical values as desired. All parameter values except lambda should be in log scale.

Value

A data frame containing NaN values for all parameter estimates.

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNls](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
trays <- getTrays(mixlowData)
parameterDefaults <- getNlsParameterDefaults(trays=trays[1:9])
parameterDefaults["t2d1", "param.lambda"] <- .05
```

getTrays

Retrieve a vector of tray names from data

Description

Convenience function called on output from `prepareData` or `readDataFile` to obtain a vector of tray names.

Usage

```
getTrays(data, drugs=NULL, cellLines=NULL)
```

Arguments

data	A list obtained from prepareData or readDataFile functions.
drugs	A vector of drug names. If NULL, trays will not be subset by drugs.
cellLines	A vector of cell line names. If NULL, trays will not be subset by cell lines.

Details

A data file will contain experimental results for multiple trays, with one drug tested per tray. Slices of the vector returned from getTrays can be used as an argument to the prepareData function in order to select a subset of the data.

Value

A vector of strings indicating which trays are contained in the input data object.

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[getCellLines](#), [getDrugs](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data("mixlowData")
trays <- getTrays(mixlowData)
```

loeweData

loeweData Dataset

Description

Dataset obtained as a result of using function doLoewe, based on data in the text file "A549_vin_topo_data.txt".

Usage

```
data(loeweData)
```

Format

See help file for function doLoewe for composition of loeweData dataset.

Details

The text file "A549_vin_topo_data.txt" provides example data for a cytotoxicity experiment in which two drugs, vincristine and topotecan, and their 1:1 mixture is tested in A549 human lung cancer cells. Each drug is tested in three replicate trays. See reference for function doLoewe.

See Also[doLoewe](#)**Examples**

```
data(loeweData)
```

mixlowData

mixlowData Dataset

Description

Dataset obtained as a result of using function `prepareData`, based on data in the text file “A549_vin_topo_data.txt”.

Usage

```
data(mixlowData)
```

Format

See help file for function `prepareData` for composition of `mixlowData` dataset.

Details

The text file “A549_vin_topo_data.txt” provides example data for a cytotoxicity experiment in which two drugs, vincristine and topotecan, and their 1:1 mixture is tested in A549 human lung cancer cells. Each drug is tested in three replicate trays. See reference for function `doLoewe`.

See Also[prepareData](#)**Examples**

```
data(mixlowData)
```

`nlmeData`*nlmeData Dataset*

Description

Dataset obtained as a result of using function `doNlme`, based on data in the text file “A549_vin_topo_data.txt”.

Usage

```
data(nlmeData)
```

Format

See help file for function `doNlme` for composition of `nlmeData` dataset.

Details

The text file “A549_vin_topo_data.txt” provides example data for a cytotoxicity experiment in which two drugs, vincristine and topotecan, and their 1:1 mixture is tested in A549 human lung cancer cells. Each drug is tested in three replicate trays. See reference for function `doLoewe`.

See Also

[doNlme](#)

Examples

```
data(nlmeData)
```

`NlmePrintVarFunctions` *Print variance function options for nonlinear mixed-effects analysis*

Description

Prints a table of variance function options that can be used in arguments to the `doNlme` function.

Usage

```
NlmePrintVarFunctions()
```

Arguments

None

Details

One argument to the `doNlme` function is specification of the variance functions to be used in `nlme` analysis. Eight different variance functions are allowed. The convenience function `NlmePrintVarFunctions` prints information on each of the eight possible choices. There are four choices for analysis type “single” and four for analysis type “multiple”.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNlme](#)

Examples

```
NlmePrintVarFunctions()
```

nlsData

nlsData Dataset

Description

Dataset obtained as a result of using function `doNls`, based on data in the text file “A549_vin_topo_data.txt”.

Usage

```
data(nlsData)
```

Format

See help file for function `doNls` for composition of `nlsData` dataset.

Details

The text file “A549_vin_topo_data.txt” provides example data for a cytotoxicity experiment in which two drugs, vincristine and topotecan, and their 1:1 mixture is tested in A549 human lung cancer cells. Each drug is tested in three replicate trays. See reference for function `doLoewe`.

See Also

[doNls](#)

Examples

```
data(nlsData)
```

plot.loeweData	<i>Plot a loeweData object obtained from Loewe analysis</i>
----------------	---

Description

Plots the loeweData object returned from the doLoewe function.

Usage

```
## S3 method for class 'loeweData'  
plot(x, ...)
```

Arguments

x	A loeweData object returned from the doLoewe function.
...	Additional optional arguments sent to the plot function, including: ask, main, xlab, ylab, and legend. Argument ask is logical. If TRUE (and the R session is interactive) the user is asked for input before a new figure is drawn.

Details

Two figures are generated from the loeweData object. The first shows the estimated index and its confidence intervals, along with the reference index at 1.0. Values below this reference line are indicative of synergism and values above are indicative of antagonism. When the upper and lower confidence interval limits both fall below 1.0, then statistically significant synergism is occurring. When the upper and lower confidence interval limits both fall above 1.0, then statistically significant antagonism is occurring. The second figure shows the estimated concentration- response curves for each drug and the mixture. To save both figures to disk, sandwich the plot command between pdf and dev.off commands.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

Examples

```
# loeweData data object is obtained using the doLoewe function  
data(loeweData)  
plot(loeweData, ask=TRUE)
```

plot.mixlowData *Plot a mixlowData object*

Description

Plots mixlowData object obtained from the prepareData function.

Usage

```
## S3 method for class 'mixlowData'  
plot(x, ...)
```

Arguments

x A mixlowData object obtained from the prepareData function.

... Additional optional arguments sent to the plot function, including: ask, trays, with default getTrays(x), and showBlanks, with default FALSE. Argument ask is logical. If TRUE (and the R session is interactive) the user is asked for input before a new figure is drawn. Argument trays is a vector of strings indicating the tray names for which data should be plotted. Argument showBlanks is logical. If TRUE, data from the optical control wells (“blanks”) in a tray are included in the graph. The remaining portions of the graph show data after adjustment for the optical control well values. Default is FALSE

Details

This function plots a mixlowData object. Adjusted concentration-response data are plotted, and mean responses over replicate wells receiving identical drug concentrations are connected by a line. The adjustments subtracted from the raw data (see PrepareData function) can be included in the plot by setting showBlanks equal to TRUE. For purposes of plotting on a log-scale plot, drug concentrations of zero are changed to 1/1000 of the lowest nonzero drug concentration. This adjustment only affects plotting, not analysis. To save both figures to disk, sandwich the plot command between pdf and dev.off commands.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[readDataFile](#), [prepareData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
trays <- getTrays(mixlowData)
plot(mixlowData, trays= trays[1:9], ask=TRUE, showBlanks= FALSE)
```

plot.nlmeData

Plot nlmeData object obtained from nonlinear mixed-effects analysis

Description

Plots nlmeData object obtained from doNlme function

Usage

```
## S3 method for class 'nlmeData'
plot(x, ...)
```

Arguments

x A nlmeData object obtained from the doNlme function
... Additional optional arguments sent to the plot function, including: ask, main, xlab, ylab, and legend. Argument ask is logical. If TRUE (and the R session is interactive) the user is asked for input before a new figure is drawn.

Details

Four sets of figures are generated per nlme analysis. The first is concentration-response curves for all trays treated with a given drug, the second is qqnorm plots of residuals for each drug, the third is residual versus fitted values for each drug, and the fourth is standardized residuals versus fitted values for each drug.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNlme](#)

Examples

```
# nlmeData data object is obtained using the doNlme function
data(nlmeData)
plot(nlmeData, ask=TRUE)
```

plot.nlsData	<i>Plot nlsData object obtained from nonlinear least-squares analysis</i>
--------------	---

Description

Plots nlsData object obtained from the doNls function

Usage

```
## S3 method for class 'nlsData'  
plot(x, ...)
```

Arguments

x	A nlsData object obtained from the doNls function
...	Additional arguments sent to the plot function, including: ask, mixlowData, trays, with default getTrays(mixlowData), and showBlanks, with default FALSE. All arguments are optional except mixlowData. Argument ask is logical. If TRUE (and the R session is interactive) the user is asked for input before a new figure is drawn. Argument mixlowData is a mixlowData object obtained from the prepareData function. Argument trays is a vector of strings indicating the tray names for which data should be plotted. Argument showBlanks is logical. If TRUE, data from the optical control wells (“blanks”) in a tray are included in the graph. The remaining portions of the graph show data after adjustment for the optical control well values. Default is FALSE

Details

Plots concentration-response curves for each tray and each drug. See plot.mixlowData for more information about the plots of concentration-response data and optical control data.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNls](#), [plot.mixlowData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
trays <- getTrays(mixlowData)
# nlsData data object is obtained using the doNls function
data(nlsData)
plot(nlsData, mixlowData=mixlowData,
     trays= trays[1:9], ask=FALSE, showBlanks= FALSE)
```

```
prepareData
```

```
Prepares raw data for use by other functions
```

Description

Prepares data for use by doNls and other functions.

Usage

```
prepareData(trayData, drugs, trays, cellLines, degree=3)
```

Arguments

trayData	A list returned from readDataFile
drugs	A vector of drug names used to subset data
trays	A vector of tray names used to subset data
cellLines	A vector of cell line names used to subset data
degree	The degree of the polynomial used in the model for bbc blank wells. Must be in the set [1,2,3,4]

Details

Data contained in a trayData object (produced by use of the readDataFile function) must be adjusted before it can be used by other functions of this package. The prepareData function adjusts observed responses based on responses in the optical control (“blank”) wells. The method used to adjust observed responses is stated in the data file (methods are either “bbt” or “bbc”). For method “bbc”, a 1st to 4th degree polynomial is fit to the concentration-dependent optical control data, depending on the value of the argument degree. Results of using different polynomial degrees can be seen by using the function plot.mixlowData with argument showBlanks set to TRUE. Predictions for each concentration are subtracted from the observed experimental responses. For type “bbt”, the average optical control response is subtracted from all observed responses.

To aid in forming the arguments drugs, trays, and cellLines, vectors of each can be obtained using the convenience functions getDrugs, getTrays, and getCellLines.

Value

A list of class `mixlowData` with the following components:

<code>concentrationResponse</code>	A data frame containing adjusted concentration-response data
<code>drugRatios</code>	A data frame containing drug ratios
<code>plottingData</code>	A list used by <code>plot.mixlowData</code> in plotting

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[readDataFile](#), [plot.mixlowData](#)

Examples

```
# trayData data object is obtained using the readDataFile function
data(trayData)
trays <- getTrays(trayData)
cellLines <- getCellLines(trayData)
mixlowData <- prepareData(trayData=trayData, trays=trays[1:9], cellLines=cellLines[1])
```

<code>print.loeweData</code>	<i>Prints a loeweData object obtained from Loewe analysis</i>
------------------------------	---

Description

Prints a `loeweData` object returned from the `doLoewe` function.

Usage

```
## S3 method for class 'loeweData'
print(x, ...)
```

Arguments

<code>x</code>	A <code>loeweData</code> object returned from the <code>doLoewe</code> function.
<code>...</code>	Optional arguments passed to <code>print.default</code> , including <code>verbose</code> , a logical argument with default <code>TRUE</code> . <code>Verbose</code> controls the amount of output printed.

Details

Prints a `loeweData` object returned from the `doLoewe` function.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

Examples

```
# loeweData data object is obtained using the doLoewe function
data(loeweData)
print(loeweData)
```

print.mixlowData *Print a mixlowData object*

Description

Prints a mixlowData object obtained from the prepareData function.

Usage

```
## S3 method for class 'mixlowData'
print(x, ...)
```

Arguments

x A mixlowData object obtained from the prepareData function.
... Optional arguments passed to print.default, including verbose, a logical argument with default TRUE. Verbose controls the amount of output printed.

Details

Prints a mixlowData object obtained from the prepareData function.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[readDataFile](#), [prepareData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
print(mixlowData)
```

print.nlmeData	<i>Print a nlmeData object obtained from nonlinear mixed-effects analysis</i>
----------------	---

Description

Prints a nlmeData object obtained from doNlme function

Usage

```
## S3 method for class 'nlmeData'
print(x, ...)
```

Arguments

x	A nlmeData object obtained from the doNlme function
...	Optional arguments passed to print.default, including verbose, a logical argument with default TRUE. Verbose controls the amount of output printed.

Details

Prints a nlmeData object obtained from doNlme function

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNlme](#)

Examples

```
# nlmeData data object is obtained using the doNlme function
data(nlmeData)
print(nlmeData)
```

print.nlsData	<i>Prints a nlsData object obtained from nonlinear least-squares analysis</i>
---------------	---

Description

Prints a nlsData object obtained from the doNls function

Usage

```
## S3 method for class 'nlsData'  
print(x, ...)
```

Arguments

x	A nlsData object obtained from the doNls function
...	Optional arguments passed to print.default, including verbose, a logical argument with default TRUE. Verbose controls the amount of output printed.

Details

Prints a nlsData object obtained from the doNls function

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNls](#)

Examples

```
# nlsData data object is obtained using the doNls function  
data(nlsData)  
print(nlsData)
```

readDataFile	<i>Read a formatted file of concentration-response data</i>
--------------	---

Description

Reads a formatted data file that contains concentration-response information.

Usage

```
readDataFile(filename, excludeWells)
```

Arguments

filename	A string designating the data file to be read.
excludeWells	An optional list that specifies cells that should be skipped in the analysis. Cells are identified by row and column numbers. Paired components of the list are: row skip a cell in row col skip a cell in column

Details

The data file read by this function must be formatted according to a specific structure. See the example file `mixlowExampleData.txt` for use as a template. The data file is tab-delimited pure ASCII (without enclosing quotations) and the first entry on each line is a line label. The first eight lines contain general information about the experiments. Of these lines, only the entries `conc_units`, `rows`, and `cols` are required. The remainder of the data file is split into blocks, one block for each tray. For each block, the required entries are `tray_label`, `cell_line`, `drug_name_short`, `composition`, as well as the entries for `conc` (concentration), `label`, and `resp` (response) for each row and column of a tray. Even if the optional entries are left blank, the line labels must still be included (including the “space” labels).

The “ray” design should be used for each experiment. In this design, various dilutions of a drug or mixture are tested in the wells. If a mixture is being tested, a fixed ratio between component drugs occurs at every concentration tested. A tray should contain data for only a single drug or mixture. Replicate trays should be analyzed for each drug or mixture (for a given cell line). At least three replicate trays are recommended. However, if only one tray is available for a drug, `doNlme` will duplicate that tray (to make two replicates) so that analysis can be done in “experimental” mode. Tray labels can be alphanumeric, but avoid any characters other than digits and letters (the underscore can be used). The same is true for the abbreviated drug names that are required.

Each tray should contain wells used as treatment controls (i.e., those that receive cells but no drug). In addition, trays should contain wells used as optical controls (called “blanks” here). These can be of two types: 1) wells that contain only media, and 2) wells at each drug concentration that contain media and drug. The first type is referred to as “bbt”, or blanks-by-tray, and the second type is referred to as “bbc”, or blanks-by-concentration. The use of “bbc” is recommended when concentration-dependent responses can be induced by the drug alone (e.g., autofluorescence). For type “bbt”, raw responses are adjusted by subtracting the mean response of all “bbt” wells in a tray.

For type “bbc”, a 4th degree polynomial is fit to the “bbc” responses and predicted values for each concentration are subtracted from treatment well responses.

In many cases, a reasonable arrangement for a 96-well tray in a cytotoxicity experiment is to use the last two rows of every column as “bbc” blanks, use the first six rows of the first two columns as treatment controls, and use the first six rows of the third to twelfth columns as treatment wells, with each column receiving a different drug concentration.

When conducting large simulations, it is recommended that many small files be read from disk rather than one large file. The time required to read large files can be prohibitive.

Value

A list of class trayData with components:

concentrationResponse	Data frame containing concentration- response information
drugRatios	Data frame containing summary information for each tray

Author(s)

John Boik <john.boik@newearthbiomed.org>

References

Boik J.C., Narasimhan N. (2010). An R Package for Assessing Drug Synergism/Antagonism. *Journal of Statistical Software*, **34**(6), 1–18.

See Also

[prepareData](#)

Examples

```
dataFile = system.file(package="mixlow", "exdata/mixlowExampleData.txt")
excludeWells= list(row= c(1,2), col= c(1,1))
trayData <- readDataFile(filename=dataFile, excludeWells=excludeWells)
```

summary.loeweData	<i>Summarizes a loeweData object obtained from Loewe analysis</i>
-------------------	---

Description

Summarizes a loeweData object returned from the doLoewe function.

Usage

```
## S3 method for class 'loeweData'
summary(object, ...)
```

Arguments

object A loeweData object returned from the doLoewe function.
... Optional arguments passed to summary.default

Details

Summarizes a loeweData object returned from the doLoewe function.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

Examples

```
# loeweData data object is obtained using the doLoewe function
data(loeweData)
summary(loeweData)
```

summary.mixlowData *Summarizes a mixlowData object*

Description

Summarizes a mixlowData object obtained from the prepareData function.

Usage

```
## S3 method for class 'mixlowData'
summary(object, ...)
```

Arguments

object A mixlowData object obtained from the prepareData function.
... Optional arguments passed to summary.default

Details

Summarizes a mixlowData object obtained from the prepareData function.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[readDataFile](#), [prepareData](#)

Examples

```
# mixlowData data object is obtained using the prepareData function
data(mixlowData)
summary(mixlowData)
```

summary.nlmeData	<i>Summarizes a nlmeData object obtained from nonlinear mixed-effects analysis</i>
------------------	--

Description

Summarizes a nlmeData object obtained from doNlme function

Usage

```
## S3 method for class 'nlmeData'
summary(object, ...)
```

Arguments

object	A nlmeData object obtained from the doNlme function
...	Optional arguments passed to summary.default

Details

Summarizes a nlmeData object obtained from doNlme function

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNlme](#)

Examples

```
# nlmeData data object is obtained using the doNlme function
data(nlmeData)
summary(nlmeData)
```

summary.nlsData	<i>Summarizes a nlsData object obtained from nonlinear least-squares analysis</i>
-----------------	---

Description

Summarizes a nlsData object obtained from the doNls function

Usage

```
## S3 method for class 'nlsData'
summary(object, ...)
```

Arguments

object	A nlsData object obtained from the doNls function
...	Optional arguments passed to summary.default

Details

Summarizes a nlsData object obtained from the doNls function

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[doNls](#)

Examples

```
# nlsData data object is obtained using the doNls function
data(nlsData)
summary(nlsData)
```

summary.trayData	<i>Summarizes a trayData object</i>
------------------	-------------------------------------

Description

Summarizes a trayData object obtained from the readDataFile function.

Usage

```
## S3 method for class 'trayData'  
summary(object, ...)
```

Arguments

object	A trayData object obtained from the readDataFile function.
...	Optional arguments passed to summary.default

Details

Summarizes a trayData object obtained from the readDataFile function.

Value

None

Author(s)

John Boik <john.boik@newearthbiomed.org>

See Also

[readDataFile](#)

Examples

```
# trayData data object is obtained using the readDataFile function  
data(trayData)  
summary(trayData)
```

trayData

trayData Dataset

Description

Dataset obtained as a result of using function `readDataFile`, based on data in the text file “mixlowExampleData.txt”. This file contains made-up example data.

Usage

```
data(trayData)
```

Format

See help file for function `readDataFile` for composition of `trayData` dataset.

Details

The text file “mixlowExampleData.txt” provides example data for a cytotoxicity experiment in which two drugs and their mixture is tested. Each drug is tested in three replicate trays.

See Also

[readDataFile](#)

Examples

```
data(trayData)
```

Index

- *Topic **aplot**
 - prepareData, 21
 - *Topic **datasets**
 - loeweData, 13
 - mixlowData, 14
 - nlmeData, 15
 - nlsData, 16
 - trayData, 32
 - *Topic **methods**
 - plot.loeweData, 17
 - plot.mixlowData, 18
 - plot.nlmeData, 19
 - plot.nlsData, 20
 - print.loeweData, 22
 - print.mixlowData, 23
 - print.nlmeData, 24
 - print.nlsData, 25
 - summary.loeweData, 27
 - summary.mixlowData, 28
 - summary.nlmeData, 29
 - summary.nlsData, 30
 - summary.trayData, 31
 - *Topic **models**
 - doLoewe, 3
 - doNlme, 5
 - doNls, 8
 - *Topic **package**
 - mixlow-package, 2
 - *Topic **print**
 - getDrugs, 10
 - NlmePrintVarFunctions, 15
 - *Topic **programming**
 - getCellLines, 9
 - getNlsParameterDefaults, 11
 - getTrays, 12
 - *Topic **utilities**
 - readDataFile, 26
- doLoewe, 3, 14
- doNlme, 4, 5, 15, 16, 19, 24, 29
- doNls, 7, 8, 12, 16, 20, 25, 30
- getCellLines, 9, 11, 13
- getDrugs, 10, 10, 13
- getNlsParameterDefaults, 11
- getTrays, 10, 11, 12
- loeweData, 13
- mixlow (mixlow-package), 2
- mixlow-package, 2
- mixlowData, 14
- nlmeData, 15
- NlmePrintVarFunctions, 15
- nlsData, 16
- plot.loeweData, 4, 17
- plot.mixlowData, 18, 20, 22
- plot.nlmeData, 7, 19
- plot.nlsData, 9, 20
- prepareData, 9, 14, 18, 21, 23, 27, 29
- print.loeweData, 22
- print.mixlowData, 23
- print.nlmeData, 24
- print.nlsData, 25
- readDataFile, 18, 22, 23, 26, 29, 31, 32
- summary.loeweData, 27
- summary.mixlowData, 28
- summary.nlmeData, 29
- summary.nlsData, 30
- summary.trayData, 31
- trayData, 32