mixtox: An R Package for Mixture Toxicity Assessment

by Xiang-Wei Zhu and Jian-Yi Chen

Abstract Mixture toxicity assessment is indeed necessary for humans and ecosystems that are continually exposed to a variety of chemical mixtures. This paper describes an R package, called **mixtox**, which offers a general framework of curve fitting, mixture experimental design, and mixture toxicity prediction for practitioners in toxicology. The unique features of **mixtox** include: (1) constructing a uniform table for mixture experimental design; and (2) predicting toxicity of a mixture with multiple components based on reference models such as concentration addition, independent action, and generalized concentration addition. We describe the various functions of the package and provide examples to illustrate their use and show the collaboration of **mixtox** with other existing packages (e.g., **drc**) in predicting toxicity of chemical mixtures.

Introduction

Environmental chemicals usually occur as complex mixtures rather than a single compound. Humans and ecosystems are continually exposed to a variety of chemical mixtures with changing composition under different circumstances. Public concern over environmental chemicals demands extensive risk assessment of various mixtures. Today, risk assessment of the effect of environmental mixtures is mainly based on a complex framework in the context of reference models (Backhaus and Faust, 2012). Predicting toxicity of mixtures using reference models is usually based on the fitting information of individual concentration response curves (Faust et al., 2003, 2001). Knowledge on the synergistic, antagonistic and additive effects of multiple stressors will be helpful for regulatory agencies in the risk assessment of environmental chemicals.

Curve fitting of individual concentration responses is the basis for predicting the effect of mixtures. Many sigmoidal regression functions have been proposed for the fitting of monotonic nonlinear concentration response data (Goutelle et al., 2008; Scholze et al., 2001; Spiess and Neumeyer, 2010). Sigmoidal functions (e.g., Logit and Weibull) with a lower limit of 0 and an upper limit of 1 (Scholze et al., 2001) are suitable for quantal response data. Other functions like three- or four-parameter Hill functions (Goutelle et al., 2008) and three- or four-parameter Weibull and Logistic functions (Spiess and Neumeyer, 2010) could be used for both quantal and continuous response data. Both proprietary (e.g., Origin) and free-to-download software (e.g., EPA Probit analysis program and BMDS; https://www.epa.gov/bmds) are available for curve fitting. Focusing on R packages we must mention **drc** (Ritz and Streibig, 2005), **drfit** (Ranke, 2016), and **ezec** (Kamvar, 2016). Package **drc** provides a suite of flexible and versatile model fitting and after-fitting functions to analyze concentration response data.

Appropriate mixture experiments need to be designed to analyze interactions (synergistic, antagonistic, or additive) between/among mixture components. Usually, fixed-ratio ray design is employed to systematically measure the effect of mixtures. One popular fixed-ratio ray design is the equal effect concentration ratio (EECR) (Faust et al., 2003, 2001), which designs mixtures according to the proportion of a particular effect concentration, say half maximal effect concentration (EC50), of individual chemicals. EECR has become a de facto standard for the experimental investigation of chemical interaction for all kind of mixtures. The call for simulating "environmentally realistic" mixtures (Backhaus and Faust, 2012) requires more sophisticated experimental design for mixtures. The uniform design concentration ratio (UDCR, or uniform design ray; Liu et al. 2015) was proposed to construct the "environmentally realistic" mixtures. Constructing appropriate uniform design tables (Ning et al., 2011) is the key to the sophisticated uniform experimental design. Uniform design tables with various number of runs, factors, and levels calculated by Professor Fang and his colleagues can be found on the website http://sites.stat.psu.edu/~rli/DMCE/UniformDesign/. However, those tables are not specifically tuned for mixture experimental design. The APTox program (Liu et al., 2012), assessment and prediction on toxicity of chemical mixtures, developed based on Visual Basic, provides a module to construct uniform design tables. To the best of our information, there is no R package available for the construction of uniform design table.

Concentration addition (CA) and independent action (IA) are two classical reference concepts that allow predicting toxicity of mixture substances (Backhaus and Faust, 2012). CA assumes to predict toxicity of mixture compounds with a similar mechanism of action. In contrast to CA, IA assumes mixture components act on different biological targets of an exposed organism independently. The

expected effect of mixtures can hence be calculated according to the joint probability of statistically independent events. The package **drc** (Ritz and Streibig, 2005, 2014) provides fitting models (e.g., CA, Hewlett, and Voelund) incorporating synergism/antagonism for binary and ternary mixtures. However, there is a need for evaluating the effect of mixtures with more than three components. The CA and IA models are suitable for fitting mixtures with quantal concentration response data (or quantal responses after proper transformation). Generalized concentration addition (GCA) (Howard and Webster, 2009) was proposed to examine mixtures containing partial agonists. GCA is appropriate for both quantal and continuous concentration response relationships. A following study (Hadrup et al., 2013) showed GCA could predict a larger range of the concentration response curve as compared with the CA and IA models. GCA is appropriate for both quantal and continuous concentration response relationships.

The goal of this paper is to describe an R package, called **mixtox** (Zhu, 2016), that allows to perform curve fitting, experimental design, and mixture toxicity assessment. In this paper, we will introduce curve fitting of individual concentration response data using a series of sigmoidal models, mixture experimental design based on different strategies, and mixture toxicity prediction based on CA, IA, and GCA, as well as follow-up analysis.

The paper is organized as follows. First, the statistical fundamentals underneath the curve fitting, experimental design, and mixture toxicity prediction are briefly recalled. Then the main functions in **mixtox** are listed and their use described. Finally, several examples are provided to illustrate the use of these functions.

Outline of statistical fundamentals

Consider a set of n concentration response data points (x_1, y_1), (x_2, y_2), $\cdots$ (x_n, y_n) and model function $y = f(x, \boldsymbol{\theta}) + \mathbf{e}$ with p parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_p)$. A non-linear least square regression was used to fit the concentration response data. The purpose of non-linear least square regression is to find the parameter θ to minimize the sum of squares of $\mathbf{e}$ (a vector of errors). With the help of the modified Levenberg-Marquardt algorithm, package **minpack.lm** (Elzhov et al., 2016) is more efficient than the built-in nls function in solving nonlinear least-squares problems. It was employed to fit the concentration response data instead of nls.

Goodness of fit statistics

The **mixtox** package provides 13 sigmoidal functions to fit concentration response data. A series of goodness of fit statistics were provided to select the best fit. These statistics include coefficient of determination (R^2), the bias-corrected coefficient of determination (R^2_{adj}), root mean squared error ($RMSE$), mean absolute error (MAE), Akaike information criterion (AIC), the bias-corrected $AICc$, and Bayesian information criterion (BIC). The mathematical expression of these statistics can be found in Liao and McGee (2003) and Spiess and Neumeyer (2010). A variety of functions can be used to fit the concentration responses of one compound on an organism. Previously, the best model was selected based on the highest R^2 and R^2_{adj}, lowest $RMSE$, and lowest MAE. However, R^2 or even R^2_{adj} are not sensitive measures for nonlinear models as compared with AIC, $AICc$, and BIC (Spiess and Neumeyer, 2010). Here, we suggest that users select the best model based on the lowest AIC, $AICc$, and BIC.

Confidence intervals

The Delta method (Dybowski and Gant, 2001) is used to construct confidence intervals for predicted responses. The (non-simultaneous) confidence interval (CI) at the predictor value x is given as follows:

$$CI = f(\hat{\boldsymbol{\theta}}) \pm t_{(n-p, (1-\alpha)/2)} \sqrt{\left(v^T Var(\hat{\boldsymbol{\theta}}) v \right)}, \tag{1}$$

where t is the inverse of Student's t cumulative distribution function, α is the confidence level (usually 95%); the superscript T denotes transpose; v is the row vector of the Jacobian evaluated at a specified predictor value.

$Var(\hat{\boldsymbol{\theta}})$ in (1) is the covariance matrix of the parameter estimates.

$$Var(\hat{\boldsymbol{\theta}}) = \sigma^2 \left(J(\boldsymbol{\theta})^T J(\boldsymbol{\theta}) \right)^{-1}, \tag{2}$$

where σ^2 is the squared residual standard error. $J(\theta)$ is the Jacobian matrix of $f(\theta)$ with respect to the coefficients.

The (non-simultaneous) prediction bounds for a new observation (Huet et al., 2004) (i.e., prediction interval, *PI*) were found to show better characterization on responses at the extremely low concentrations (Zhu et al., 2013). The *PI* (De Gryze et al., 2007; Huet et al., 2004) was recommended as the primary choice for uncertainty characterization:

$$PI = f(\hat{\theta}) \pm t_{(n-m,\,(1-\alpha)/2)} \sqrt{\sigma^2 + \left(\nu^T Var(\hat{\theta})\nu\right)}. \tag{3}$$

The *PI* for new observations should be wider than the *CI* for the additional variance (σ^2) in predicting new responses (the fit plus random errors).

Uniform design table

Many experimental design methods such as generalized Latin hypercube design (Dette and Pepelyshev, 2010) and uniform design (Ning et al., 2011) can be used to construct chemical mixtures. The uniform design was incorporated into package **mixtox**. Uniform design is an efficient experimental design method to simulate environmentally realistic mixtures. It allows researchers to investigate mixtures with more chemicals (i.e., factors in uniform design) and concentrations (levels) while simultaneously minimizing the number of experiments (Liu et al., 2015).

As a prerequisite for constructing a uniform design table, researchers need to plan the number of runs, factors, and levels for a mixture experiment according to their experiment. Usually, the number of runs is an integer multiple of levels. Assume that we want to test the mixture effect of three compounds each with four different concentrations. Based on the three factors (compounds) and four levels (concentrations), a uniform design table with four runs would be suitable for this case. If we want to use a uniform design table with eight runs, then the levels (concentrations) need to be allocated in repetition to form pseudo-levels (Liang et al., 2001), the number of which equals that of runs.

The good lattice point method (Zhou and Fang, 2013) with a power generator was used to construct a uniform table based on the number of runs. The centered L_2-discrepancy (CD_2) (Hickernell, 1996; Zhou and Fang, 2013) was employed to measure the uniformity and find the one with lowest discrepancy.

$$CD_2(P) = \left[\left(\frac{13}{12}\right)^s - \frac{2^{1-s}}{n} \sum_{k=1}^{n} \prod_{i=1}^{s} \theta_{ki} + \frac{1}{n^2} \sum_{k,l=1}^{n} \prod_{i=1}^{s} \phi_{k,li} \right]^{1/2}, \tag{4}$$

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

$$\theta_{ki} = 2 + \left|x_{ki} - \frac{1}{2}\right| - \left|x_{ki} - \frac{1}{2}\right|^2, \tag{5}$$

$$\phi_{k,li} = 1 + \frac{1}{2}\left(\left|x_{ki} - \frac{1}{2}\right| + \left|x_{li} - \frac{1}{2}\right| - \left|x_{ki} - x_{li}\right|\right), \tag{6}$$

and where n and s are the number of runs (levels or pseudo-levels) and the number of factors, respectively.

Reference models

For a well-defined mixture of n components, concentration addition is expressed mathematically as (Faust et al., 2001):

$$\sum_{i=1}^{n} \frac{c_i}{ECx_i} = 1, \tag{7}$$

where ECx_i is the effect concentration of the ith component that causes an $x\%$ effect when applied individually at c_i. The c_i is expressed as:

$$c_i = p_i \cdot c_{mix} = p_i \cdot EC_{x,mix}, \tag{8}$$

where p_i is the proportion of the ith component in a mixture, c_{mix} the concentration of the mixture and $EC_{x,mix}$ the concentration of the mixture that causes an effect of $x\%$. The formula to predict the effect

of a mixture can be expressed as:

$$EC_{x,mix} = \left(\sum_{i=1}^{n} \frac{p_i}{EC_{x,i}} \right)^{-1}. \tag{9}$$

The theoretical basis for independent action is defined as (Faust et al., 2001):

$$E(c_{mix}) = 1 - \left(1 - E(c_1)\right)\left(1 - E(c_2)\right) \cdots \left(1 - E(c_n)\right) = 1 - \prod_{i=1}^{n} \left(1 - E(c_i)\right), \tag{10}$$

where $E(c_{mix})$ is the total effect caused by a mixture at the concentration of c_{mix}, and $E(c_i)$ is the effect caused at c_i of individual chemicals. For a fitted function (f_i) based on the concentration response data of the ith component, $E(c_i)$ is equal to $f_i(c_i)$. When $E(c_{mix}) = x$, (10) can be expressed as:

$$x\% = 1 - \prod_{i=1}^{n} \left(1 - f_i\left(p_i(EC_{x,mix})\right)\right). \tag{11}$$

(11) can be used to predict the mixture effect for IA.

Generalized concentration addition (GCA) is a natural extension of CA that could be applied to mixtures including full agonists and partial agonists or mixtures including full agonists and competitive antagonists (Howard and Webster, 2009). The general form of GCA is expressed as follows:

$$\sum_{i=1}^{n} \frac{c_i}{f_i^{-1}(E)} = 1. \tag{12}$$

Empirical data were used to fit function $f_i(c_i)$, and then predict the mixture response using the inverse function $f_i^{-1}(E)$. Previous studies used the Hill function with Hill coefficient equal to 1 to fit individual concentration response curves (Howard et al., 2010). It is suitable to fit response data with lower limit fixed at 0 and no fixed maximal effect.

$$f(c) = \frac{\alpha c}{K + c}, \tag{13}$$

where c is the concentration, K is the concentration causes 50% effect, and α is the maximal effect level of a chemical on a test organism. In **mixtox**, function (13) was labeled as Hill_two. The equation to predict the effect of a multiple-component mixture based on Hill_two is expressed as follows (Howard and Webster, 2009):

$$E_{mix}^{GCA} = \left(\sum_{i=1}^{n} \frac{\alpha_i c_i}{K_i} \right) \Big/ \left(1 + \sum_{i=1}^{n} \frac{c_i}{K_i} \right). \tag{14}$$

Overview of the package

This section provides a description of functions in the R package **mixtox**. We first detail the main functions that allow to fit concentration response data. Second, we describe a function to construct uniform design tables. Then we discuss functions for mixture toxicity prediction.

Concentration-response curve fitting

Package **mixtox** provides 13 sigmoidal functions to fit monotonic concentration responses. First, the name of 13 sigmoidal functions will display through the command showEq("sigmoid"):

```
[1] "Hill"                "Hill_two"                "Hill_three"
[4] "Hill_four"           "Weibull"                 "Weibull_three"
[7] "Weibull_four"        "Logit"                   "Logit_three"
[10] "Logit_four"         "BCW(Box-Cox-Weibull)"    "BCL(Box-Cox-Logit)"
[13] "GL(Generalized Logit)"
```

Then the formula of those functions can be displayed through querying the function name using showEq (e.g., showEq("Hill")). Six functions (i.e., Hill, Weibull, Logit, BCW, BCL, and GL) are suitable to fit quantal responses in the range of $[0, 1]$. The other 7 functions are suitable to fit continuous concentration responses with response range out of $[0, 1]$.

The main function for curve fitting in **mixtox** is curveFit. This function provides various options. It requires concentration (x), experimental responses (expr), a suitable equation (eq), and corresponding

starting values (param).

```
R> curveFit <- function(x, expr, eq , param, effv, sigLev = 0.05)
```

The eq term can be any of the 13 sigmoidal equations. The default significant level (sigLev) is 0.05 to calculate the *PI* and *CI*. Dunnett test (Dunnett, 1964) was used to calculate the non-observed effect concentration (NOEC) and lowest observed effect concentration (LOEC) for quantal concentration response data or response data with lower limit fixed at 0. The response values for the control in those data were set to 0 in the calculation of NOEC and LOEC.

```
R> NOEC <- function(x, expr, sigLev = 0.05)
```

The starting values (param) are indispensable for curveFit. The function tuneFit could help users to find proper starting values. Users need to provide their experimental concentration (conc), corresponding responses (rspn) and the equation (eq) they choose to fit the data (e.g., Weibull). Other terms (effv, rsq, highBar, bar, and sav) are optional and the detailed explanation can be found in the reference manual.

```
R> tuneFit <- function(conc, rspn, eq = "Weibull", effv, rsq = 0.6, highBar = 5000,
+    bar = 1000, sav = FALSE)
```

tuneFit is essentially a sophisticated wrapper for nlsLM in the package **minpack.lm**. Like the function drm in package **drc** (Ritz and Streibig, 2005) which provides self starting values, tuneFit also provides starting values for all of the 13 sigmoidal functions. Those starting values were collected through the fitting of various concentration response data from various sources (e.g., our lab tests, published literature, and large databases such as ToxCast; Kavlock et al. 2012). The starting values for all of the 13 functions are stored in staval (e.g., staval$Logit would show all of the 1786 starting values for the Logit function). tuneFit provides a high frequency trial and error approach to deploy those starting values one by one until getting the best fit. This approach is quite efficient especially for the quantal concentration response data.

Construction of a uniform design table

The function unidTab can be called to generate uniform design tables.

```
R> unidTab <- function(lev, fac, algo = "cd2")
```

To call unidTab, users need to provide the number of runs (levels, lev) and factors (fac). Normally, the number of runs equals the number of levels. If users need to do more runs than levels (the number of runs is n times that of levels, $n = 1, 2, \cdots$), a pseudo-level design needs to be performed. Usually, unidTab is incorporated into functions (e.g., caPred and iaPred) to predict toxicity of mixtures. Once users provide the levels (pseudo-levels) and fitting information of individual compounds, the unidTab will be called by caPred and iaPred to generate uniform table(s) with the lowest discrepancy.

Mixture toxicity prediction

The function caPred can be called to predict the effect of mixtures based on concentration addition.

```
R> caPred <- function(model, param, mixType = c("acr", "eecr", "udcr"), effv,
+    effPoints)
```

Mixture toxicity prediction is based on the fitting information of individual concentration response curves. Users need to fit the individual concentration response data first and provide the fitting information (model used to fit individual concentration responses and corresponding fitted parameters param) to caPred. Function caPred provides three optional fix-ratio ray design methods: (1) arbitrary concentration ratio (acr), users can arbitrarily define the proportion of a component in a mixture. It also allows users to design mixtures according to certain experimental design methods (e.g., Latin hypercube design; Dette and Pepelyshev 2010); (2) equal effect concentration ratio (eecr); and (3) uniform design concentration ratio (udcr). If the mixture type is acr, the term effv is a vector of ratios or concentrations that will eventually be converted into the proportion of mixture components. If the mixture type is eecr or udcr, the argument effv is a numeric vector with single or multiple effect values. The argument effPoints is a vector of effect values (with a range of $[0, 1]$).

The function iaPred can be called to predict the effect of mixtures based on IA.

```
R> iaPred <- function(model, param, mixType = c("acr", "eecr", "udcr"), effv,
+    effPoints, lb = 1e-9, ub = 6)
```

Most of the arguments in `iaPred` are the same as those in `caPred`. The arguments of `lb` and `ub` are the lower and upper bounds of the concentration range where to find a solution for the constructed IA equation based on (11), respectively. Users can change these values based on their practical experimental system.

The function `caPred` can only predict mixture effects of chemicals with quantal concentration responses. That is, `caPred` only covers concentration response curves fitted by six functions (i.e., Hill, Weibull, Logit, BCW, BCL, and GL).

The function `gcaHill` can be used to predict the effect of mixtures based on (14) on the condition that all individual concentration responses should be fitted by the function Hill_two.

```
R> gcaHill <- function(model, param, mixType = c("acr", "eecr", "udcr"), effv,
+    refEffv = c(0.10, 0.50))
```

The reference effects (`refEffv`) need to be provided by users to define the range of responses. To extend the GCA to be more general, 12 more functions (Hill, Hill_three, Hill_four, Weibull, Weibull_three, Weibull_four, Logit, Logit_three, Logit_four, BCW, BCL, and GL) were incorporated to `gcaPred` according to the general form of GCA equation in (12).

```
R> gcaPred <- function(model, param, mixType = c("acr", "eecr", "udcr"), effv,
+    refEffv = c(0.05, 0.50, 0.90), lb = 1E-8, ub =  0.90)
```

Illustrations

In this section, we illustrate several examples of the use of the functions described above. Example 1 deals with curve fitting. Example 2 deals with experimental design and mixture toxicity prediction based on CA, IA, and GCA. Example 3 deals with the connection of mixture toxicity prediction in **mixtox** with curve fitting in **drc**. Example 4 deals with follow-up analysis on mixture prediction. The **mixtox** package provides sigmoidal concentration response data (`antibiotox`) to demonstrate the usage of different functions.

The `antibiotox` dataset includes the long term toxicity of seven aminoglycosides (paromomycin sulfate (PAR), spectinomycin dihydrochloridehydrate (SPE), kanamycin sulfate (KAN), streptomycin sulfate (STR), dihydrostreptomycin sesquisulfate hydrate (DIH), gentamycin sulfate (GEN), and Neomycin sulfate (NEO)) and their 12 mixtures designed using different methods on fresh water photobacteria *Vibro-qinghaiensis* sp. Q67. The concentration unit of these antibiotics is mole per liter (mol/L). The bioluminesence of photobacteria was transformed into quantal form with response approaches 0 if chemical's concentration was extremely low and 100% inhibition if chemical's concentration was infinitely high. Detailed information on these data and test systems can be found in the reference manual.

Example 1: Curve fitting

Mixture toxicity prediction is based on the curve fitting information of individual compounds. For example, we need to use the Logit function to fit the concentration response data of PAR on photobacteria. To get proper starting values of α (location parameter) and β (slope parameter) for Logit, users need to use the function `tuneFit`.

```
R> x <- antibiotox$PAR$x
R> expr <- antibiotox$PAR$y
R> y <- rowMeans(expr)
R> tuneFit(x, y, eq = "Logit")
```

The function `tuneFit` will return a list as follows:

```
$sta
      Alpha Beta   r2 adjr2    MAE  RMSE   AIC  AICc   BIC     ecx
fit_1  26.3 4.66 0.995 0.994 0.0234 0.029 -49.1 -47.8 -48.2 1000001
```

The number 1000001 means the effect concentration (ecx) was not calculated because no `effv` was provided to `tuneFit`. To compare the result with Weibull:

```
R> tuneFit(x, y, eq = "Weibull")
```

```
$sta
      Alpha Beta   r2 adjr2    MAE   RMSE AIC  AICc BIC     ecx
fit_1  18.3 3.32 0.993 0.992 0.0223 0.0331 -46 -44.6 -45 1000001
```
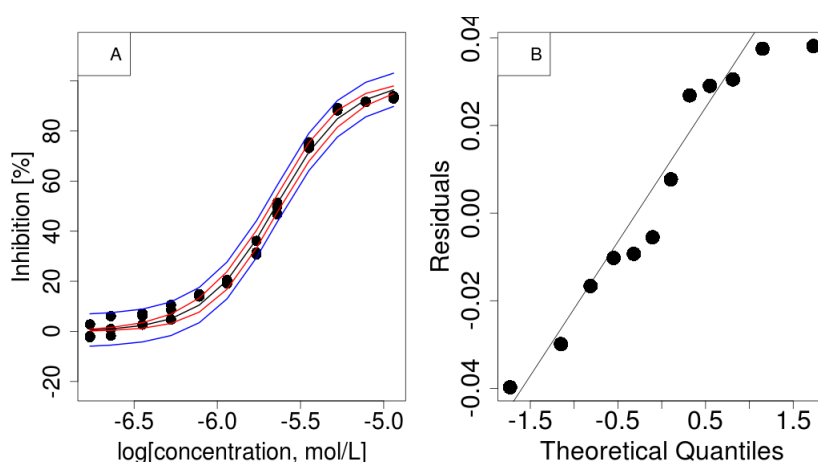
Figure 1: The fitting information of antibiotic PAR on photobacteria: (A) concentration response curves. Dot: observed; Black line: fitted CRC; Red dotted lines: 95% *CI*; Blue dotted lines: 95% *PI*; and (B) normal Q-Q plots of residuals.

These two functions could well describe the concentration response data of PAR on photobacteria ($r^2 > 0.99$). It is difficult to distinguish which fit is better purely based on r^2. Users need to choose the better fit (i.e., Logit) based on AIC or BIC (the lower, the better).

Function `tuneFit` also provides an important feature of fitting a batch of concentration response data of different chemicals. Here, we would like to example curve fitting of two compounds in a single run.

```
R> omim.x <- cytotox$Omim$x
R> omim.y <- rowMeans(cytotox$Omim$y)
R> emim.x <- cytotox$Emim$x
R> emim.y <- rowMeans(cytotox$Emim$y)
R> x <- rbind(omim.x, emim.x)
R> y <- rbind(omim.y, emim.y)
R> tuneFit(x, y, eq = "Weibull")


$sta
      Alpha Beta   r2 adjr2    MAE   RMSE   AIC  AICc   BIC     ecx
fit_1  6.42 2.25 0.946 0.941 0.0447 0.0625 -30.7 -29.3 -29.7 1000001
fit_2  5.57 2.27 0.974 0.972 0.0452 0.0545 -34.0 -32.6 -33.0 1000001
```

In this case, r^2 is still useful to give an illustrative evaluation of the goodness of fit. In cases of various curves fitted by different functions, there is no way to compare their goodness of fit just based on AIC or BIC. Thus, we provide as many statistics as possible to allow users to choose the best fit in the context of their experimental systems.

Function `curveFit` could be used to fit the concentration response data of PAR using the starting values of 26.31 and 4.66 for Logit.

```
R> fit <- curveFit(x, expr, eq = "Logit", param = c(26.31, 4.66), effv = c(0.05, 0.5))
```

The plot of the fitted concentration response curve (Figure 1A) can be shown through function `figPlot`. The residuals of curve fitting can be found in `fit$res`. The normal Q-Q plot of residuals was plotted in Figure 1B using function qq4res in **mixtox**. The R code for Figure 1 is as follows:

```
R> par(mfrow = c(1, 2))
R> x <- antibiotox$PAR$x
R> expr <- antibiotox$PAR$y
R> fit <- curveFit(x, expr, eq = "Logit", param = c(26.31, 4.66), effv = c(0.05, 0.5))
R> figPlot(fit$crcInfo)
R> legend("topleft", legend = "A", border = "white", cex = 1.4)
R> qq4res(fit$res)
R> legend("topleft", legend = "B", border = "white", cex = 1.4)
```

The `effv` values of 0.05 and 0.5 imply that the effect concentrations that cause 5% and 50% effect (EC5 and EC50) are calculated. *PI*s and *CI*s were both calculated for comparison. The fitted α and β are 26.31 and 4.66, respectively. The goodness of fit statistics are 0.995, 0.994, 0.02342, 0.02897,

-49.13, -47.80, and -48.16 for R^2, R^2_{adj}, MAE, $RMSE$, AIC, $AICc$ and BIC, respectively. The function `curveFit` will call `ECx` to calculate effect concentration of `effv`. The EC50 of PAR on the inhibition of luminescence of phtotobacteria is 2.25E−06 mol/L with 95% CI of [2.11E−06, 2.40E−06] mol/L and 95% PI of [1.95E−06, 2.57E−06] mol/L. The EC5 of PAR on the inhibition of luminescence of phtotobacteria is 5.24E−07 mol/L with 95% CI of [4.37E−07, 6.44E−07] mol/L and 95% PI of [0, 8.44E−07] mol/L. Similarly, PAR will cause 50% inhibition with 95% PI of [42.7%, 57.3%] and 95% CI of [46.5%, 53.5%] at the concentration of 2.25E−06 mol/L. PAR will cause 5% inhibition with 95% PI of [0%, 11.7%] and 95% CI of [3.14%, 6.86%] at the concentration of 5.24E−07 mol/L.

The NOEC and LOEC can be calculated using the function `NOEC`:

```
R> NOEC(x, expr)
```

The values of NOEC and LOEC are 2.29E−07 mol/L and 3.55E−07 mol/L for PAR, respectively.

Example 2: Mixture toxicity prediction

The function `caPred` can be called to predict the effect of mixtures based on concentration addition. It provides three optional mixture design methods as mentioned before (ACR, EECR, and UDCR). The effect of mixtures of seven aminoglycosides combined on photobacteria according to the EECR design for 5% and 50% effect levels can be predicted by:

```
R> model <- antibiotox$sgl$model
R> param <- antibiotox$sgl$param
R> caPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))
```

Function `caPred` will return a series of effects e in a range of [0, 1], predicted concentrations ca that cause effects e, and the proportion of individual components (pct) for the EECR mixture. The proportion of individual components is very useful for users to prepare mixtures in practical experiments.

Here, we want to design 10 UDCR mixtures based on seven antibiotics and five concentration levels (effect concentrations at the responses of 5%, 10%, 20%, 30% and 50%, respectively). The function `unidTab` can be called in `caPred` to generate a uniform design table of $U_{10}(10^7)$. The uniform table can also be generated with 10 runs (pseudo-levels) and 7 factors as follows:

```
R> unidTab(10, 7)
```

No	1	2	3	4	5	6	9
1	1	2	3	4	5	6	9
2	2	4	6	8	10	1	7
3	3	6	9	1	4	7	5
4	4	8	1	5	9	2	3
5	5	10	4	9	3	8	1
6	6	1	7	2	8	3	10
7	7	3	10	6	2	9	8
8	8	5	2	10	7	4	6
9	9	7	5	3	1	10	4
10	10	9	8	7	6	5	2

The CA prediction of those UDCR mixtures is as follows:

```
R> caPred(model, param, mixType = "udcr", effv = rep(c(0.05, 0.1, 0.2, 0.3, 0.5), 2))
```

The argument `effv` exemplifies the pseudo-level design of the five levels. Function `caPred` also returns the uniform table employed to construct UDCR mixtures.

Similarly, the effect of mixtures based on IA for the EECR and UDCR design can be predicted as follows:

```
R> iaPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))
R> iaPred(model, param, mixType = "udcr", effv = rep(c(0.05, 0.1, 0.2, 0.3, 0.5), 2))
```

Both the `caPred` and `iaPred` would return the proportion of individual compounds in each mixture.

Users need to fit all of the individual concentration response data using Hill_two (13) before using function `gcaHill`. Assume we have fitted four curves using Hill_two with corresponding fitting information stored in `model_hill2` and `param_hill2`. The effect of mixtures predicted based on generalized concentration addition for the EECR and UDCR design can be calculated as follows:

```
R> model_hill2 <- c("Hill_two", "Hill_two", "Hill_two", "Hill_two")
R> param_hill2 <- matrix(c(3.94e-5, 0.97, 5.16e-4, 1.50, 3.43e-6, 1.04, 9.18e-6, 0.77),
+    nrow = 4, ncol = 2, byrow = TRUE)
R> gcaHill(model_hill2, param_hill2, mixType = "eecr", effv = c(0.05, 0.5))
R> effv <- c(0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.50)
R> gcaHill(model_hill2, param_hill2, mixType = "udcr", effv)
```

Example 3: Connection of mixtox with drc

Package **drc** is widely used for curve fitting of concentration response data. Many users are accustomed to its convenience in curve fitting. The two-parameter log-logistic function (*LL*.2) and two-parameter Weibull functions (*W*1.2 and *W*2.2) with the lower limit fixed at 0 and the upper limit fixed at 1 in **drc** are suitable for quantal responses. The function drm is a general model fitting function for analysis of concentration response data in **drc**.

Here we would like to show an example of connecting the curve fitting (drm) in **drc** with the mixture effects prediction (caPred and iaPred) in **mixtox**. Assume we need to predict the effect of mixtures of four antibiotics (PAR, SPE, KAN, and STR) on photobacteria. First, we need to fit these concentration response data (stored in antibiotox) using drm. The *LL*.2 function is selected to fit these response data.

```
R> library(mixtox)
R> library(drc)
R> PAR.x <- antibiotox$PAR$x
R> PAR.y <- rowMeans(antibiotox$PAR$y)
R> PAR <- data.frame(x = PAR.x, y = PAR.y)
R> SPE <- data.frame(x = antibiotox$SPE$x, y = rowMeans(antibiotox$SPE$y))
R> KAN <- data.frame(x = antibiotox$KAN$x, y = rowMeans(antibiotox$KAN$y))
R> STR <- data.frame(x = antibiotox$STR$x, y = rowMeans(antibiotox$STR$y))

R> PAR.fit <- drm(y ~ x, data = PAR, fct = LL.2())
R> SPE.fit <- drm(y ~ x, data = SPE, fct = LL.2())
R> KAN.fit <- drm(y ~ x, data = KAN, fct = LL.2())
R> STR.fit <- drm(y ~ x, data = STR, fct = LL.2())
R> param.LL.2 <- rbind(PAR.fit$fit$par, SPE.fit$fit$par, KAN.fit$fit$par,
+    STR.fit$fit$par)
R> rownames(param.LL.2) <- c("PAR", "SPE", "KAN", "STR")
R> colnames(param.LL.2) <- c("b", "e")
```

The fitted parameters *b* and *e* for *LL*.2 (Ritz and Streibig, 2005) are stored in param.LL.2. The formula of *LL*.2 in **drc** and Logit in **mixtox** seem different, but they are essentially the same expression. The following equations can be used to transform *b* and *e* in *LL*.2 into α and β in Logit, respectively.

$$\alpha = -\log(e) \cdot \beta, \qquad\qquad \beta = -b \cdot \ln(10). \qquad\qquad (15)$$

The following code can be used to transform *b* and *e* in *LL*.2 into α and β in Logit numerically.

```
R> Beta <- -param.LL.2[, 1] * log(10)
R> Alpha <- -log10(param.LL.2[, 2]) * Beta
R> param.Logit <- cbind(Alpha, Beta)
```

The effect of mixtures of four antibiotics on photobacteria according to, say, the EECR design at 5% and 50% effect levels can be predicted by:

```
R> model.Logit <- c("Logit", "Logit"," Logit", "Logit")
R> caPred(model.Logit, param.Logit, mixType = "eecr", effv = c(0.05, 0.5))
R> iaPred(model.Logit, param.Logit, mixType = "eecr", effv = c(0.05, 0.5))
```

Example 4: Follow-up analysis

Figure 2 shows the comparison of CA and IA prediction with experimental observation of EECR mixtures. The code for this comparison is shown in Appendix.

We can conclude that the mixture toxicity of seven antibiotics can be predicted by the IA model since the IA curve locates in the 95% *PI* of the experimental curve.

Despite the CA and IA prediction at particular effect concentrations, it is also meaningful to know the CA and IA prediction at particular effects. For example, the luminescence of photobacteria will
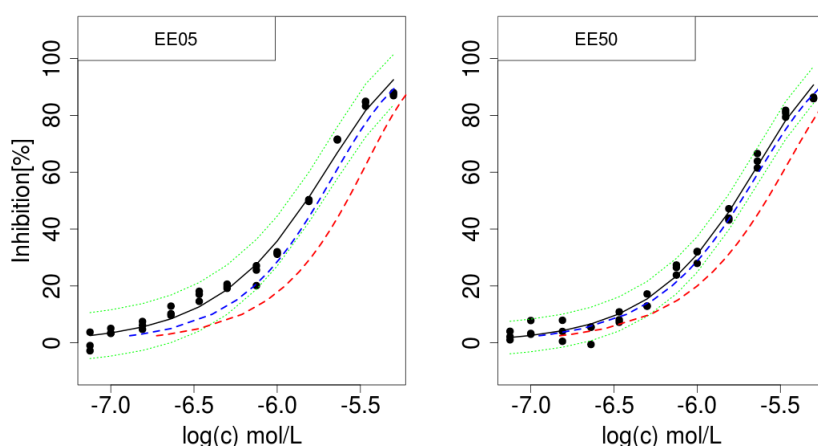
Figure 2: Experimental, predicted CA, and IA concentration response curves of 2 mixtures designed by equal effect concentration ratio (EE05 and EE50). Dot: observed; Black line: fitted CRC; Grey dotted lines: 95% *PI*; Blue dashed line: IA prediction; Red dashed line: CA. The code is listed in Appendix.

be inhibited by 5% at the concentration of 1.41E−7 mol/L and 1.77E−7 mol/L for EE05 and EE50 mixtures, respectively. The CA and IA prediction at the concentration of 1.41E−7 mol/L (EE05) and 1.77E−7 mol/L (EE50) can be calculated based on (7) and (10), respectively. The dichotomy technique was employed to solve the constructed CA equation based on (7). This calculation requires fitted concentration response information of individual chemicals, their ratios in the mixtures, and the established concentration response information of mixtures. Function `eiaPred` and `ecaPred` can be called as follows:

```
R> model <- antibiotox$eecr.mix$model
R> param <- antibiotox$eecr.mix$param
R> pct <- antibiotox$eecr.pct
R> mix <- list(model = model, param = param)
R> eiaPred(effv = 0.05, antibiotox$sgl, mix, pct)
R> ecaPred(effv = 0.05, antibiotox$sgl, mix, pct)
```

The two functions first calculate the effect concentrations based on the fitted concentration response information of mixtures (i.e., the selected equations and associated coefficients contained in `antibiotox$eecr.mix`) according to the input effects `effv` (e.g., 5.0%). The concentration of an individual component (c_i) is computed from (8) based on the mixture's effect concentration and the ratio of components in the mixture `antibiotox$pct` (p_i). The IA predicted effect for EE5 and EE50 are 1.80% and 3.04%, respectively. The CA predicted effect of EE5 and EE50 are 2.78% and 4.25%, respectively.

Conclusions

Package **mixtox** mainly targets toxicologists in the study of chemicals' toxicity and the effect of mixtures. It offers a general framework of curve fitting, experimental design, and mixture prediction for practitioners in toxicology. The unique features of **mixtox** include: (1) construction of uniform design table; and (2) mixture toxicity prediction for multiple components based on CA, IA, and GCA. Function `gcaHill` is capable of examining mixtures containing partial agonists or even full agonists. However, we are lacking experimental data in **mixtox** to verify its ability. **mixtox** aims to predict the effect of mixtures with more than two components. It cannot make isoboles for binary mixtures. Users may need to draw isoboles using package **drc** or other software package based on the results calculated with **mixtox**. The current version of **mixtox** (version number 1.3.1) is available on CRAN. We received a lot of feedback from users since its release last year. Now it is still under rapid development to improve its performance and try to include more features in toxicology.

Acknowledgement

Bibliography

T. Backhaus and M. Faust. Predictive environmental risk assessment of chemical mixtures: A conceptual framework. *Environmental Science & Technology*, 46(5):2564–2573, 2012. doi: 10.1021/es2034125. [p422]

S. De Gryze, I. Langhans, and M. Vandebroek. Using the correct intervals for prediction: A tutorial on tolerance intervals for ordinary least-squares regression. *Chemometrics and Intelligent Laboratory Systems*, 87(2):147–154, 2007. doi: 10.1016/j.chemolab.2007.03.002. [p424]

H. Dette and A. Pepelyshev. Generalized latin hypercube design for computer experiments. *Technometrics*, 52(4):421–429, 2010. doi: 10.1198/tech.2010.09157. [p424, 426]

C. W. Dunnett. New tables for multiple comparisons with a control. *Biometrics*, 30(3):482–491, 1964. doi: 10.2307/2528490. [p426]

R. Dybowski and V. Gant. *Clinical Applications of Artificial Neural Networks*. Cambridge University Press, Cambridge, 1st edition, 2001. doi: 10.1017/cbo9780511543494. [p423]

T. V. Elzhov, K. M. Mullen, A.-N. Spiess, and B. Bolker. *minpack.lm: R Interface to the Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in MINPACK, Plus Support for Bounds*, 2016. URL https://CRAN.R-project.org/package=minpack.lm. R package version 1.2-1. [p423]

M. Faust, R. Altenburger, T. Backhaus, H. Blanck, W. Boedeker, P. Gramatica, V. Hamer, M. Scholze, M. Vighi, and L. H. Grimme. Predicting the joint algal toxicity of multi-component s-triazine mixtures at low-effect concentrations of individual toxicants. *Aquatic Toxicology*, 56(1):13–32, 2001. doi: 10.1016/s0166-445x(01)00187-4. [p422, 424, 425]

M. Faust, R. Altenburger, T. Backhaus, H. Blanck, W. Boedeker, P. Gramatica, V. Hamer, M. Scholze, M. Vighi, and L. H. Grimme. Joint algal toxicity of 16 dissimilarly acting chemicals is predictable by the concept of independent action. *Aquatic Toxicology*, 63(1):43–63, 2003. doi: 10.1016/s0166-445x(02)00133-9. [p422]

S. Goutelle, M. Maurin, F. Rougier, X. Barbaut, L. Bourguignon, M. Ducher, and P. Maire. The hill equation: a review of its capabilities in pharmacological modelling. *Fundamental & Clinical Pharmacology*, 22(6):633–648, 2008. doi: 10.1111/j.1472-8206.2008.00633.x. [p422]

N. Hadrup, C. Taxvig, M. Pedersen, C. Nellemann, U. Hass, and A. M. Vinggaard. Concentration addition, independent action and generalized concentration addition models for mixture effect prediction of sex hormone synthesis in vitro. *PLoS ONE*, 8(8):e70490, 2013. doi: 10.1371/journal.pone.0070490. [p423]

F. J. Hickernell. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67(211):299–322, 1996. doi: 10.1090/s0025-5718-98-00894-1. [p424]

G. J. Howard and T. F. Webster. Generalized concentration addition: A method for examining mixtures containing partial agonists. *Journal of Theoretical Biology*, 259(3):469–477, 2009. doi: 10.1016/j.jtbi.2009.03.030. [p423, 425]

G. J. Howard, J. J. Schlezinger, M. E. Hahn, and T. F. Webster. Generalized Concentration Addition Predicts Joint Effects of Aryl Hydrocarbon Receptor Agonists with Partial Agonists and Competitive Antagonists. *Environmental Health Perspectives*, 118(5):666–672, 2010. doi: 10.1289/ehp.0901312. [p425]

S. Huet, A. Bouvier, M.-A. Poursat, and E. Jolivet. *Statistical Tools for Nonlinear Regression: A Practical Guide with S-PLUS and R Examples*. Springer-Verlag, 2004. doi: 10.1007/978-1-4757-2523-0. [p424]

Z. N. Kamvar. ezec: Easy interface to effective concentration calculations, 2016. URL https://CRAN.R-project.org/package=ezec. R package version 1.0.1. [p422]

R. Kavlock, K. Chandler, K. Houck, S. Hunter, R. Judson, N. Kleinstreuer, T. Knudsen, M. Martin, S. Padilla, D. Reif, A. Richard, D. Rotroff, N. Sipes, and D. Dix. Update on EPA's ToxCast program: Providing high throughput decision support tools for chemical risk management. *Chemical Research in Toxicology*, 25(7):1287–1302, 2012. doi: 10.1021/tx3000939. [p426]

Y.-Z. Liang, K.-T. Fang, and Q.-S. Xu. Uniform design and its applications in chemistry and chemical engineering. *Chemometrics and Intelligent Laboratory Systems*, 58(1):43–57, 2001. doi: 10.1016/s0169-7439(01)00139-3. [p424]

J. G. Liao and D. McGee. Adjusted coefficients of determination for logistic regression. *The American Statistician*, 57(3):161–165, 2003. doi: 10.1198/0003130031964. [p423]

S.-S. Liu, J. Zhang, Y.-H. Zhang, and L.-T. Qin. APTox: Assessment and prediction on toxicity of chemical mixtures. *Acta Chimica Sinica*, 70(14):1511–1517, 2012. doi: 10.6023/a12050175. [p422]

S.-S. Liu, Q.-F. Xiao, J. Zhang, and M. Yu. Uniform design ray in the assessment of combined toxicities of multi-component mixtures. *Science Bulletin*, 61(1):52–58, 2015. doi: 10.1007/s11434-015-0925-6. [p422, 424]

J.-H. Ning, K.-T. Fang, and Y.-D. Zhou. Uniform design for experiments with mixtures. *Communications in Statistics – Theory and Methods*, 40(10):1734–1742, 2011. doi: 10.1080/03610921003637470. [p422, 424]

J. Ranke. *drfit: Dose-Response Data Evaluation*, 2016. URL https://CRAN.R-project.org/package=drfit. R package version 0.6.7. [p422]

C. Ritz and J. C. Streibig. Bioassay analysis using R. *Journal of Statistical Software*, 12(5):1–22, 2005. doi: 10.18637/jss.v012.i05. [p422, 423, 426, 430]

C. Ritz and J. C. Streibig. From Additivity to Synergism – A Modelling Perspective. *Synergy*, 1(1): 22–29, 2014. doi: 10.1016/j.synres.2014.07.010. [p423]

M. Scholze, W. Boedeker, M. Faust, T. Backhaus, R. Altenburger, and L. H. Grimme. A general best-fit method for concentration-response curves and the estimation of low-effect concentrations. *Environmental Toxicology and Chemistry*, 20(2):448–457, 2001. doi: 10.1002/etc.5620200228. [p422]

A.-N. Spiess and N. Neumeyer. An evaluation of R^2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: A Monte Carlo approach. *BMC Pharmacology*, 10(6): 11, 2010. doi: 10.1186/1471-2210-10-6. [p422, 423]

Y.-D. Zhou and K.-T. Fang. An efficient method for constructing uniform designs with large size. *Computational Statistics*, 28(3):1319–1331, 2013. doi: 10.1007/s00180-012-0359-4. [p424]

X. Zhu. *mixtox: Curve Fitting and Mixture Toxicity Assessment*, 2016. URL https://CRAN.R-project.org/package=mixtox. R package version 1.3.1. [p423]

X.-W. Zhu, S.-S. Liu, L.-T. Qin, F. Chen, and H.-L. Liu. Modeling non-monotonic dose–response relationships: Model evaluation and hormetic quantities exploration. *Ecotoxicology and Environmental Safety*, 89:130–136, 2013. doi: 10.1016/j.ecoenv.2012.11.022. [p424]

Appendix

```
library(mixtox)
par(mfrow = c(1, 2))
model <- antibiotox$sgl$model
param <- antibiotox$sgl$param
eeca <- caPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))
eeia <- iaPred(model, param, mixType = "eecr", effv = c(0.05, 0.5))
# plot EE05 mixture
par(mar = c(5, 5, 1, 1))
x <- antibiotox$ee05$x
expr <- antibiotox$ee05$y
ee05fit <- curveFit(x, expr, eq = antibiotox$eecr.mix$model[1],
  param = antibiotox$eecr.mix$param[1, ])
plot(rep(log10(x), ncol(expr)), expr * 100, pch = 20, ylim = c(-10, 110),
  xlab = "log(c) mol/L", ylab = "Inhibition[\%]", cex = 1.8, cex.lab = 1.8,
  cex.axis = 1.8)
lines(log10(x), ee05fit$crcInfo[, 2] * 100, col = 1, lwd = 2)
lines(log10(x), ee05fit$crcInfo[, 6] * 100, col = "green", lwd = 1.5, lty = 3)
lines(log10(x), ee05fit$crcInfo[, 7] * 100, col = "green", lwd = 1.5,  lty = 3)
lines(log10(eeia$ia[1, ]), eeia$e * 100, col = "red", lwd = 2.5, lty = 2)
lines(log10(eeca$ca[1, ]), eeca$e * 100, col = "blue", lwd = 2.5, lty = 2)
legend("topleft", legend = "EE05", border = "white", cex = 1.4)
# plot EE50 mixture
par(mar = c(5, 5, 1, 1))
x <- antibiotox$ee50$x
expr <- antibiotox$ee50$y
ee50fit <- curveFit(x, expr, eq = antibiotox$eecr.mix$model[1],
  param = antibiotox$eecr.mix$param[1, ])
plot(rep(log10(x), ncol(expr)), expr * 100, pch = 20, ylim = c(-10, 110),
  xlab = "log(c) mol/L", ylab = "", cex = 1.8, cex.lab = 1.8, cex.axis = 1.8)
lines(log10(x), ee50fit$crcInfo[, 2] * 100, col = 1, lwd = 2)
lines(log10(x), ee50fit$crcInfo[, 6] * 100, col = "green", lwd = 1.5, lty = 3)
lines(log10(x), ee50fit$crcInfo[, 7] * 100, col = "green", lwd = 1.5, lty = 3)
lines(log10(eeia$ia[2, ]), eeia$e * 100, col = "red", lwd = 2.5, lty = 2)
lines(log10(eeca$ca[2, ]), eeca$e * 100, col = "blue", lwd = 2.5, lty = 2)
legend("topleft", legend = "EE50", border = "white", cex = 1.4)
```

Xiang-Wei Zhu
Qingdao Agricultural University
College of Resource and Environment
and
Qingdao Engineering Research Center for Rural Environment
266109 Qingdao, China
xwzhunc@gmail.com

Jian-Yi Chen
Qingdao Agricultural University
Science and Information College
266109 Qingdao, China
chenjy@amss.ac.cn