Package ‘mixtox’

February 18, 2017

Type Package

Title Curve Fitting and Mixture Toxicity Assessment

Version 1.3.2

Depends R (>= 3.0.2)

Imports minpack.lm

LazyData true

Description Curve Fitting for monotonic(sigmoidal) & non-monotonic(J-shaped) concentration-response data. Prediction of mixture toxicity based on reference models such as 'concentration addition', 'independent action', and 'generalized concentration addition'.

License GPL-2

Date 2017-02-18

Author Xiangwei Zhu

Maintainer Xiangwei Zhu <xwzhunc@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-18 21:23:02

R topics documented:

antibiotox 2
BMD 3
caPred 5
CEx 7
curveFit 8
cytotox 12
DTcv 14
ecaPred 14
ECx 16
eiaPred 17
figPlot 19

gcaHill 20
gcaPred 22
getCI 24
hormesis 25
iaPred 26
jacobian 28
mixtox 29
nmECx 30
NOEC 31
qq4res 33
showEq 33
staval 35
tuneFit 36
unidTab 37

Index **40**

`antibiotox` *Toxicity of Seven Antibiotics on Photobacteria*

Description

Seven antibiotics are: Neomycin sulfate(NEO), streptomycin sulfate(STR), kanamycin sulfate(KAN), spectinomycin dihydrochloridehydrate(SPE), paromomycin sulfate(PAR), dihydrostreptomycin sesquisulfate hydrate(DIH), and gentamycin sulfate(GEN). Their toxicity on photobacteria Q67 were tested using microplate toxicity analysis.
The concentration-responses include the toxicity of seven antibiotics, two eecr mixtures, and ten mixtures designed by udcr. The curve fitting information of seven antibiotics and a total of 12 mixtures.

Usage

```
data(antibiotox)
```

Format

A list with concentration-response data of 7 antibiotics, 12 mixtures, and associated fitting information.

`PAR$x` a numeric vector of concentrations

`PAR$y` a numeric matrix of responses

`PAR$name` name of test substance

`PAR$concNum` the number of concentrations

`PAR$tierNum` the number of repetitions

`PAR$type` type of test substance

`sgl$model` models used to fit the concentration-response data `sgl$param` fitted coefficients of concentration-response curves

`udcr.mix$model` models used to fit the concentration-response data of udcr mixtures on photobacteria

`udcr.mix$param` fitted coefficients of concentration-response curves

`udcr.pct` the percentage of individual chemicals in the udcr mixtures

`eecr.mix$model` models used to fit the concentration-response data of eecr mixtures on photobacteria

`eecr.mix$param` fitted coefficients of concentration-response curves

`eecr.pct` the proportion of individual chemicals in the eecr mixtures

Details

Quantal responses[0, 1] are needed for curve fitting using the following six equaitons: i.e., Weibull, Logit, Hill, BCL, GL, BCW. The following equation could transform continous responses to quantal ones:

$$E = \frac{I_0 - I_i}{I_0}$$

where I_0 is the average of controls for inhibition tests or the average of the maximum effect for stimulation tests and I_i the average effect of the i^{th} treatment.

Examples

```
# example 1
## Retrive the toxicity information of PAR on photobacteria.
antibiotox$PAR
# example 2
## Retrive the toxicity information of two eecr mixtures on photobacteria.
antibiotox$eecr.mix
```

BMD *Calculating benchmark dose (BMD) and lower limit of benchmark dose (BMDL)*

Description

Calculation of BMD and BMDL for both quantal and continuous dose responses. Six 2- or 3-paramter models ('Hill', 'Weibull', 'Logit', 'Weibull_three', 'Hill_three', 'Logit_three') were employed for quantal dose responses. Three 4-parameter models ('Weibull_four', 'Logit_four', 'Hill_four') were employed for continuous data.

Usage

```
BMD(object, bmr = 0.10, backg = 0, def = 'additional', eq = 'as.is',
        sigLev = 0.05, ci = 'CI')
```

Arguments

`object` object of class curveFit.

`bmr` numeric vector of bench mark response levels for which to calculate benchmark doses (should be between 0 and 1)

`backg` numeric value specifying the background level (defaults to 0)

`def` character string specifiying the definition of the benchmark dose to use in the calculations. "excess" and "additional" are for binomial response whereas "relative" and "hybrid" (additive hybrid) are for continuous response.

`eq` default list of equations.

`sigLev` the significance level for Dunnett's test. The default is 0.05.

`ci` types of confidence intervals (CI or PI).CI: non-simultaneous confidence intervals; PI: non-simultaneous prediction intervals.

Details

Selecting the Benchmark Response Level (BMR) (https://www.epa.gov/bmds/ benchmark-dose-bmd-methods#bmr).BMR is usually set as 0.10.

Value

`bmds` values of BMDL, BMD, and BMDU.

Note

three default equations (Hill, Weibull, and Logit) were used to calculate BMD for quantal dose response. Three default equations('Weibull_four', 'Logit_four', 'Hill_four') were used to calculate BMD for continuous dose response. BMD calculation is only availabale for monotonic dose reponses in this version.

References

Benchmark Dose Technical Guidance, Risk Assessment Forum, U.S. Environmental Protection Agency, Washington, DC 20460, EPA/100/R-12/001, June 2012

Kennyp (2002). Critical Issues in Benchmark Calculations from Continuous Data. Crit. Rev. Toxicol., 32, 133-153.

Examples

```
## example 1
# calcualte the BMD of heavy metal Ni(2+) on the MCF-7 cells
x <- cytotox$Ni$x
rspn <- cytotox$Ni$y
obj <- curveFit(x, rspn, eq = 'Logit', param = c(12, 3), effv = c(0.05, 0.5), rtype = 'quantal')
BMD(obj, bmr = 0.10, backg = 0, def = 'additional', eq = 'default', sigLev = 0.05, ci = 'CI')
```

caPred *Mixture Toxicity Prediction Based on Concentration Addition*

Description

Predicting mixture toxicity based on individual concentration-response information fitted only based on the following six models: Hill, Weibull, Logit, BCW, BCL, and GL. Three optional mixture design methods are provided:(1)arbitrary concentration ratio (acr), users can set an arbitary proportion for each component in a mixture; (2)equal effect concentration ratio (eecr); (3) uniform design concentration ratio (udcr).

Usage

```
caPred(model, param, mixType = c("acr", "eecr", "udcr"), effv, effPoints)
```

Arguments

`model`	vector of models: Hill, Weibull, Logit, BCW, BCL, and GL
`param`	numeric matrix of fitting coefficients with row names (model selected) and column names (Alpha, Beta, and Gamma). For models with only two parameters (i.e., Hill, Weibull, and Logit), Gamma can be set to zero or any other numeric value.
`mixType`	experimental design of the mixture. acr: arbitrary concentration ratio; eecr: equal effect concentration ratio; udcr: uniform design concentration ratio.
`effv`	numeric vector with single or multiple effects (0 ~ 1).
`effPoints`	numeric vector [0 ~ 1] of effects to predict effect concentrations.

Details

Concentration addition (CA) is designed for mixtures of chemicals that have similar mechanisms of action. For a well-defined mixture (e.g., a mixture of n components), CA is expressed mathematically as:

$$\sum_{i=1}^{n} \frac{c_i}{ECx_i} = 1$$

where ECx_i is the effect concentration of the i^{th} compound that causes x% effect when applied individually at c_i. The c_i can be computed from the following equation:

$$c_i = p_i \cdot c_{mix} = p_i \cdot EC_{x,mix}$$

where p_i is the proportion of i^{th} component in the mixture, c_{mix} the mixture concentration and $EC_{x,mix}$ the concentration of the mixture at a specific effect x%. The prediction of combined effects of mixture-components based on CA can then be expressed as:

$$EC_{x,mix} = \left(\sum_{i=1}^{n} \frac{p_i}{EC_{x,i}} \right)^{-1}$$

Value

ca	a series of effect concentrations predicted by CA at effPoints
e	a series of effects (effPoints) associated with the effect concentrations in ca
pct	the proportion of every component in a mixture
uniTab	the uniform design table used to construct the mixture when mixType is 'udcr'

Note

Note that effv is dependent on the mixType. if the mixType is acr, the length of effv should be the same as that of model. when a total ratio was less than 100 if the mixType is eecr, effv should contain at least one effect value (e.g., 0.5). For example, the EC50 values (10-6 mol/L) were 41.8, 238.0, 3.07, 23.4, 978.0, 32.8, 2200.0, and 1530.0 for chemical A, B, C, D, E, F, G, and H, respectively. According to their ratio of EC50 values, the proportion of the eight components in the mixture would be 0.006484, 0.06973, 0.0006581, 0.0003494, 0.1726, 0.001497, 0.6774, and 0.07131.
if the mixTpe is udcr, the length of effv is the number of levels (runs). the number of runs should be in accordance with the length of model based on the uniform design

References

Liang, Yi-zeng, Kai-tai Fang, and Qing-song Xu. 2001. Uniform Design and Its Applications in Chemistry and Chemical Engineering. Chemometrics and Intelligent Laboratory Systems 58(1):43-57.
Backhaus, T., Faust, M., 2012. Predictive environmental risk assessment of chemical mixtures: A conceptual framework. Environmental Science and Technology. 46, 2564-2573.

See Also

iaPred

Examples

```
## example 1
# using CA to predict the toxicity of mixture designed by eecr at the
# effect concentration of EC05 and EC50
# eecr mixture design is based on seven antibiotics(factors).
model <- antibiotox$sgl$model
param <- antibiotox$sgl$param
caPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))

## example 2
# using CA to predict the mixtures designed by udcr
# the udcr mixture design is based on four heavy metals and four ionic liquids (eight factors).
# five levels (EC05, EC10, EC20, EC30, and EC50 ) are allocated in the uniform table using the
# pseudo-level technique (Liang et al., 2001)
model <- cytotox$sgl$model
param <- cytotox$sgl$param
effv <- c(0.05, 0.05, 0.10, 0.10, 0.20, 0.20, 0.30, 0.30, 0.50, 0.50)
caPred(model, param, mixType = "udcr", effv)
```

```
## example 3
# using CA to predict the mixtures designed by acr
# the udcr mixture design is based on five antibiotics (five factors).
# the every component in the mixture shares exactly the same ratio (0.20)
model <- antibiotox$sgl$model[1 : 5]
param <- antibiotox$sgl$param[1 : 5, ]
effv <- c(0.2, 0.2, 0.2, 0.2, 0.2)
caPred(model, param, mixType = "acr", effv)
```

`CEx` *Effect Calculation for All Ninteen Curves*

Description

Calculating responses at particular concentrations.

Usage

```
CEx(model, param, conc)
```

Arguments

`model` a character vector of equation names

`param` a numeric matrix of fitting coefficients with rownames (equation selected) and colnames (ALpha, Beta, and Gamma). For equations with two parameters, Gamma can be set as zero or any other numeric value.

`conc` a numeric vector with single or multiple concentrations.

Details

Responses will be calculated with provided equations (model), associated fitting parameters (param), and concentrations.

Value

`effv` a numeric vector of effect(s)

References

Zhu X-W, et.al. 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicol. Environ. Saf. 89:130-136.
Hill equation (biochemistry) http://en.wikipedia.org/wiki/Hill_equation_(biochemistry)
Scholze, M. et al. 2001. A General Best-Fit Method for Concentration-Response Curves and the Estimation of Low-Effect Concentrations. Environmental Toxicology and Chemistry 20(2):448-457.

Examples

```
## example 1
# calculate the responses of hormesis curves at the concentration of 0.1 and 0.02 mol/L
model <- hormesis$sgl$model
param <- hormesis$sgl$param
CEx(model, param, conc = c(0.1, 0.02))

## example 2
# calculate the effect caused by four heavy metals and four ionic liquids at the concentration of
# 0.00001 and 0.00002 mol/L on the MCF-7 cells
model <- cytotox$sgl$model
param <- cytotox$sgl$param
CEx(model, param, conc = c(0.00001, 0.00002))

## example 3
# calculate the response ranges
model <- hormesis$sgl$model
param <- hormesis$sgl$param
CEx(model, param, conc = c(0, 1e20))
```

---

curveFit *Curve Fitting*

Description

Thirteen monotonic(sigmoidal) models ("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)") and four non-monotonic(J-shaped) models ("Brain_Consens", "BCV", "Biphasic", "Hill_five") are provided to fit dose-response data. The goodness of fit of a model is evaluated by the following statistics: coefficient of determination (R^2), adjusted coefficient of determination (R^2_{adj}), root mean squared error (RMSE), mean absolute error (MAE), Akaike information criterion (AIC), bias-corrected Akaike information criterion(AICc), and Bayesian information criterion (BIC).

Usage

```
curveFit(x, rspn, eq , param, effv, rtype = 'quantal', sigLev = 0.05, ...)
```

Arguments

`x` a numeric vector of experimental concentration.

`rspn` a numeric matrix of experimental responses with one or more replicates.

`eq` equation used for curve fitting: "Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW", "BCL", "GL", "Brain_Consens", "BCV", "Biphasic", "Hill_five".

`param` a vector of starting parameters. Use tuneFit to get the starting values.

`effv` a numeric vector of responses for the calculation of effect concentrations. Minus values(e.g., -5%) are permited only in the condition of 'hormesis' dose-responses. Relative values(e.g., 5%, 10%) in the condition of 'continuous' dose-responses.

`rtype` three dose-response types: 'quantal', 'continuous', 'hormesis'. Default is 'quantal'. 'quantal': dose-responses with lower limit fixed at 0 and higher limit at 1 (100 'continuous': dose-responses with no fixed lower or higher limits. 'hormesis': non-monotonic J or U-shaped dose-responses with lower limit fixed at 0 and higher limit at 1 (100%).

`sigLev` the significant level for confidence intervals and Dunnett\'s test. Default is 0.05.

`...` other arguments passed to nlsLM in minpack.lm.

Details

Curve fitting is dependent on the package minpack.lm (http://cran.r-project.org/web/packages/minpack.lm/index.html). Monotonic(sigmoidal) equations are as follows:
Hill:

$$E = 1/\left(1 + (\alpha/c)^{\beta}\right)$$

Hill_two:

$$E = \beta c/\left(\alpha + c\right)$$

Hill_three:

$$E = \gamma/\left(1 + (\alpha/c)^{\beta}\right)$$

Hill_four:

$$E = \delta + (\gamma - \delta)\,/\left(1 + (\alpha/c)^{\beta}\right)$$

where α = EC50, β = H (Hill coefficient), γ = Top, and δ = Bottom

Weibull:

$$E = 1 - \exp(-\exp(\alpha + \beta\log(c)))$$

Weibull_three:

$$E = \gamma\left(1 - \exp\left(-\exp\left(\alpha + \beta\log\left(c\right)\right)\right)\right)$$

Weibull_four:

$$E = \gamma + \left(\delta - \gamma\right)\exp\left(-\exp\left(\alpha + \beta\log\left(c\right)\right)\right)$$

Logit:

$$E = (1 + \exp(-\alpha - \beta\log(c)))^{-1}$$

Logit_three:

$$E = \gamma/\left(1 + \exp\left(\left(-\alpha\right) - \beta\log\left(c\right)\right)\right)$$

Logit_four:

$$E = \delta + \left(\gamma - \delta\right)/\left(1 + \exp\left(\left(-\alpha\right) - \beta\log\left(c\right)\right)\right)$$

where α is the location parameter and β slope parameter. γ = Top, and δ = Bottom

BCW:

$$E = 1 - \exp\left(-\exp\left(\alpha + \beta\left(\frac{c^\gamma - 1}{\gamma}\right)\right)\right)$$

BCL:

$$E = (1 + \exp(-\alpha - \beta((c^\gamma - 1)/\gamma)))^{-1}$$

GL:

$$E = 1/(1 + \exp(-\alpha - \beta\log(c)))^\gamma$$

Non-monotonic(J-shaped) models:
Hill_five:

$$E = 1 - \left(1 + (\gamma - 1) / \left(1 + (\alpha/c)^\beta\right)\right)(1 - 1/(1 + (\delta/c)^\varepsilon))$$

Brain_Consens:

$$E = 1 - (1 + \alpha c) / \left(1 + \exp(\beta\gamma)\, c^\beta\right)$$

where α is the initial rate of increase at low concentration, β the way in which response decreases with concentration, and γ no simple interpretation.

BCV:

$$E = 1 - \alpha(1 + \beta c) / \left(1 + (1 + 2\beta\gamma)(c/\gamma)^\delta\right)$$

where α is untreated control, β the initial rate of increase at low concentration, γ the concentration cause 50% inhibition, and δ no simple interpretation.

Cedergreen:

$$E = 1 - \left(1 + \alpha\exp\left(-1/\left(c^\beta\right)\right)\right) / (1 + \exp(\gamma(\ln(c) - \ln(\delta))))$$

where α the initial rate of increase at low concentration, β the rate of hormetic effect manifests itself, γ the steepness of the curve after maximum hormetic effect, and δ the lower bound on the EC50 level.

Beckon:

$$E = (\alpha + (1 - \alpha) / (1 + (\beta/c)^\gamma)) / (1 + (c/\delta)^\varepsilon)$$

where α is the minimum effect that would be approached by the downslope in the absence of the upslope, β the concentration at the midpoint of the falling slope, γ the steepness of the rising(positive) slope, δ the concentration at the midpoint of the rising slope, and ϵ the steepness of the falling(negative) slope.

Biphasic:

$$E = \alpha - \alpha/\left(1 + 10^{((c-\beta)\gamma)}\right) + (1 - \alpha)/\left(1 + 10^{((\delta-c)\varepsilon)}\right)$$

where α is the minimum effect that would be approached by the downslope in the absence of the upslope, β the concentration at the midpoint of the falling slope, γ the steepness of the rising(positive) slope, δ the concentration at the midpoint of the rising slope, and ϵ the steepness of the falling(negative) slope.
In all, E represents effect and c represents concentration.

Value

`fitInfo` curve fitting information.

`eq` equation used in curve fitting.

`p` fitted parameters.

`res` residual.

`sta` goodness of fit.

`crcInfo` a numeric matrix with the experimental concentration (x), predicted and experimental responses, experimental responses, lower and upper bounds of (non-simultaneous) prediction intervals (PI.low and PI.up), and lower and upper bounds of (non-simultaneous) confidence intervals (CI.low and CI.up).

`ecx` effect concentrations only if effv is provided.

`effvAbs` Absolute effects corresponding to effv only in the condition of 'continuous' dose-responses.

`rtype` dose-response type.

`rspnRange` response range. The lower limit is the response at extremely low dose. The higher limit is the response at infinite high dose.

`minx` concentration to induce the maximum stimulation for 'continuous' dose-response

`miny` the maximum stimulation for 'continuous' data.

Note

tuneFit is recommended to find the starting values.

References

Scholze, M. et al. 2001. A General Best-Fit Method for dose-response Curves and the Estimation of Low-Effect Concentrations. Environmental Toxicology and Chemistry 20(2):448-457.
Zhu X-W, et.al. 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicol. Environ. Saf. 89:130-136.
Howard GJ, Webster TF. 2009. Generalized concentration addition: A method for examining mixtures containing partial agonists. J. Theor. Biol. 259:469-477.
Spiess, A.-N., Neumeyer, N., 2010. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: A Monte Carlo approach. BMC Pharmacol. 10, 11.
Huet, S., Bouvier, A., Poursat, M.-A., Jolivet, E., 2004. Statistical tools for nonlinear regression: a practical guide with S-PLUS and R examples. Springer Science & Business Media.
Gryze, S. De, Langhans, I., Vandebroek, M., 2007. Using the correct intervals for prediction: A tutorial on tolerance intervals for ordinary least-squares regression. Chemom. Intell. Lab. Syst. 87, 147-154.

Examples

```
## example 1
# Fit hormesis dose-response data.
# Calculate the concentrations that cause 5% of 50% inhibition.
```

```
x <- hormesis$OmimCl$x
rspn <- hormesis$OmimCl$y
curveFit(x, rspn, eq = 'Biphasic', param = c(-0.34, 0.001, 884, 0.01, 128),
effv = 0.5, rtype = 'hormesis')

x <- hormesis$HmimCl$x
rspn <- hormesis$HmimCl$y
curveFit(x, rspn, eq = 'Biphasic', param = c(-0.59, 0.001, 160,0.05, 19),
effv = c(0.05, 0.5), rtype = 'hormesis')

x <- hormesis$ACN$x
rspn <- hormesis$ACN$y
curveFit(x, rspn, eq = 'Brain_Consens', param = c(2.5, 2.8, 0.6, 2.44),
effv = c(0.05, 0.5), rtype = 'hormesis')

x <- hormesis$Acetone$x
rspn <- hormesis$Acetone$y
curveFit(x, rspn, eq = 'BCV', param = c(1.0, 3.8, 0.6, 2.44),  effv = c(0.05, 0.5),
rtype = 'hormesis')

## example 2
# Fit quantal dose-responses: the inhibition of heavy metal Ni(2+) on the growth of MCF-7 cells.
# Calculate the concentrations that cause 5% and 50% inhibition.
x <- cytotox$Ni$x
rspn <- cytotox$Ni$y
curveFit(x, rspn, eq = 'Logit', param = c(12, 3), effv = c(0.05, 0.5), rtype = 'quantal')

## example 3
# Fit quantal dose-responses: the inhibition effect of Paromomycin Sulfate (PAR) on photobacteria.
# Calculate the concentrations that cause 5% and 50% inhibition.
x <- antibiotox$PAR$x
rspn <- antibiotox$PAR$y
curveFit(x, rspn, eq = 'Logit', param = c(26, 4), effv = c(0.05, 0.5))
```

`cytotox` *Cytotoxicity of Heavy Metal Ions and Ionic Liquids on MCF-7*

Description

Chemicals include four heavy metal ions: $NiNO_3 \cdot 6H_2O$(Ni), $ZnSO_4 \cdot 7H_2O$ (Zn), $CuSO_4 \cdot 5H_2O$ (Cu), and $MnCl_2 \cdot 4H_2O$ (Mn); four ionic liquids are 1-Octyl-3-methylimidazolium chloride (Omim), 1-Dodecyl-3-methylimidazolium chloride (Dmim), 1-Ethyl-3-methylimidazolium tetrafluoroborate (Emim), and 1-Hexyl-3-Methylimidazolium tetrafluoroborate (Hmim).
The concentration-response data include the cytotoxicity of eight compounds, two mixtures designed by eecr, and ten mixtures designed by udcr.
The fitting information of eight chemicals and a total of 12 mixtures.

Usage

```
data(cytotox)
```

Format

A list with concentration-response data of 8 chemicals, 2 eecr mixtures, 10 udcr mixtures, and associated fitting information.

`Ni$x` a numeric vector of test concentrations

`Ni$y` a numeric matrix of responses

`Ni$name` test substances

`Ni$concNum` the number of test concentrations

`Ni$tierNum` the number of repetitions

`Ni$type` type of test substance: single chemicals or mixtures

`sgl$model` models used to fit the concentration-response data of individual chemicals

`sgl$param` fitted coefficients of concentration-response curves

`udcr.mix$model` models used to fit the concentration-response data of udcr mixtures

`udcr.mix$param` fitted coefficients of the concentration-response curves corresponding to udcr.mix\$model

`$udcr.pct` the proportion of individual chemicals in udcr mixtures

Details

Quantal responses[0, 1] are needed for curve fitting using the following six equaitons: i.e., Weibull, Logit, Hill, BCL, GL, BCW. The following equation could transform continous responses to quantal ones:

$$E = \frac{I_0 - I_i}{I_0}$$

where I_0 is the average of controls for inhibition tests or the average of the maximum effect for stimulation tests and I_i the average effect of the i^{th} treatment.

Source

The cytotoxicity experiments were conducted in our lab

Examples

```
# example 1
## Retrive the toxicity data of Ni on MCF-7.
cytotox$Ni

# example 2
## Retrive the toxicity information of ten udcr mixtures on MCF-7.
cytotox$udcr.mix
```

`DTcv` *critical value for Dunnett's test*

Description

DTcv provides tThe critical constants calculated based step-down Dunnett test procedure. Three significance level (0.01, 0.05, and 0.1) each with two alternative hypothesis ("U"=upper one-sided test; "B"=two-sided test) are supported.

Usage

```
data(DTcv)
```

Format

at most 30 treatments (1 : 30), and 35 degree of freedom(c(5 : 30, 40, 50, 60, 80, 100, 120, 200, 1000, 3000)).

`DTcv` a matrix of critical value for Dunnett's test

Details

```
> head(DTcv)
df p twoside.01 twoside.05 twoside.10 oneside.01 oneside.05 oneside.10
[1,] 5 1 4.032 2.571 2.015 3.365 2.015 1.476
[2,] 5 2 4.627 3.030 2.433 3.900 2.440 1.873
[3,] 5 3 4.948 3.294 2.669 4.225 2.681 2.095
[4,] 5 4 5.218 3.474 2.831 4.434 2.848 2.245
[5,] 5 5 5.416 3.616 2.956 4.585 2.974 2.360
[6,] 5 6 5.538 3.727 3.055 4.723 3.080 2.451
```

df: degree of freedom; p: the number of treatment.

Source

The critical constants (store in DTcv) were calculated using step-down Dunnett test procedure(the cvSDDT funtion in R package DunnettTests).

`ecaPred` *Mixture Effect Predicted by CA at Particular Concentration of a Mixture*

Description

According to the fitted concentration-response information of mixtures. The concentration (e.g., ECx) that causes certain effect in the mixture will be calculated. ecaPred will predict how much effect will be caused at ECx according to concentration addition. The individual concentration-responses should be fitted only based on the following six models: Hill, Weibull, Logit, BCW, BCL, and GL.

Usage

```
ecaPred(effv, sgl, mix, pctMix)
```

Arguments

`effv` numeric vector with single or multiple effect values (0 ~ 1).

`sgl` A list with sgl$model and sgl$param. sgl$model is character vector of equations used to fit the concentration-response data of individual chemicals: Hill, Weibull, Logit, BCW, BCL, GL. sgl$param is numeric matrix of fitting coefficients with rownames (equations) and colnames (Alpha, Beta, and Gamma). For equations with only two parameters, Gamma can be set as zero or any other numeric value.

`mix` A list with mix$model and mix$param. mix$model is character vector of equations used to fit the concentration-response data of mixtures: Hill, Weibull, Logit, BCW, BCL, GL. mix$param is numeric matrix of fitting coefficients of mixtures' concentration-response data with rownames (selected equations) and colnames (Alpha, Beta, and Gamma). For equations with only two parameters, Gamma can be set as zero or any other numeric value.

`pctMix` A numeric matrix, the concentration ratio (percent) of every component in the mixture.

Details

ecaPred calculate the effect concentrations based on the fitted concentration-response information of the mixture according to the input effects effv (e.g., 0.05 and 0.5). The concentration of individual component c_i is computed based on mixture's ECx and the proportion of component in the mixture p_i. Then the CA effect will be calculated based on the concentration addition.

Value

A numeric matrix of predicted effects

See Also

`eiaPred`

Examples

```
## example
# predict the CA predicted response at the concentrations that cause 5%, 10%, 20%, and 50%
# effect of antibiotic mixtures
# each mixture contains eight components. Totally, there are 10 mixtures designed by the udcr.

sgl <- antibiotox$sgl
mix <- antibiotox$udcr.mix
pct <- antibiotox$udcr.pct
ecaPred(effv = c(0.05, 0.1, 0.20, 0.5), sgl, mix, pct)
```

ECx *Effect Concentration Calculation for Sigmoidal Models*

Description

Effect concentrations are calculated at particular effects based on the fitting coefficients of 13 sigmoidal models.

Usage

```
ECx(model, param, effv, rtype = 'quantal', Scaled = TRUE)
```

Arguments

`model` a character vector of equations:("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)")

`param` a numeric matrix of fitting coefficients with rownames (equations) and colnames (Alpha, Beta, Gamma, Delta, and Epsilon).

`effv` a numeric vector with single or multiple effect values

`rtype` the response type of endpoint: 'continuous' or 'quantal' data.

`Scaled` only for 'continuous' data. To indicate if the effv is scaled by response ranges to 0~1 or not (default is TRUE).

Details

effect concentrations will be calculated with provided equations (model), associated fitting parameters (param), and effect levels (effv). For example, $effv$ should be 0.5 if we want to calculate a concentration causes 50% effect.
The inverse functions of the six quantal sigmoidal equations are listed as follows:
inverse Hill_two:

$$c = \beta E / \left(\alpha - E\right)$$

inverse Weibull:

$$c = 10^{(\ln(-\ln(1-E))-\alpha)/\beta}$$

inverse Logit:

$$c = 10^{(\ln(E/(1-E))-\alpha)/\beta}$$

inverse BCW:

$$c = ((\gamma/\beta)(\ln(-\ln(1-E)) - \alpha) + 1)^{1/\gamma}$$

inverse BCL:

$$c = ((\gamma/\beta)(-\ln((1-E)/E) - \alpha) + 1)^{1/\gamma}$$

inverse GL:

$$c = 10^{((-\ln((1/E)^{1/\gamma}-1)-\alpha)/\beta)}$$

where E is effect and c is the concentration.

Value

ecx a numeric vector of effect concentration(s)

effvAbs absolute effect levels. Only for 'continuous' data with scaled effv. The corresponding absolute effect is calculated.

References

Hill equation (biochemistry) http://en.wikipedia.org/wiki/Hill_equation_(biochemistry)
Reference to curveFit

Examples

```
## example 1
# calculate EC5 and EC50 of seven antibiotics on the photobacteria
model <- antibiotox$sgl$model
param <- antibiotox$sgl$param
effv <- c(0.05, 0.5)
ECx(model, param, effv = c(0.05, 0.50))

## example 2
# calculate EC5 and EC50 of four heavy metals and four ionic liquids on the MCF-7 cells
model <- cytotox$sgl$model
param <- cytotox$sgl$param
ECx(model, param, effv = c(0.05, 0.50), rtype = 'quantal')
```

eiaPred *Mixture Effect Predicted by IA at Particular Concentration of a Mixture*

Description

According to the fitted concentration-response information of mixtures. The concentration (e.g., ECx) that causes certain effect in the mixture will be calculated. eiaPred will predict how much effect will be caused at those concentrations according to independent action. The individual concentration-responses should be fitted only based on the following six models: Hill, Weibull, Logit, BCW, BCL, and GL.

Usage

```
eiaPred(effv, sgl, mix, pctMix)
```

Arguments

effv numeric vector with single or multiple effect values (0 ~ 1).

`sgl` A list with sgl$model and sgl$param. sgl$model is character vector of equations used to fit the concentration-response data of individual chemicals: Hill, Weibull, Logit, BCW, BCL, GL. sgl$param is numeric matrix of fitting coefficients with rownames (selected equations) and colnames (Alpha, Beta, and Gamma). For equations with only two parameters, Gamma can be set as zero or any other numeric value.

`mix` A list with mix$model and mix$param. mix$model is character vector of equations used to fit the concentration-response data of mixtures: Hill, Weibull, Logit, BCW, BCL, GL. mix$param is numeric matrix of fitting coefficients of mixtures' concentration-response data with rownames (equations) and colnames (Alpha, Beta, and Gamma). For equations with two parameters, Gamma can be set as zero or any other numeric value.

`pctMix` A numeric matrix, the concentration ratio (percent) of every component in the mixture.

Details

eiaPred calculate the effect concentrations based on the fitted concentration-response information of the mixture according to the input effects effv (e.g., 0.05 and 0.5). The concentration of individual component c_i is computed based on mixture's ECx and the proportion of component in the mixture p_i. Then the IA effect will be calculated based on the independent action.

Value

A numeric matrix of predicted effects

See Also

ecaPred

Examples

```
## example 1
# predict the IA predicted response (cytotoxicity) at the concentrations that cause 10% and 50%
# effect of an mixture.
# each mixture contains eight components. Totally, there are 10 mixtures designed by the udcr.

sgl <- cytotox$sgl
mix <- cytotox$udcr.mix
pct <- cytotox$udcr.pct
eiaPred(effv = c(0.1, 0.5), sgl, mix, pct)

## example 2
# predict the IA predicted response at the concentrations that cause 5% and 50% effect
# of antibiotic mixtures.
# each mixture contains eight components. Totally, there are 2 mixtures designed by the eecr.

sgl <- antibiotox$sgl
mix <- antibiotox$eecr.mix
pct <- antibiotox$eecr.pct
```

```
eiaPred(effv = c(0.05, 0.5), sgl, mix, pct)

## example 2
# predict the IA predicted response at the concentrations that cause 5%, 10%, 20%, and
# 50% effect of antibiotic mixtures.
# each mixture contains eight components. Totally, there are 10 mixtures designed by the udcr.

sgl <- antibiotox$sgl
mix <- antibiotox$udcr.mix
pct <- antibiotox$udcr.pct
eiaPred(effv = c(0.05, 0.10, 0.20, 0.5), sgl, mix, pct)
```

figPlot *Ploting concentration response curve*

Description

Plotting concentration response curves with/without confidence intervals.

Usage

```
figPlot(object, logT = TRUE, xlabel = "concentration (mol/L)", ylabel = "Response",
                ylimit, lgd = NULL)
```

Arguments

`object`	object of class curveFit.
`logT`	logarithm transformation on concentration: TRUE or FALSE(defalt if TRUE).
`xlabel, ylabel`	plot labels.
`ylimit`	range of the y axis.
`lgd`	legend of the plot.

Details

Plot the concentration response curves, experimental data, CI, and PI in one figure.

Examples

```
## example 1
#
x <- antibiotox$PAR$x
expr <- antibiotox$PAR$y
obj <- curveFit(x, expr, eq = 'Logit', rtype = 'quantal', param = c(26, 4), effv = c(0.05, 0.5))
figPlot(obj)

## example 2
#
x <- hormesis$HmimCl$x
```

```
rspn <- hormesis$HmimCl$y
obj <- curveFit(x, rspn, eq = 'Biphasic', param = c(-0.59, 0.001, 160,0.05, 19),
        effv = c(0.05, 0.5), rtype = 'hormesis')
figPlot(obj, logT = TRUE)
```

gcaHill *Mixture Toxicity Prediction Using GCA (Hill_two)*

Description

Predict the mixture toxicity based on individual concentration-response information fitted by Hill_two equation. An explicit formula for gca prediction were used instead of the dichotomy algorithm in gcaPred. Three optional mixture design methods are provided. One is the arbitrary concentration ratio (acr) for mixture components.
Users can deign random ratios for components in the mixture. Other two options are equal effect concentration ratio (eecr) and uniform design concentration ratio (udcr).

Usage

```
gcaHill(model, param, mixType, effv, refEffv = c(0.10, 0.50), rtype)
```

Arguments

`model`	character vector of equation names, just Hill_two
`param`	numeric matrix of fitting coefficients with rownames (equations) and colnames (Alpha, Beta).
`mixType`	experimental design of the mixture. acr: arbitrary concentration ratio; eecr: equal effect concentration ratio; udcr: uniform design concentration ratio.
`effv`	numeric vector with single or multiple (scaled) effect values (0 ~ 1).
`refEffv`	to determine the concentration ranges for predicting effect. Use scaled values (0 ~ 1).
`rtype`	the response type of endpoint: 'continuous' or 'quantal' data.

Details

The generalized concentration addition (GCA) model was proposed by Howard and Webster to predict mixtures containing partial agonists (Howard and Webster, 2009). Empirical data are used to fit concentration-response function, and then predict the mixture response using the inverse function.

$$E_{mix}^{GCA} = \frac{\sum\limits_{i=1}^{n} \frac{\alpha_i c_i}{K_i}}{1 + \sum\limits_{i=1}^{n} \frac{c_i}{K_i}}$$

where c_i is the concentration of component i in the mixture. Parameter α_i and K_i are fitted coefficient of i^{th} component, which are the same as β and α in Hill_two equation. Right, the α_i and K_i are corresponding to β and α in Hill_two equation.

Value

x	a series of concentrations
e	a series of effects caused by the concentrations (x) as predicted by gca
pct	the concentration ratio (percent) of every component in the mixture
uniTab	the uniform design table used to construct the mixture when mixType is udcr

Note

Only for concentration-response curves fitted by the Hill_two equation in curveFit.

References

Howard, G.J., Schlezinger, J.J., Hahn, M.E., Webster, T.F., 2010. Generalized Concentration Addition Predicts Joint Effects of Aryl Hydrocarbon Receptor Agonists with Partial Agonists and Competitive Antagonists. Environ. Health Perspect. 118, 666-672.
Howard, G.J., Webster, T.F., 2009. Generalized concentration addition: A method for examining mixtures containing partial agonists. J. Theor. Biol. 259, 469-477.
Hadrup, N., Taxvig, C., Pedersen, M., Nellemann, C., Hass, U., Vinggaard, A.M., 2013. Concentration addition, independent action and generalized concentration addition models for mixture effect prediction of sex hormone synthesis in vitro. PLoS One 8, e70490.

See Also

gcaPred

Examples

```
model <- c("Hill_two", "Hill_two", "Hill_two", "Hill_two")
param <- matrix(c(3.94e-5, 0.97, 0, 5.16e-4, 1.50, 0, 3.43e-6, 1.04, 0, 9.18e-6, 0.77, 0),
nrow = 4, ncol = 3, byrow = TRUE)
rownames(param) <- c('Ni', 'Zn', 'Cu', 'Mn')
colnames(param) <- c('Alpha', 'Beta', 'Gamma')
## example 1
# using GCA to predict the mixtures designed by equal effect concentration ratio (eecr) at
# the effect concentration of EC05 and EC50
# the eecr mixture design is based on four heavy metals (four factors).
gcaHill(model, param, mixType = "eecr", effv = c(0.05, 0.5), rtype = 'continuous')

## example 2
# using GCA to predict the mixtures designed by uniform design concentration ratio (udcr)
# the udcr mixture design is based on four heavy metals (four factors).
# Seven levels (EC05, EC10, EC15, EC20, EC25, EC30, and EC50 ) are allocated in
# the uniform table
effv <- c(0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.50)
gcaHill(model, param, mixType = "udcr", effv, rtype = 'quantal')

## example 3
# using GCA to predict the mixtures designed by arbitrary concentration ratio (acr)
# the udcr mixture design is based on four heavy metals (four factors).
# the every component in the mixture shares exactly the same ratio (0.25)
```

```
effv <- c(0.25, 0.25, 0.25, 0.25)
gcaHill(model, param, mixType = ”acr”, effv)
```

`gcaPred` *Mixture Toxicity Prediction Using GCA (General)*

Description

Predict the mixture toxicity based on individual concentration-response information. Thirteen monotonic(sigmoidal) models ("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)") are incorporated to construct the GCA equation. The dichotomy technique is used to solve the constructed equation. Three optional mixture design methods are provided. One is the arbitrary concentration ratio (acr) for mixture components. Users can arbitrarily deign a random ratio for each component in the mixture. Other two options are equal effect concentration ratio (eecr) and uniform design concentration ratio (udcr).

Usage

```
gcaPred(model, param, mixType, effv, refEffv, rtype, lb, ub)
```

Arguments

`model` vector of equation names

`param` numeric matrix of fitting coefficients with rownames (equation selected) and colnames (ALpha, Beta, and Gamma). For equations with two parameters, Gamma can be set as zero or any other numeric value.

`mixType` experimental design of the mixture. acr: arbitrary concentration ratio; eecr: equal effect concentration ratio; udcr: uniform design concentration ratio.

`effv` numeric vector with single or multiple (scaled) effect values (0 ~ 1).

`refEffv` to determine the concentration ranges for predicting effect. Use scaled values (0 ~ 1).

`lb` lower bound for solving constructed IA using diochotomy method (default is 1E-9).

`ub` upper bound bound for solving constructed IA using diochotomy method(default is 9).

`rtype` the response type of endpoint: ’continuous’ or ’quantal’ data.

Details

The generalized concentration addition (GCA) model was proposed by Howard and Webster to predict mixtures containing partial agonists (Howard and Webster, 2009).

$$\sum_{i=1}^{n} \frac{c_i}{f_i^{-1}(E)} = 1$$

Empirical data were used to fit concentration-response function, and then predict the mixture response using the inverse function. Previous studies used Hill_two function to fit individual concentration response curves in the GCA prediction (Hadrup et al., 2013; Howard et al., 2010). Here, we incorporated thirteen functions to construct the GCA equation and the dichotomy technique is used to solve the constructed equation.

Value

x	a series of concentrations
e	a series of effects caused by the concentrations (x) as predicted by gca
pct	the concentration ratio (percent) of every component in the mixture
uniTab	the uniform design table used to construct the mixture when mixType is udcr

References

Howard, G.J., Schlezinger, J.J., Hahn, M.E., Webster, T.F., 2010. Generalized Concentration Addition Predicts Joint Effects of Aryl Hydrocarbon Receptor Agonists with Partial Agonists and Competitive Antagonists. Environ. Health Perspect. 118, 666-672.
Howard, G.J., Webster, T.F., 2009. Generalized concentration addition: A method for examining mixtures containing partial agonists. J. Theor. Biol. 259, 469-477.

See Also

gcaHill

Examples

```
## example 1
# using GCA to predict the mixtures designed by equal effect concentration ratio (eecr) at the
# effect concentration of EC50
# the eecr mixture design is based on seven antibiotics(seven factors).
model <- antibiotox$sgl$model
param <- antibiotox$sgl$param
refEffv <- c(0.1, 0.50, 0.80)
gcaPred(model, param, mixType = "eecr", effv = 0.5, refEffv, rtype = 'quantal')

## example 2
# using GCA to predict the mixtures designed by uniform design concentration ratio (udcr)
# the udcr mixture design is based on 2 antibiotics(2 factors) and
# three levels (EC05, EC20, and EC50 )
model <- antibiotox$sgl$model[1 : 2]
param <- antibiotox$sgl$param[1 : 2, ]
effv <- c(0.05, 0.20, 0.50)
refEffv <- c(0.1, 0.80)
gcaPred(model, param, mixType = "udcr", effv, refEffv, rtype = 'quantal')

## example 3
# using GCA to predict the mixtures designed by arbitrary concentration ratio (acr)
# the udcr mixture design is based on 2 heavy metals (2 factors).
# the every component in the mixture shares exactly the same ratio (0.5)
```

```
model <- cytotox$sgl$model[1 : 2]
param <- cytotox$sgl$param[1 : 2, ]
effv <- c(0.5, 0.5)
refEffv <- c(0.1, 0.80)
gcaPred(model, param, mixType = ”acr”, effv, refEffv, rtype = 'quantal')
```

`getCI` *Calculating Confidence Intervals*

Description

calculating non-simultanous confidence intervals and prediction intervals

Usage

```
getCI(object, effv, Scaled = TRUE, sigLev = 0.05)
```

Arguments

`object` object of class curveFit.

`effv` numeric matrix of experimental responses with at least three replicates.

`Scaled` indicating if effv was scaled or not(TRUE/FALSE) in continuous dose-response (rtype = ’continuous’)

`sigLev` significance level(default is 0.05).

Details

The Delta method (Dybowski et al, 2001) is used to construct confidence intervals for predicted responses.

Value

`xmat` effect concentration(s) and corresponding CIs and PIs

`emat` effect(s) and and corresponding CIs and PIs

References

Zhu, X.-W. and Chen, J.-Y. (2016). mixtox: An R Package for Mixture Toxicity Assessment. R Journal, 8(2).
Dybowski, R. and Gant, V. (2001). Clinical applications of artificial neural networks. Cambridge University Press, Cambridge.
Gryze, S. De, Langhans, I., and Vandebroek, M. (2007). Using the correct intervals for prediction: A tutorial on tolerance intervals for ordinary least-squares regression. Chemom. Intell. Lab. Syst., 87, 147-154.

Examples

```
## example 1
x <- cytotox$Ni$x
rspn <- cytotox$Ni$y
obj <- curveFit(x, rspn, eq = 'Logit', param = c(12, 3), effv = c(0.05, 0.5), rtype = 'quantal')
getCI(obj, effv = c(0.05, 0.50))
```

`hormesis` *Non-monotonic Concentration-response Data*

Description

Two ionic liquids: 1-Octyl-3-methylimidazolium chloride (Omim) and 1-Hexyl-3-Methylimidazolium tetrafluoroborate (Hmim). Two organic solvents: Acetonitrile(ACN) and acetone.
The concentration-response data include the effect of these four compounds on firefly luciferase. Those concentration-responses were fitted using non-monotonic models.

Usage

```
data(hormesis)
```

Format

A list with non-monotonic concentration-responses of four chemicals and associated fitting information.

`OmimCl$x` a numeric vector of concentrations

`OmimCl$y` a numeric matrix of responses

`OmimCl$name` name of test substance

`OmimCl$concNum` the number of concentrations

`OmimCl$tierNum` the number of repetitions

`OmimCl$type` type of test substance

`sgl$model` model used to fit the concentration-response data

`sgl$param` fitted coefficients of those curves corresponding to sgl$model

`sgl$minx` numeric vector with multiple concentrations that induce maximun stimulation

`sgl$miny` the largest stimulation

Details

The non-monotonic concentration-responses need to be scaled into [0, 1] using the following equation:

$$E = \frac{I_0 - I_i}{I_0}$$

where I_0 is the average of controls for inhibition test or the average of the maximum effect for the stimulation test and I_i the average effect of the i^{th} treatment.

Source

Eexperiments were conducted in our lab. Detailed descripttion can be found in references

References

Zhu X-W, et.al. 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicol. Environ. Saf. 89:130-136.

Examples

```
#example 1
## Retrive the toxicity data of acetonitrile on firefly luciferase.
hormesis$ACN

#example 2
## Retrive the minx of OmimCl, HmimCl, ACN, and Acetone
hormesis$sgl$minx
```

---

iaPred *Mixture Toxicity Prediction Based on Independent Action*

Description

Predict the mixture toxicity based on individual concentration-response information fitted only based on the following six models: Hill, Weibull, Logit, BCW, BCL, and GL. Three optional mixture design methods are provided:(1)arbitrary concentration ratio (acr), users can set an arbitary proportion for each component in a mixture; (2)equal effect concentration ratio (eecr); (3) uniform design concentration ratio (udcr).

Usage

```
iaPred(model, param, mixType, effv, effPoints, lb = 1e-9, ub = 6)
```

Arguments

`model` character vector of models: Hill, Weibull, Logit, BCW, BCL, GL

`param` numeric matrix of fitting coefficients with row names (selected equations) and column names (Alpha, Beta, and Gamma). For equations with two parameters, Gamma can be set to zero or any other numeric value.

`mixType` experimental design of the mixture. acr: arbitrary concentration ratio; eecr: equal effect concentration ratio; udcr: uniform design concentration ratio.

`effv` numeric vector with single or multiple effect values

`effPoints` numeric vector [0 ~ 1] to predict effect concentrations.

`lb` lower bound for solving constructed IA using diochotomy method.

`ub` upper bound bound for solving constructed IA using diochotomy method.

Details

IA is designed for mixtures of chemicals that have distinct mechanisms of action. The IA model is commonly defined as:

$$\mathrm{E}(c_{mix}) = 1 - (1 - \mathrm{E}(c_1))(1 - \mathrm{E}(c_2)) \cdots (1 - \mathrm{E}(c_n)) = 1 - \prod_{i=1}^{n} (1 - \mathrm{E}(c_i))$$

where $E\left(c_{mix}\right)$ is the overall effect caused by c_{mix}, and $E(c_i)$ is the effect elicited by c_i when applied individually. For a fitted function f_i based on the concentration-response data of the i^{th} component, $E(c_i)$ is equal to $f_i(c_i)$. When $E(c_{mix}) = x$, the equation can be expressed as:

$$x\% = 1 - \prod_{i=1}^{n} \left(1 - f_i\left(p_i\left(EC_{x,mix}\right)\right)\right)$$

This equation can be used to predict the combined effects of mixture-components based on IA. The dichotomy technique is used to solve the constructed equation.

Value

`ia`	a series of effect concentrations predicted by IA
`e`	a series of effects (effPoints) associated with the effect concentrations in ia
`pct`	the proportion of every component in a mixture
`uniTab`	the uniform design table to construct the mixture when mixType is 'udcr'

Note

Note that effv is dependent on the mixType. if the mixType is acr, the length of effv should be the same as that of model. when a total ratio was less than 100 if the mixType is eecr, effv should contain at least one value if the mixTpe is udcr, elements in effv are levels, the length of effv is the same as the number of runs. the number of runs should be in accordance with the length of model based on the uniform design.

References

Liang, Yi-zeng, Kai-tai Fang, and Qing-song Xu. 2001. Uniform Design and Its Applications in Chemistry and Chemical Engineering. Chemometrics and Intelligent Laboratory Systems 58(1):43-57.

Backhaus, T., Faust, M., 2012. Predictive environmental risk assessment of chemical mixtures: A conceptual framework. Environmental Science and Technology. 46, 2564-2573.

See Also

`caPred`

Examples

```
# data(cytotox)

## example 1
# using IA to predict the mixtures designed by equal effect concentration ratio (eecr) at the
# effect concentration of EC05 and EC50
# the eecr mixture design is based on four heavy metals and four ion liquids(eight factors).
model <- cytotox$sgl$model
param <- cytotox$sgl$param
iaPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))

## example 2
# using IA to predict the mixtures designed by uniform design concentration ratio (udcr)
# the udcr mixture design is based on seven antibiotics (seven factors).
# five levels (EC05, EC10, EC20, EC30, and EC50 ) are allocated in the uniform table using the
# pseudo-level technique (Liang et al., 2001)
model <- antibiotox$sgl$model
param <- antibiotox$sgl$param
effv <- c(0.05, 0.05, 0.10, 0.10, 0.20, 0.20, 0.30, 0.30, 0.50, 0.50)
iaPred(model, param, mixType = "udcr", effv)

## example 3
# using IA to predict the mixtures designed by arbitrary concentration ratio (acr)
# the udcr mixture design is based on four antibiotics (four factors).
# the every component in the mixture shares exactly the same ratio (0.25)
model <- antibiotox$sgl$model[1 : 4]
param <- antibiotox$sgl$param[1 : 4, ]
effv <- c(0.25, 0.25, 0.25, 0.25)
iaPred(model, param, mixType = "acr", effv)
```

jacobian *Jacobian Matrix Calculation*

Description

calculating the Jacobian matrix for confidence intervals

Usage

```
jacobian(eq, x, paraHat)
```

Arguments

eq	equation
x	numeric vector of experimental concentrations
paraHat	fitted parameters

Details

Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function.

Value

`jac`	Jacobian matrix.

References

https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

Examples

```
## example 1
x <- cytotox$Ni$x
rspn <- cytotox$Ni$y
obj <- curveFit(x, rspn, eq = 'Logit', param = c(12, 3), effv = c(0.05, 0.5), rtype = 'quantal')
jacobian('Logit', x, obj$p)
```

`mixtox` *Curve Fitting and Mixture Toxicity Assessment*

Description

Curve Fitting for monotonic(sigmoidal) & non-monotonic(J-shaped) concentration-response data. Prediction of mixture toxicity based on reference models such as ’concentration addition’, ’independent action’, and ’generalized concentration addition’.

Details

Package:	mixtox
Type:	Package
Version:	1.3.2
Date:	2017-02-18
License:	GPL-2

(1) Curve fitting of concentration-responses based on 13 monotonic(sigmoidal) and 4 non-monotonic(J-shaped) models, goodness of fit, and calculation of confidence interval .
(2) Experimental design for mixture toxicity. acr: arbitrary concentration ratio; eecr: equal effect concentration ratio; udcr: uniform design concentration ratio.
(3) Mixture toxicity prediction based on reference models such as concentration addition (CA), independent action (IA), and generalized concentration addition (GCA).

Author(s)

xiangwei zhu
Maintainer: xiangwei zhu <xwzhunc@gmail.com>

References

Dunnett, C. W. 1964. New Tables for Multiple Comparisons with a Control. Biometrics 30(3):482-491.
Hickernell, Fred J. 1996. A Generalized Discrepancy and Quadrature Error Bound. Mathematics of Computation 67(211):299-322.
Howard, Gregory J., Jennifer J. Schlezinger, Mark E. Hahn, and Thomas F. Webster. 2010. Generalized Concentration Addition Predicts Joint Effects of Aryl Hydrocarbon Receptor Agonists with Partial Agonists and Competitive Antagonists. Environmental Health Perspectives 118(5):666-672.
Scholze, M. et al. 2001. A General Best-Fit Method for Concentration-Response Curves and the Estimation of Low-Effect Concentrations. Environmental Toxicology and Chemistry 20(2):448-457.
Wang, Yuan and Kai-Tai Fang. 1996. Uniform Design of Experiments with Mixtures. Science in China Series A-Mathematics Physics Astronomy 39(3):264-275.
Backhaus, T., Faust, M., 2012. Predictive environmental risk assessment of chemical mixtures: A conceptual framework. Environmental Science and Technology. 46, 2564-2573.
Zhu X-W, Liu S-S, Qin L-T, Chen F, Liu H-L. 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicology and Environmental Safety 89:130-136.

`nmECx` *Effect Concentration Calculation for J-shaped Models*

Description

Effect concentrations are calculated at particular effects based on the fitting coefficients of J-shaped Models.

Usage

```
nmECx(model, param, effv, minx, gap = -1e-6)
```

Arguments

`model`	a character vector of equations:("Brain_Consens", "BCV", "Biphasic", "Hill_five").
`param`	a numeric matrix of fitting coefficients with rownames (models) and colnames (ALpha, Beta, Gamma, Delta, and Epsilon).
`effv`	a numeric value (vector) with single or multiple effect values (miny ~ 1).
`minx`	a numeric value (vector) with single or multiple concentrations that induce maximun stimulation.
`gap`	theoritical response at the extreme low concentration predicted by a fitted model.

Details

effect concentrations will be calculated with provided equations(model), associated fitting parameters (param), and effects (effv). Effect (effv) should be a value(s) between miny ~ 1. For example, $effv$ should be 0.5 if we want to calculate a concentration causes 50% effect. $minx$ should be calculated by curveFit or tuneFit.

Value

`ecx`	a numeric vector of effect concentration.

References

Zhu X-W, Liu S-S, Qin L-T, Chen F, Liu H-L. 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicology and Environmental Safety 89:130-136.

See Also

`CEx` `curveFit`

Examples

```
## example 1
# calculate ECL-10, ECR-10, EC5, and EC50 of the four hormetic curves
model <- hormesis$sgl$model
param <- hormesis$sgl$param
minx <- hormesis$sgl$minx
nmECx(model, param, effv = c(-0.10, 0.05, 0.50), minx)
```

`NOEC` *NOEC and LOEC Calculation*

Description

calculating the NOEC and LOEC using Dunnett's test

Usage

```
NOEC(x, rspn, blankC = FALSE, sigLev = 0.05, alternertive = 'B')
```

Arguments

`x`	a numeric vector of experimental concentrations
`rspn`	a numeric matrix of experimental responses with at least three replicates.
`blankC`	TRUE if rspn contains responses of blank control. The default is FALSE.
`sigLev`	the significance level for Dunnett's test. The default is 0.05.
`alternertive`	the alternative hypothesis: "U"=upper one-sided test; "B"=two-sided test(default).

Details

Dunnett's test (Dunnett, 1964) is performed to compare the treatment groups with the blank controls. The critical constants (store in DTcv) were calculated using step-down Dunnett test procedure. Three significance level (0.01, 0.05, and 0.1) are supported. ## Q: One dataset has four blank controls (C1, C2, C3, C4) and one treatment has three replicates (T1, T2, T3), ## another treatment has five replicates (R1, R2, R3, R4, R5), how to arrange the response matrix (rspn)? ## A: Label the missing values as NA, the response matrix (rspn) can be arranged as follows:

C1 C2 C3 C4 NA
T1 T2 T3 NA NA
R1 R2 R3 R4 R5

The adjustation of critical value for the unequal variances or unequal number of control and replicates is skipped in this program.

Value

`mat` information on Dunnett's test.DT: Dunnett's test values; DTcv: critical values for Dunnett's test at the significance level of sigLev.

`noec` non-observed effect concentration (NOEC).

`loec` least-observed effect concentration (LOEC).

`sigLev` the significance level used in the Dunnett's test.

`DF` the number of treatments and degree of fredom.

Note

x a vector of concentrations or levels in an ascending order. response matrix with at least 3 replicates. if the response matrix (rspn) contains blank controls (blankC = TRUE), the blank controls should be allocated in the first tow of rspn matrix. missing values should be labled as NA.

References

Dunnett, C.W., 1964. New tables for multiple comparisons with a control. Biometrics 30, 482-491.

Examples

```
## example 1
# calcualte the NOEC and LOEC of heavy metal Ni(2+) on the MCF-7 cells at the default significance
# level of 0.05
x <- cytotox$Ni$x
rspn <- cytotox$Ni$y
NOEC(x, rspn)

## example 2
# calcualte the NOEC and LOEC of Neomycin sulfate on the phtotobacteria at the significance
# level of 0.01
x <- antibiotox$NEO$x
rspn <- antibiotox$NEO$y
NOEC(x, rspn, sigLev = 0.01)
```

qq4res *Residual Normal QQ Plot*

Description

Producing a side-by-side QQ plot of the residuals against standard normal quantiles.

Usage

```
qq4res(object, xlabel = 'Theoretical Quantiles', ylabel = 'Residuals',
       lgd = NULL)
```

Arguments

`object`	object of class curveFit.
`xlabel, ylabel`	plot labels.
`lgd`	legend of the plot

Details

The empirical quantiles are plotted against the quantiles of a standard normal distribution. If the residuals are from a normal distribution with mean 0, the points tend to fall along the reference line that has an intercept of 0 and a slope equal to the estimated standard deviation.

Examples

```
## example 1
#
x <- antibiotox$PAR$x
expr <- antibiotox$PAR$y
obj <- curveFit(x, expr, eq = 'Logit', rtype = 'quantal', param = c(26, 4), effv = c(0.05, 0.5))
qq4res(obj)
```

showEq *List Requested Equations*

Description

Show the formula of different equations upto request.

Usage

```
showEq(eq)
```

Arguments

eq equation name to qury

Details

Thirteen monotonic(sigmoidal) equations ("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)") and four non-monotonic(J-shaped) equations ("Brain_Consens", "BCV", "Biphasic", "Hill_five") are provided to fit concentration-response data.

Value

The formula of requested equations (with abbr.) will show up.

References

Scholze, M. et al. 2001. A General Best-Fit Method for Concentration-Response Curves and the Estimation of Low-Effect Concentrations. Environmental Toxicology and Chemistry 20(2):448-457.
Zhu X-W, et.al . 2013. Modeling non-monotonic dose-response relationships: Model evaluation and hormetic quantities exploration. Ecotoxicol. Environ. Saf. 89:130-136.
Howard GJ, Webster TF. 2009. Generalized concentration addition: A method for examining mixtures containing partial agonists. J. Theor. Biol. 259:469-477.
Spiess, A.-N., Neumeyer, N., 2010. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: A Monte Carlo approach. BMC Pharmacol. 10, 11.
Di Veroli GY, Fornari C, Goldlust I, Mills G, Koh SB, Bramhall JL, et al. 2015. An automated fitting procedure and software for dose-response curves with multiphasic features. Scitific Report 5: 14701.
Gryze, S. De, Langhans, I., Vandebroek, M., 2007. Using the correct intervals for prediction: A tutorial on tolerance intervals for ordinary least-squares regression. Chemom. Intell. Lab. Syst. 87, 147-154.

See Also

curveFit

Examples

```
# example 1
## show Weibull model
showEq('Weibull')

# example 2
## show the name of all sigmoidal models
showEq('sigmoid')
```

staval *Starting Values for 13 Sigmoidal and 4 Hormetic Models*

Description

providing staring values for function tuneFit.

Usage

```
data(staval)
```

Format

A list with starting values for 13 sigmoidal models and six hormetic models

`staval$Hill` sarting values for Hill model

`staval$GL` sarting values for generalized logit model

`staval` starting values for 13 sigmoidal and five hormetic models

Details

Thirteen monotonic(sigmoidal) models ("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)") and four non-monotonic(J-shaped) models ("Brain_Consens", "BCV", "Biphasic", "Hill_five")

Examples

```
# example 1
## Retrive the starting values for Hill.
staval$Hill

# example 2
## Retrive the starting values for Weibull.
staval$Weibull
```

tuneFit *Find Optimal Starting values for Curve Fitting*

Description

Curve fitting is dependent on the package minpack.lm. This generic function first seaches optimal starting values based on trial and error. The the concentration response data will be fitted using the optimal starting values. The statistics for goodness of fit is evaluated by the following statistics: coefficient of determination (R^2), adjusted coefficient of determination (R^2_{adj}), root mean squared error (RMSE), mean absolute error (MAE), Akaike information criterion (AIC), bias-corrected Akaike information criterion(AICc), and Bayesian information criterion (BIC). Thirteen sigmoidal models ("Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW(Box-Cox-Weibull)", "BCL(Box-Cox-Logit)", "GL(Generalized Logit)") and four J-shaped models ("Brain_Consens", "BCV", "Biphasic", "Hill_five") are provided to fit the concentration-response data.

Usage

```
tuneFit(conc, rspn, eq = 'Weibull', effv, rtype = 'quantal', rsq = 0.6, highBar = 5000,
bar = 1000, sav = FALSE)
```

Arguments

conc	a numeric vector (matrix) of experimental concentrations
rspn	a numeric vector (matrix) of responses corresponding to conc, it should have the same length (or rows and columns for matrix) as conc.
eq	models for curve fitting: "Hill", "Hill_two", "Hill_three", "Hill_four", "Weibull", "Weibull_three", "Weibull_four", "Logit", "Logit_three", "Logit_four", "BCW", "BCL", "GL", "Brain_Consens", "BCV", "Biphasic", "Hill_five".
rtype	the response type of endpoint: 'continuous' or 'quantal' data.
effv	numeric response to calculate effect concentration, scaled respones [0, 1].
rsq	r2 below which would be ignored.
highBar	if the number of starting values exceeds highBar, a random sample of starting values will be taken.
bar	the number of random samples.
sav	save fitting results in a text file (TRUE/FALSE).

Details

tuneFit provides high frequency trial and error approach to find appropriate starting values for users. It will deploy those starting values one by one until finding the right one.
Function tuneFit can also be used to fit the concentration response data for a batch of chemicals and calculate corresponding effect concentration.

Value

`sta` goodness of fit statistics: (R^2, R^2_{adj}, MAE, RMSE, AIC, AICc, and BIC)

Note

tuneFit will load the file staval.rda which contains hundreds of starting values for each of the sigmoidal and hormetic models. However, those starting values are also limited. Users are encouraged to send their fitted coefficients to us to extent the coverage of staval.

Examples

```
## example 1
# Fit the non-monotonic concentration-response data
# we'll get a fit with r2 of 0.740
x <- hormesis$OmimCl$x
expr <- hormesis$OmimCl$y
y <- rowMeans(expr)
tuneFit(x, y, eq = 'Biphasic')

## example 2
# Fit the non-monotonic concentration-response data
# use r2 (rsq) of 0.9, we'll get a fit with r2 of 0.989
# calcualte the effect concentration that causes 5% inhibition
x <- hormesis$OmimCl$x
expr <- hormesis$OmimCl$y
y <- rowMeans(expr)
tuneFit(x, y, eq = 'Biphasic', effv = 0.05, rsq = 0.9)

## example 3
# Fit the concentration-response data of heavy metal Ni(2+) on MCF-7 cells.
# Calculate the concentration that causes 5% inhibition on the growth of MCF-7

x <- cytotox$Ni$x
expr <- cytotox$Ni$y
y <- rowMeans(expr)
tuneFit(x, y, eq = 'Logit', effv = 0.05)

## example 4
# Fit the concentration-response data of Paromomycin Sulfate (PAR) on photobacteria.
# Calculate the concentrations that cause 50% inhibition on the growth of photobacteria

x <- antibiotox$PAR$x
expr <- antibiotox$PAR$y
y <- rowMeans(expr)
tuneFit(x, y, eq = 'Logit', effv = 0.5)
```

unidTab *Uniform Design Table*

Description

The good lattice point method with a power generator was used to construct the uniform experimental tables . The centered L2-discrepancy (CD2) and the symmetric discrepancy algorithms (sd2) were employed to measure the uniformity and find the one with lowest discrepancy.

Usage

```
unidTab(lev, fac, algo = "cd2")
```

Arguments

`lev`	the number of runs (levels or pseudo-levels)
`fac`	the number of factors
`algo`	algorithms used to calculate the discrepancy. "cd2": the centered L2-discrepancy algorithm. "sd2": the symmetric discrepancy algorithm.

Details

Uniform design (UD) was proposed by Fang et al (Acta Math Appl Sin 3:363-372 (1980)). An appropriate uniform design table is constructed according to the factor (the number mixture components) and level (the number of experiments need to run). Many methods can be used to construct the uniform table. In the past decades many methods have been proposed for constructing (nearly) uniform designs, such as the good lattice point (glp) method, the glp method with a power generator (pglp method) (Fang 1980; Fang andWang 1994), the cutting method (Maand Fang 2004), the optimization method (Winker and Fang 1998).
However, when s is large, the glp method has a large computational cost. And the pglp method has the lowest computation complexity among various methods in quasi-Monte Carlo methods and a good performance when n or $n+1$ is a prime number and s is small (Fang 1980; Fang and Wang 1994), while the pglp method may have a poor performance when s is large. Here, we choose the glp method with a power generator to construct the uniform table. The centered L2-discrepancy (cd2) is set as default over the symmetric discrepancy algorithm for its accuracy. The cd2 algorithm is defined as follows:

$$CD_2(P) = \left[\left(\frac{13}{12}\right)^s - \frac{2^{1-s}}{n}\sum_{k=1}^{n}\prod_{i=1}^{s}\theta_{ki} + \frac{1}{n^2}\sum_{k,l=1}^{n}\prod_{i=1}^{s}\phi_{k,li}\right]^{\frac{1}{2}}$$

With the definition of θ_{ki} and $\phi_{k,li}$ as follows:

$$\theta_{ki} = 2 + \left|x_{ki} - \frac{1}{2}\right| - \left|x_{ki} - \frac{1}{2}\right|^2$$

$$\phi_{k,li} = 1 + \frac{1}{2}\left(\left|x_{ki} - \frac{1}{2}\right| + \left|x_{li} - \frac{1}{2}\right| - |x_{ki} - x_{li}|\right)$$

where n, s are the number of runs (levels or multiple of levels) and the number of input variables (factors), respectively.

Value

T	the selected uniform table(s) . It may contain one or more uniform tables with the same discrepancy
D	the discrepancy of the constructed uniform table

References

Wang, Y., Fang, K.-T., 1996. Uniform design of experiments with mixtures. Sci. China Ser. A-Mathematics Phys. Astron. 39, 264-275.
Hickernell, F.J., 1996. A generalized discrepancy and quadrature error bound. Math. Comput. 67, 299-322.

Examples

```
## example 1
# construct uniform table with 11 runs and 7 factors using the default centered L2-discrepancy
# algorithm
unidTab(11, 7)

## example 2
# construct uniform table with 37 runs and 13 factors using the symmetric discrepancy algorithm
unidTab(lev = 37, fac = 13, algo = "sd2" )

## example 3
# construct uniform table with 37 runs and 13 factors using default centered L2-discrepancy
# algorithm
unidTab(lev = 37, fac = 13, algo = "cd2" )
```

Index

∗Topic **BMDL**
BMD, 3
∗Topic **BMD**
BMD, 3
∗Topic **Hill_two**
gcaHill, 20
∗Topic **Jacobian matrix**
jacobian, 28
∗Topic **LOEC**
NOEC, 31
∗Topic **NOEC**
NOEC, 31
∗Topic **arbitrary concentration ratio**
caPred, 5
gcaHill, 20
gcaPred, 22
iaPred, 26
∗Topic **concentration addition**
caPred, 5
ecaPred, 14
mixtox, 29
∗Topic **concentration response curve**
figPlot, 19
∗Topic **concentration-response curve**
ECx, 16
∗Topic **continuous dose-reponses**
curveFit, 8
∗Topic **continuous responses**
antibiotox, 2
cytotox, 12
∗Topic **critical value**
DTcv, 14
∗Topic **curve fitting**
curveFit, 8
mixtox, 29
staval, 35
tuneFit, 36
∗Topic **cytotoxicity**
cytotox, 12
∗Topic **degree of freedom**
DTcv, 14
∗Topic **dose-response curve**
curveFit, 8
∗Topic **effect concentration**
CEx, 7
curveFit, 8
ECx, 16
nmECx, 30
tuneFit, 36
∗Topic **equal effect concentration ratio**
caPred, 5
gcaHill, 20
gcaPred, 22
iaPred, 26
∗Topic **generalized concentration addition**
gcaHill, 20
gcaPred, 22
mixtox, 29
∗Topic **good lattice point**
unidTab, 37
∗Topic **goodness of fit**
curveFit, 8
mixtox, 29
tuneFit, 36
∗Topic **hormesis dose-reponses**
curveFit, 8
∗Topic **hormesis**
hormesis, 25
∗Topic **independent action**
eiaPred, 17
iaPred, 26
mixtox, 29
∗Topic **inverse function**
CEx, 7
∗Topic **mixture effect**
caPred, 5

ecaPred, 14
eiaPred, 17
∗Topic **monotonic and non-monotonic equations**
showEq, 33
∗Topic **non-monotonic concentration-response data**
hormesis, 25
∗Topic **non-monotonic curve**
nmECx, 30
∗Topic **non-simultanous confidence intervals**
getCI, 24
∗Topic **non-simultanous prediction intervals**
getCI, 24
∗Topic **normal QQ plot**
qq4res, 33
∗Topic **phtotobacteria**
antibiotox, 2
∗Topic **quantal dose-reponses**
curveFit, 8
∗Topic **quantal responses**
antibiotox, 2
cytotox, 12
∗Topic **response range**
curveFit, 8
∗Topic **significance level**
DTcv, 14
∗Topic **starting values**
staval, 35
∗Topic **trial and error**
tuneFit, 36
∗Topic **uniform design concentration ratio**
caPred, 5
gcaHill, 20
gcaPred, 22
iaPred, 26
∗Topic **uniform design table**
caPred, 5
gcaHill, 20
gcaPred, 22
iaPred, 26
∗Topic **uniform design**
unidTab, 37

antibiotox, 2

BMD, 3

caPred, 5, *27*
CEx, 7, *31*
curveFit, 8, *31*, *34*
cytotox, 12

DTcv, 14

ecaPred, 14, *18*
ECx, 16
eiaPred, *15*, 17

figPlot, 19

gcaHill, 20, *23*
gcaPred, *21*, 22
getCI, 24

hormesis, 25

iaPred, *6*, 26

jacobian, 28

mixtox, 29

nmECx, 30
NOEC, 31

qq4res, 33

showEq, 33
staval, 35

tuneFit, 36

unidTab, 37