

Dose-response Explorer: Exploring different models and drug-drug interaction

Kevin R Coombes and Pan Tong

October 1, 2014

Contents

1	Introduction	1
2	Outlier identification	2
3	Assessing Reproducibility	3
4	Fit dose-response models	4
5	Predict response	5
6	Obtain IC values	6
7	Comparing multiple dose-response curves	6
8	Drug Interaction Index	9
9	GUI Usage	16
10	Session Info	17

1 Introduction

There is a great variety of models developed for dose-response data, many of which have been implemented in the *drc* and *DoseFinding* packages. *drexplorer* combines both packages to aid the user to visually examine and compare how existing models perform on the data. We also incorporate model selection. Another important feature for *drexplorer* is to allow the user to identify outlier measurements and visually examine how these outliers affect the fitted model.

In addition to fit dose-response models, *drexplorer* also implements methods previously published to assess drug-drug interaction. Graphical User Interfaces

(GUIs) have been designed to allow users without advanced programming skills to perform dose-response analysis.

The main entry for *drexplorer* is the `drFit()` function and `computeIC()` function. `drFit()` fits a model. Outlier detection is also embedded into `drFit()`. Once a model is fitted, `computeIC()` computes IC values at specified percentiles.

2 Outlier identification

This package implements the method by Newman, D. The test statistic $q=w/s$ where w is the range of the data and s is sample standard deviation estimated from controls. The null distribution for q has been derived and 1% and 5% quantiles have been given in the paper.

We implement this procedure. In particular, `NewmanTest()` returns a logic vector specifying whether the observations are outliers. Usually, drug-response data contains multiple doses. Therefore, we write a wrapper `drOutlier()` that compute the result for all doses each dose at a time.

We use the `ryegrass` data from *drc* package for illustration purpose.

First, we load the *drexplorer* package and attach the *ryegrass* data.

```
library(drexplorer)
data(ryegrass)
```

At dose=3.75 and significance level 0.05, we find there is one outlier identified:

```
dose <- ryegrass[, 2]
response <- ryegrass[, 1]
## potential outlier at dose 3.75
NewmanTest(ref = response[dose == 0], obs = response[dose ==
  3.75], alpha = 0.05)

## [1] FALSE FALSE TRUE
```

We also examine all dose levels and find no further outliers:

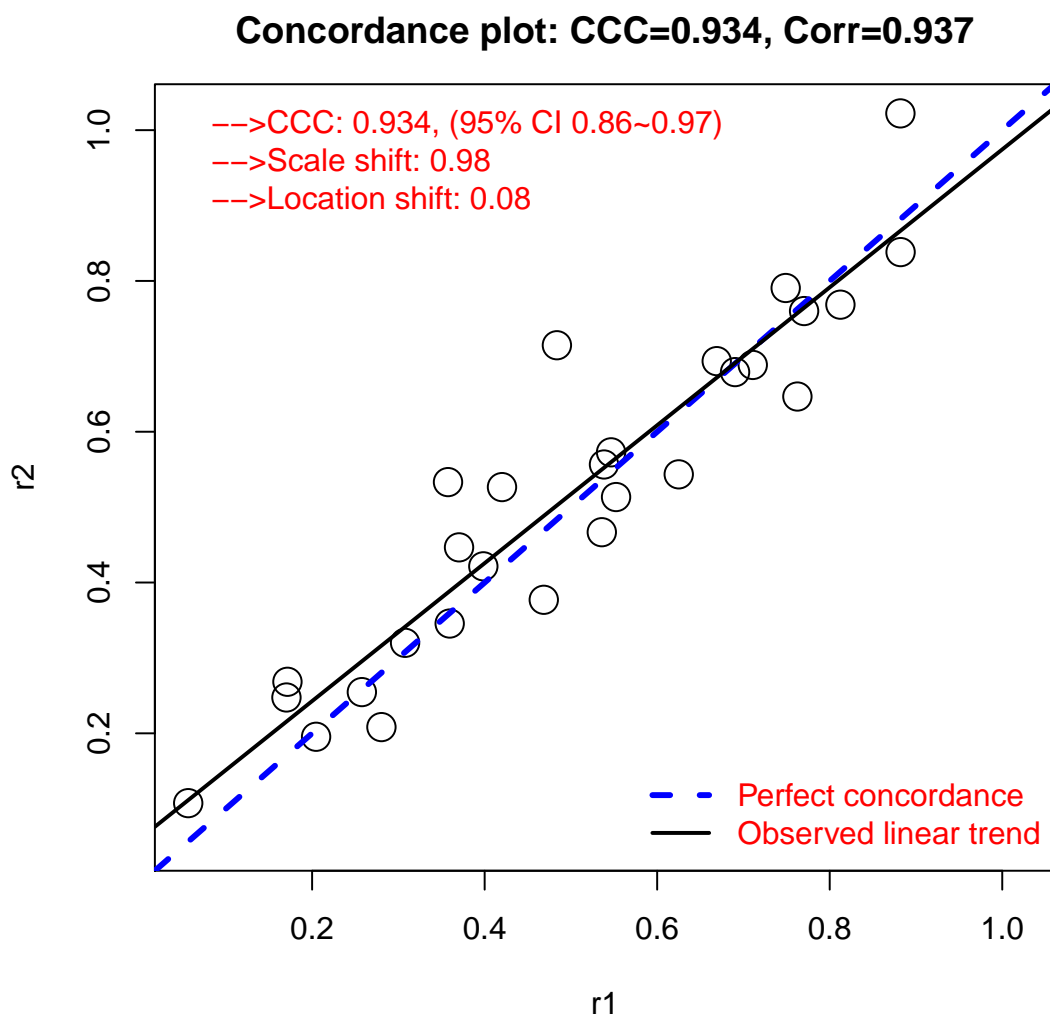
```
drOutlier(drMat = ryegrass[, c(2, 1)], alpha = 0.05)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

3 Assessing Reproducibility

Sometimes replicated viability assays are performed. In such case, it is useful to examine if the experiments are reproducible. A good metric is the Concordance Correlation Coefficient (CCC) that captures both the location shift as well as scale shift between the replicates. The `plotCCC` function can be used to compute CCC and visualize the replicated data.

```
set.seed(100)
r1 <- runif(28)
r2 <- r1 + rnorm(28, 0, 0.1)
ccc <- plotCCC(r1, r2)
```



```
ccc
```

```
##      ccc s_shift l_shift  ccc_lo  ccc_hi      Cb    corr mean_x mean_y  sd_x  
## 0.93365 0.97649 0.07578 0.86264 0.96857 0.99686 0.93660 0.50119 0.51820 0.23138
```

```
##      sd_y
## 0.22594
```

Here we simulate two response vectors and calculate CCC. The computed CCC value is 0.934, location shift is 0.076, scale shift is 0.976, Pearson correlation is 0.937.

4 Fit dose-response models

Below we show how to fit a dose-response model. The `fitDRC()` function is a wrapper to the *drc* and *DoseFinding* packages. Therefore, all models implemented by either package can be fitted. A model is specified by a `modelName` and package name to be passed to this function.

Outliers can be identified and removed from model fitting by specifying the parameter `alpha` (either at significance level of 0.01 or 0.05). To disable outlier identification, set `alpha=1`.

To remove controls (responses at `dose=0`) during model fitting, we can set `fitCtr=FALSE`.

Note that the responses are scaled by mean response at `dose=0` before model fitting.

Below we fit a `sigmaEmax` model. We set `alpha=1` to disable outlier removal and `fitCtr=FALSE` to exclude controls.

```
fit_sigEmax_alpha1 <- drFit(drMat = ryegrass[, c(2,
  1)], modelName = "sigEmax", alpha = 1, fitCtr = FALSE)

## Message: Need bounds in "bnds" for nonlinear models, using default
## bounds from "defBnds".
```

The result is slightly different when outliers passing significance level of 0.05 is removed.

```
fit_sigEmax_alpha_o5 <- drFit(drMat = ryegrass[, c(2,
  1)], modelName = "sigEmax", alpha = 0.05, fitCtr = FALSE)

## Message: Need bounds in "bnds" for nonlinear models, using default
## bounds from "defBnds".

fit_sigEmax_alpha1@fit

## Dose Response Model
##
## Model: sigEmax
```

```
## Fit-type: normal
##
## Coefficients dose-response model
##      e0      eMax      ed50      h
## 1.0290 -0.9703  2.9920  2.8090
##
## Degrees of freedom: 14
## Residual standard error: 0.073

fit_sigEmax_alpha_o5@fit

## Dose Response Model
##
## Model: sigEmax
## Fit-type: normal
##
## Coefficients dose-response model
##      e0      eMax      ed50      h
## 1.017 -0.945  2.789  3.384
##
## Degrees of freedom: 13
## Residual standard error: 0.05673
```

5 Predict response

Once a model is fitted, it can be used to make predictions.

Below we make predictions at the observed dose levels with a previously fitted model. Since the responses are scaled by mean response at dose=0 in model fitting, the predicted responses are also scaled by the mean response from controls. By default, the predict function makes predictions at observed doses.

```
y <- predict(fit_sigEmax_alpha_o5)
y

##      1      2      3      4      5      6      7      8      9     10
## 1.01741 1.01741 1.01741 1.01741 1.01741 1.01741 0.99418 0.99418 0.99418 0.82053
##      11     12     13     14     15     16     17     18     19     20
## 0.82053 0.82053 0.32623 0.32623 0.32623 0.10454 0.10454 0.10454 0.07561 0.07561
##      21     22     23     24
## 0.07561 0.07274 0.07274 0.07274
```

6 Obtain IC values

We implement two approaches for IC value computation. One is to interpolate the observed dosages and try to use the dose that has the predicted response closest to the specified percentile of IC value. The second approach is to use root finding by setting the fitted model to equal to the specified percentile. In most cases, the result are similar. However, the latter approach may give IC50 values beyond observed dosages and sometimes not robust. The `computeIC()` function implements both approaches. By setting `interpolation=TRUE` (the default value) in the `computeIC()` function, the interpolation approach will be selected.

Computing IC values at different quantiles is also easy. Similar to the `fitDRC()` function, different models as well as other options (`alpha` and `fitCtr`) can be specified in estimating IC value.

Below we estimate IC50 at different percentiles with the sigmoid Emax model with outlier removal (`alpha=0.05`) fitted previously. We see that estimates from interpolation and prediction by the model are quite similar.

```
computeIC(fit_sigEmax_alpha_o5, percent = seq(0, 1,
  by = 0.1), log.d = FALSE, interpolation = TRUE)

##      IC0      IC10      IC20      IC30      IC40      IC50      IC60      IC70      IC80      IC90      IC100
## 0.940  1.566  1.952  2.281  2.600  2.948  3.362  3.916  4.828  7.855 30.000

computeIC(fit_sigEmax_alpha_o5, percent = seq(0, 1,
  by = 0.1), log.d = FALSE, interpolation = FALSE)

##      IC0      IC10      IC20      IC30      IC40      IC50      IC60      IC70
## 8.616e-01 1.566e+00 1.952e+00 2.280e+00 2.603e+00 2.951e+00 3.364e+00 3.916e+00
##      IC80      IC90      IC100
## 4.829e+00 7.857e+00 1.000e+30
```

7 Comparing multiple dose-response curves

We provide S4 generic functions (`plot` and `lines`) for fitted model. As a result, it is easy to compare different models and graphically examine outliers through multiple dose-response curves.

Outliers at significance levels 0.01 and 0.05 are indicated by different colors and symbols. Below we show the LL.3, LL.3u and sigEmax curves in this example corresponding to the three-parameter log-logistic model with lower limit 0, three-parameter log-logistic with upper limit 1 and the sigmoid Emax model.

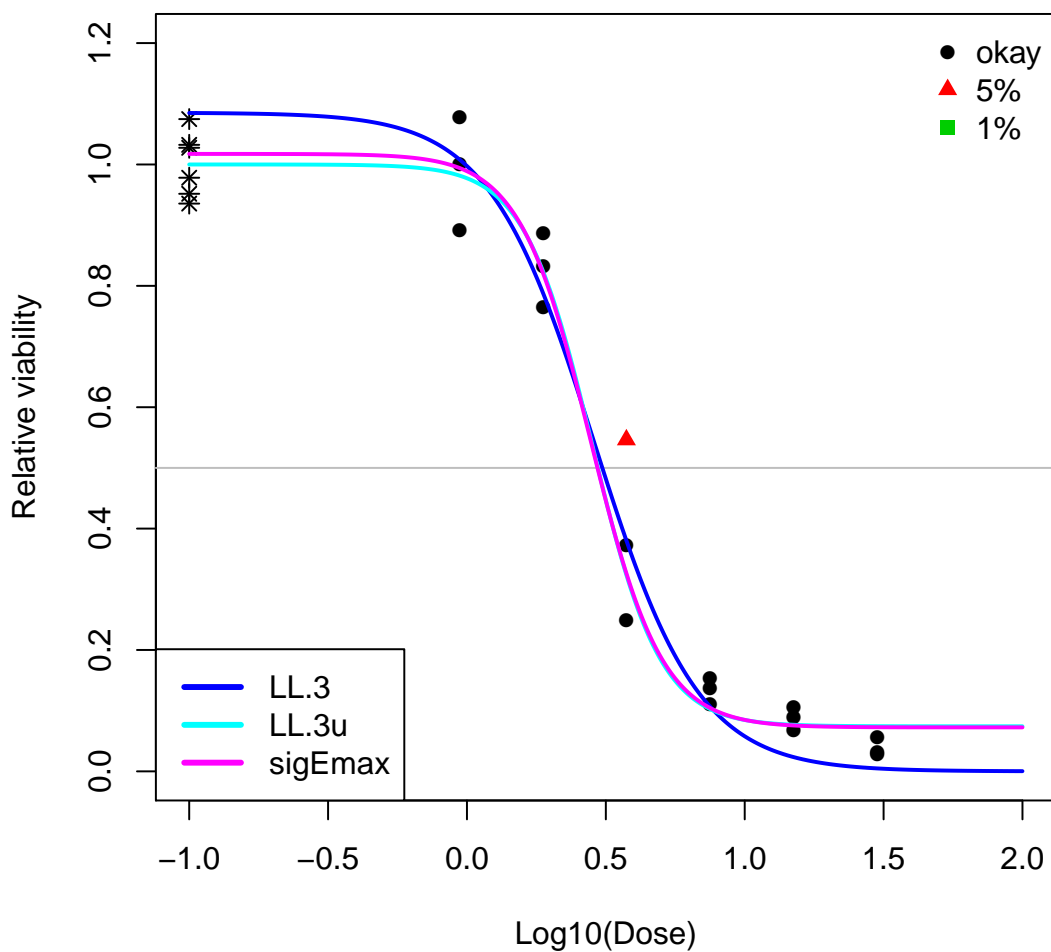
```

fit.LL.3 <- drFit(drMat = ryegrass[, c(2, 1)], modelName = "LL.3",
  alpha = 0.05, fitCtr = FALSE)
fit.LL.3u <- drFit(drMat = ryegrass[, c(2, 1)], modelName = "LL.3u",
  alpha = 0.05, fitCtr = FALSE)
fit.sigEmax <- drFit(drMat = ryegrass[, c(2, 1)], modelName = "sigEmax",
  alpha = 0.05, fitCtr = FALSE)

## Message: Need bounds in "bnds" for nonlinear models, using default
bounds from "defBnds".

###
plot(fit.LL.3, main = "", col = 4, lwd = 2)
lines(fit.LL.3u, col = 5, lwd = 2)
lines(fit.sigEmax, col = 6, lwd = 2)
legend("bottomleft", c("LL.3", "LL.3u", "sigEmax"),
  col = 4:6, lwd = 3)

```



With these many models fitted, which one should be preferred? One way is to look at the Residual Standard Error (RSE) as below. We see that the LL.3u model is best by the RSE criteria.

```
sapply(list(fit.LL.3, fit.LL.3u, fit.sigEmax), function(x) x@info$RSE)

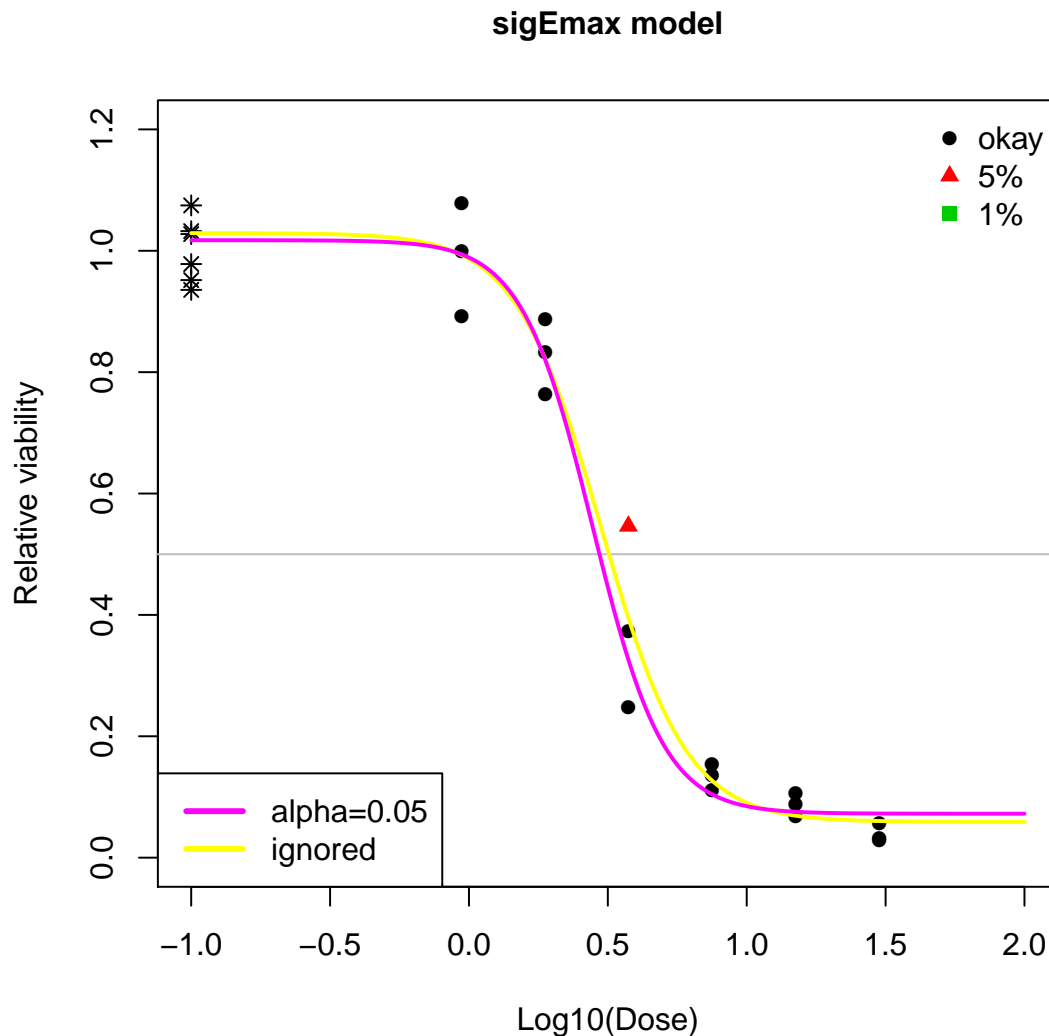
## [1] 0.06987 0.05501 0.05673
```

We also compare the curve using sigEmax model with and without outlier identification.

```
# no outlier excluded
fit.sigEmax0 <- drFit(drMat = ryegrass[, c(2, 1)],
  modelName = "sigEmax", alpha = 1, fitCtr = FALSE)

## Message: Need bounds in "bnds" for nonlinear models, using default
bounds from "defBnds".

####
plot(fit.sigEmax0, main = "sigEmax model", col = 7,
  lwd = 2)
lines(fit.sigEmax, col = 6, lwd = 2)
legend("bottomleft", c("alpha=0.05", "ignored"), col = c(6,
  7), lwd = 3)
```

8 Drug Interaction Index

Administering two drugs simultaneously might induce stronger effect than if administered separately. This is called synergism. Experiments to detect synergism (or antagonism which is the opposite) are usually in two forms. One is the fixed ratio design (ray design) where the ratio of doses between two drugs is a constant. Another one is grid design which means all-possible combinations of drug doses are available.

Two papers have been published regarding to drug interaction index (IAI) by Lee et al, one in 2007 (Lee2007) [?] and on in 2009 (Lee2009) [?]. The Lee2007 paper described five methods to assess interaction: (1) Lowewe additivity model using interaction index (IAI) (2) Model of Greco et al 1990. This approach uses α as the metric and it can be related to IAI (3) Model of Machado and Robinson which uses a metric denoted as η (4) Model of Plummer and Short which can also be linked to IAI through the parameter β_4 (5) Model of Carter et al that can

be linked to IAI through the parameter β_{12} . For more details of these models, please refer to Lee2007 [?].

The two papers by Lee et al discussed the fixed ratio design and the source code for doing this is incorporated into *drexplorer*. To work on grid design, a fixed ratio from the data needs to be selected in order to apply their method. For example, the Lee2007 paper provided an example of grid design. A fixed ratio of 1 was specified in the paper. The specification of fixed ratio would affect the fitted median effect model (see definition in [?]) for the drug mixture as well as estimation of IAI. As a result, IAI has a ratio dependent interpretation.

Below we load the UMSCC22B data from [?]. This data has a fixed ratio design. The `fitIAI` function estimates IAI as well as its confidence interval after specifying dose1, dose2 and effect (between 0 and 1).

```
data(UMSCC22B)
fit_fixedRay <- fitIAI(d1 = UMSCC22B[, 1], d2 = UMSCC22B[,
  2], e = UMSCC22B[, 3], name1 = "SCH66336", name2 = "4HPR")

## Warning: Fixed ratio design detected. Fixed ratio is 1.000
```

The `plotIAI` function is then used to generate different plots including IAI versus response, IAI versus dose (predicted dose for the drug mixture, see equation (6) in [?]), median effect plot and dose response curves. We can also plot IAI versus response as well as IAI versus dose in one figure by specifying `mode='both'`.

The median effect equation [?] is as following:

$$E = \frac{(d/D_m)^m}{1 + (d/D_m)^m}$$

where E is the induced effect of a drug with dose d whose median effective dose is D_m and m is a slope parameter.

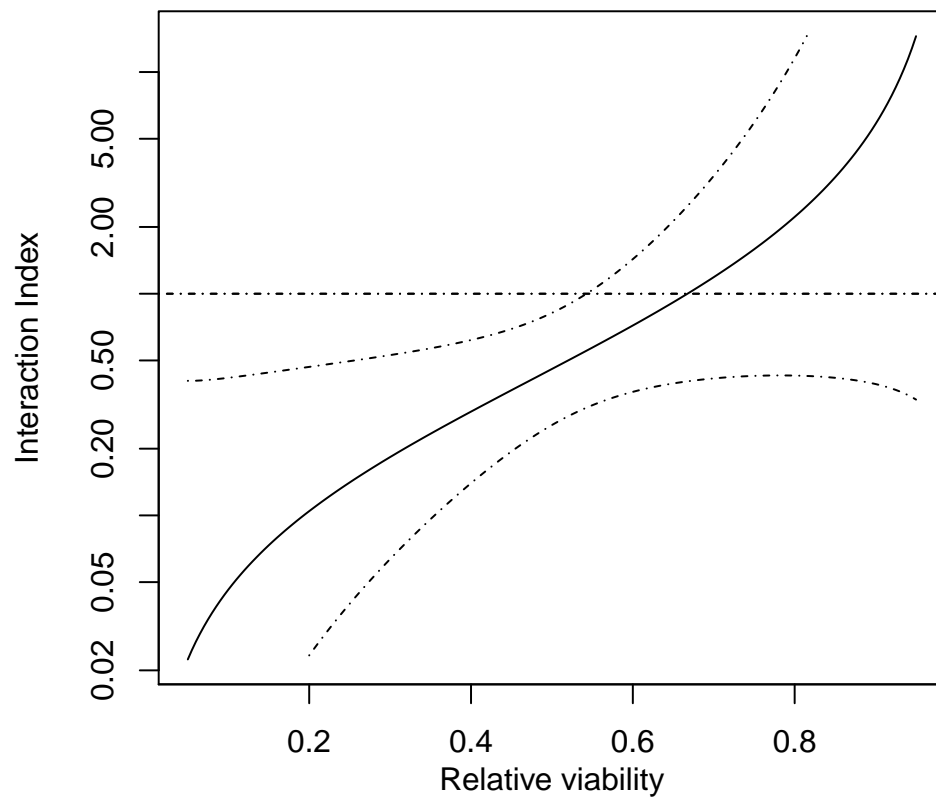
This equation can be arranged as:

$$\text{logit}(E) = m(\log d - \log D_m)$$

The median effect plot is just plotting $\text{logit}(E)$ versus \log_{10} dose; The dose response curve is plotting E versus dose.

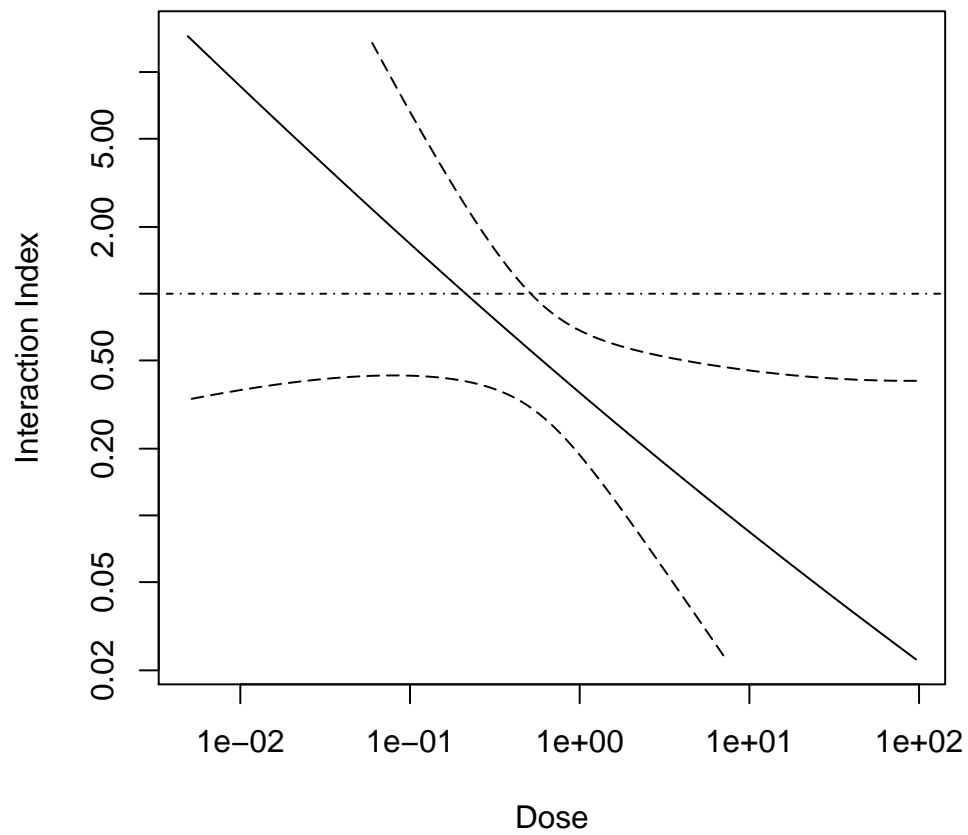
```
# IAI vs response
plotIAI(fit_fixedRay, type = "IAI", mode = "response")
```

Interaction plot

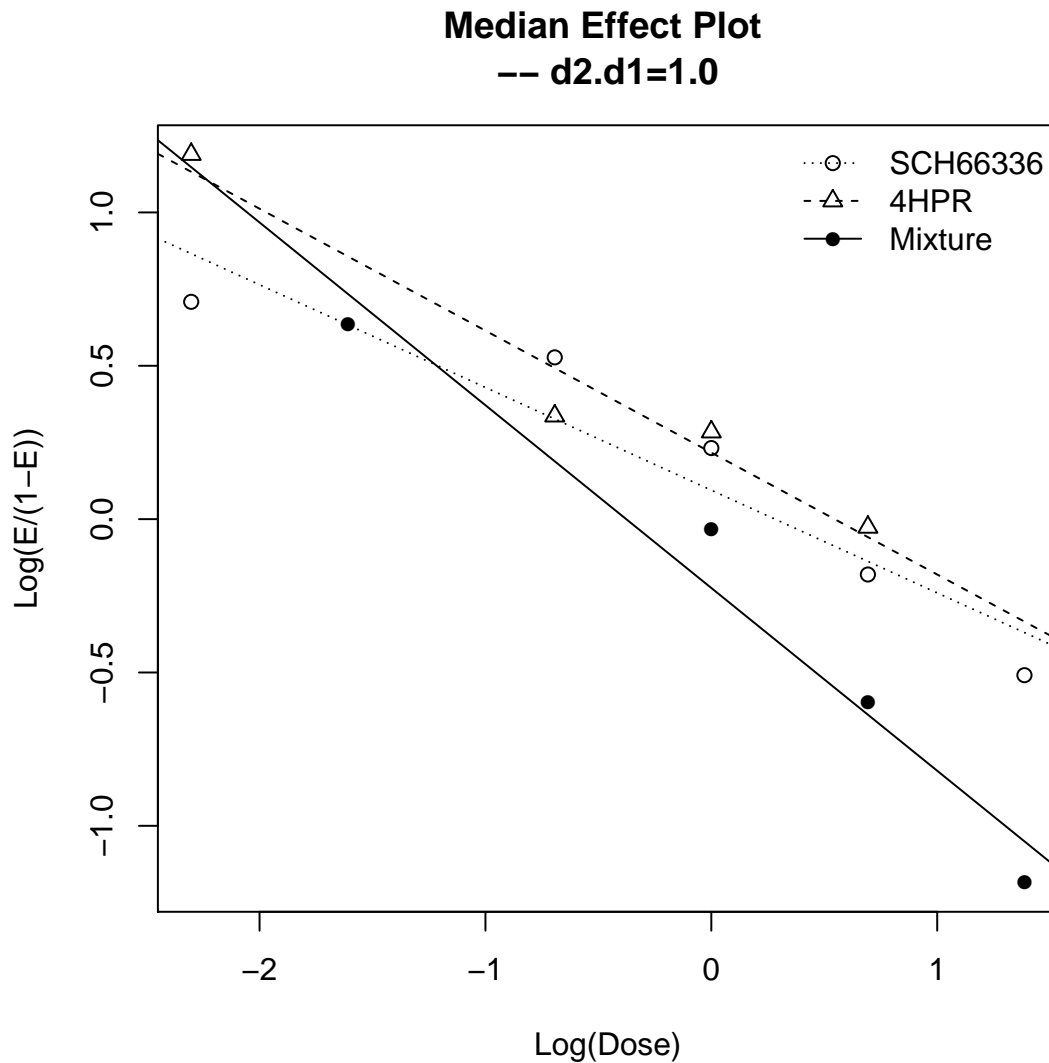


```
# IAI versus dose  
plotIAI(fit_fixedRay, type = "IAI", mode = "dose")
```

Interaction plot



```
# median effect  
plotIAI(fit_fixedRay, type = "medianEffect")
```



In [?], there is an example data (nl22B2) using grid design. Here we examine the estimate of IAI at different fixed ratios.

```
data(nl22B2)
fit_allPoss_1 <- fitIAI(d1 = nl22B2$schd, d2 = nl22B2$hpr,
  e = nl22B2$y1, name1 = "SCH66336", name2 = "4HPR",
  d2.d1.force = 1)

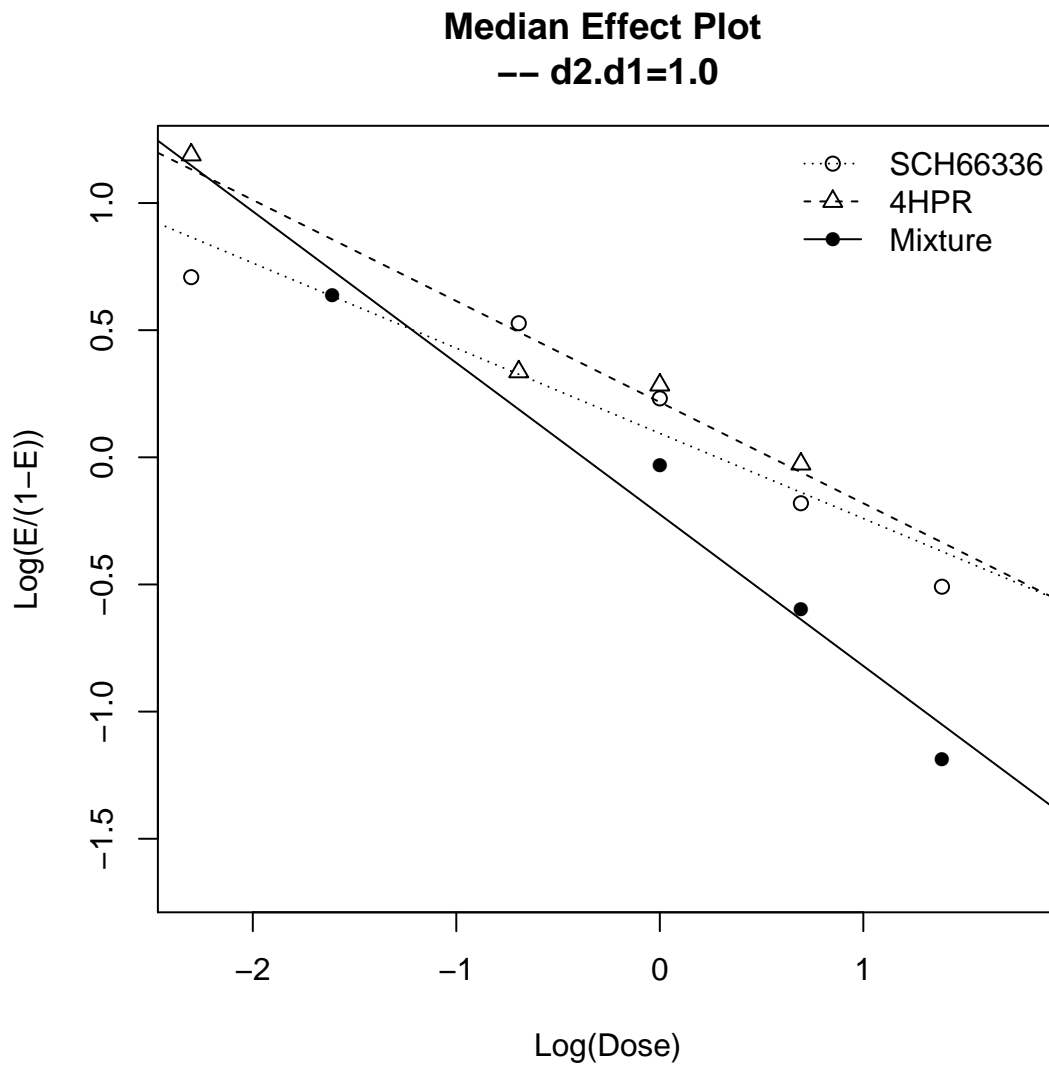
## Warning: This is not a ray design; computation is done with method
derived from fixed ratio!

fit_allPoss_2 <- fitIAI(d1 = nl22B2$schd, d2 = nl22B2$hpr,
  e = nl22B2$y1, name1 = "SCH66336", name2 = "4HPR",
  d2.d1.force = 2)

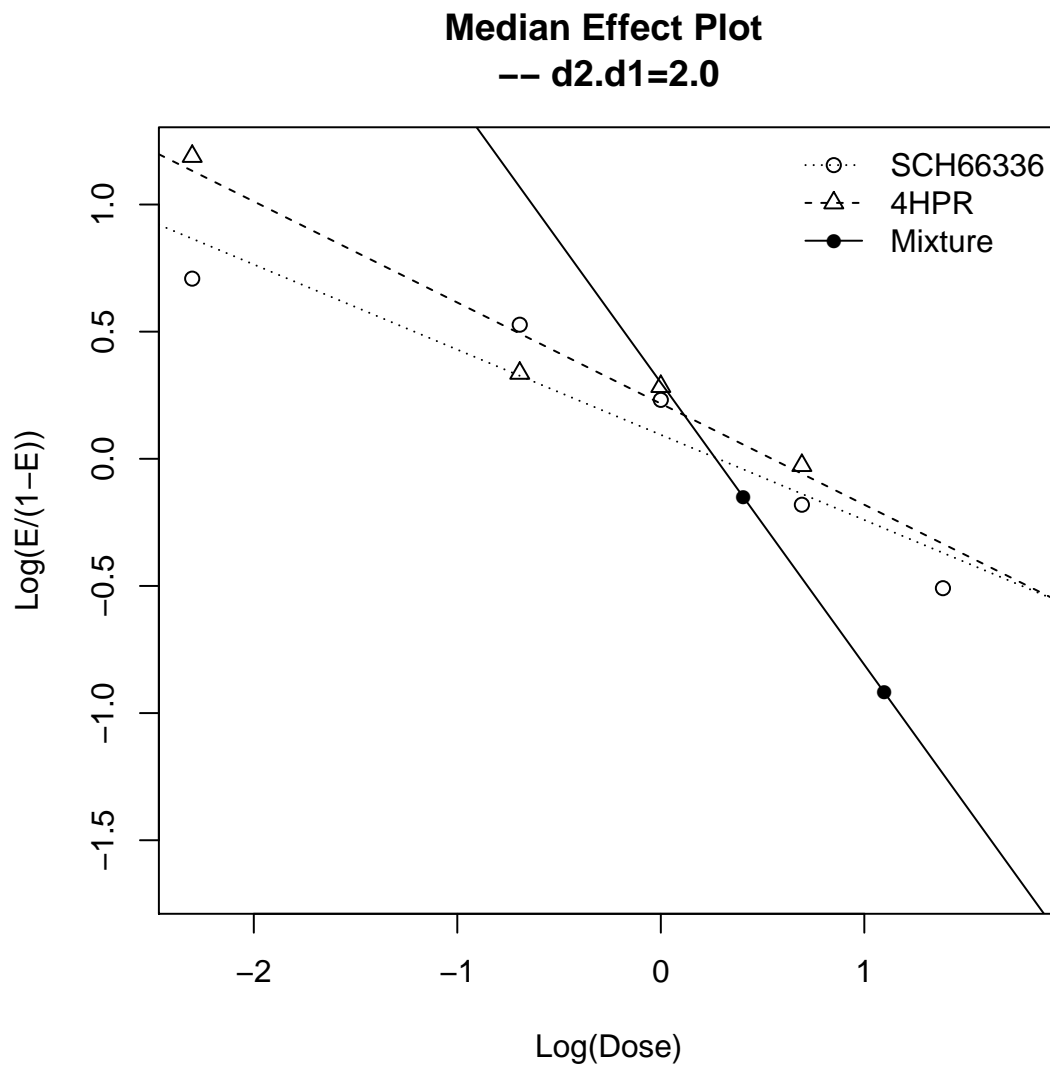
## Warning: This is not a ray design; computation is done with method
derived from fixed ratio!
```

From the median effect plot, we can find that there are 4 data points for drug mixtures at fixed ratio of 1 while only 2 data points are available at fixed ratio of 2.

```
# median effect  
plotIAI(fit_allPoss_1, type = "medianEffect")
```



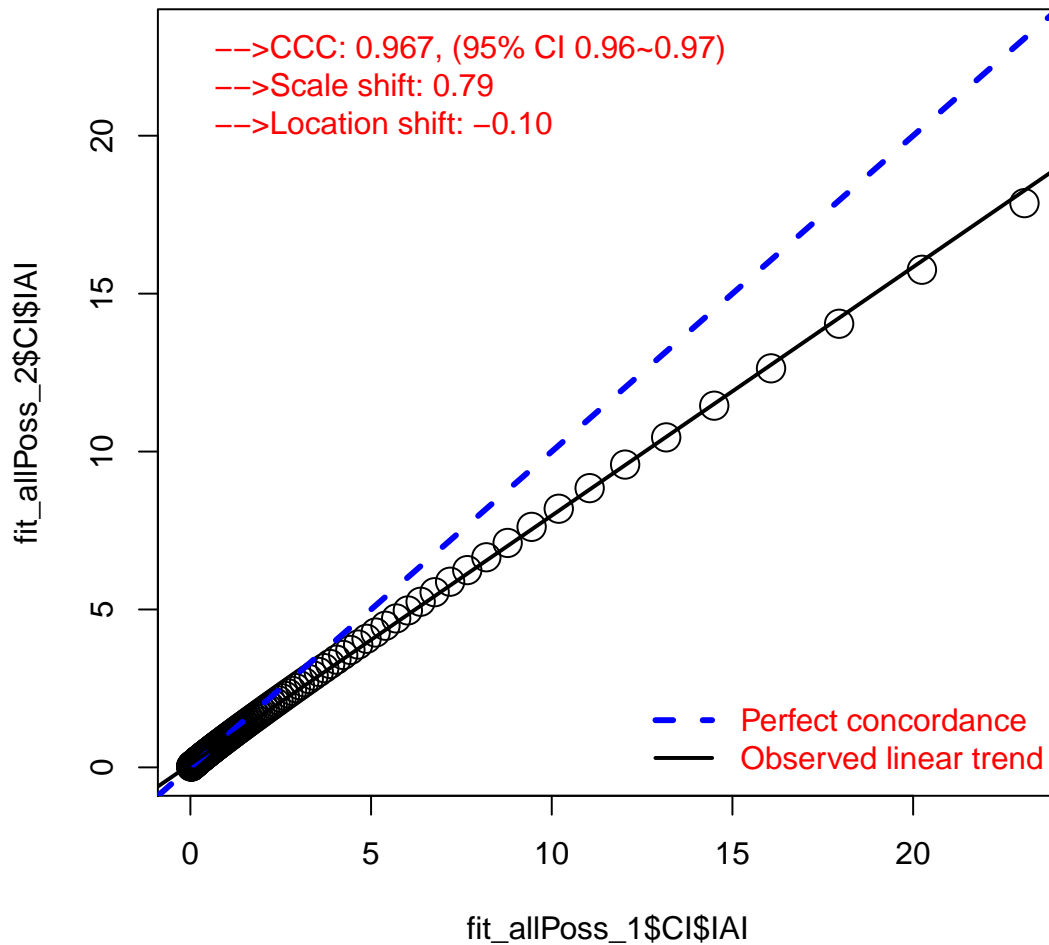
```
plotIAI(fit_allPoss_2, type = "medianEffect")
```



Below we compare IAI estimated from the two scenarios.

```
plotCCC(fit_allPoss_1$CI$IAI, fit_allPoss_2$CI$IAI)
```

Concordance plot: CCC=0.967, Corr=1.000



##	ccc	s_shift	l_shift	ccc_lo	ccc_hi	Cb	corr	mean_x
##	0.96701	0.78723	-0.09956	0.96324	0.97040	0.96739	0.99961	2.00115
##	mean_y	sd_x	sd_y					
##	1.67744	3.67464	2.89279					

9 GUI Usage

GUI interface has been shipped with *drexplore* which is built upon the *fgui*. After loading the *drexplore* package, typing

```
drexploreGUI.1()
```

in the R console will bring out the GUI for fitting dose-response curves. Similarly, typing

```
drexploreGUI.2()
```


will bring out the GUI for drug-drug interaction analysis. In both case, example data sets have been attached.

10 Session Info

- R version 3.1.0 (2014-04-10), x86_64-unknown-linux-gnu
- Locale: LC_CTYPE=en_US.UTF-8, LC_NUMERIC=C, LC_TIME=en_US.UTF-8, LC_COLLATE=en_US.UTF-8, LC_MONETARY=en_US.UTF-8, LC_MESSAGES=en_US.UTF-8, LC_PAPER=en_US.UTF-8, LC_NAME=C, LC_ADDRESS=C, LC_TELEPHONE=C, LC_MEASUREMENT=en_US.UTF-8, LC_IDENTIFICATION=C
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, tools, utils
- Other packages: abind 1.4-0, calibrate 1.7.2, car 2.0-21, DoseFinding 0.9-11, drc 2.3-96, drexplorer 0.1.0, epiR 0.9-58, fgui 1.0-5, gtools 3.4.1, knitr 1.6, lattice 0.20-29, magic 1.5-6, MASS 7.3-33, mvtnorm 1.0-0, plotrix 3.5-7, plyr 1.8.1, RColorBrewer 1.0-5, scales 0.2.4, stringr 0.6.2, survival 2.37-7
- Loaded via a namespace (and not attached): colorspace 1.2-4, evaluate 0.5.5, formatR 1.0, grid 3.1.0, highr 0.3, munsell 0.4.2, nnet 7.3-8, Rcpp 0.11.2, tcltk 3.1.0