Package: braidrm (via r-universe)

August 24, 2024

Type Package

Title Fitting Dose Response with the BRAID Combined Action Model

Version 0.71

Date 2016-02-02

Author Nathaniel R. Twarog

Maintainer Nathaniel R. Twarog <nathaniel.twarog@stjude.org>

Depends stats

Description Contains functions for evaluating, analyzing, and fitting combined action dose response surfaces with the Bivariate Response to Additive Interacting Dose (BRAID) model of combined action.

License GPL (>= 3)

Repository https://nathanieltwarog.r-universe.dev

RemoteUrl https://github.com/nathanieltwarog/braidrm

RemoteRef HEAD

RemoteSha 46821550f50393fa2bde6a463e211cb6e26761ae

Contents

`braidrm-package` *Fitting Dose Response with the BRAID Combined Action Model*

Description

Contains functions necessary for evaluating, analyzing, and fitting combined action dose response surfaces with the Bivariate Response to Additive Interacting Dose (BRAID) model of combined action.

Details

Package:	braidrm
Type:	Package
Version:	0.71
Date:	2016-03-07
License:	GPL (>= 3)

A mathematical description of the BRAID surface model can be found in `evalBRAIDrsm`. The heart of the package is the function `braidrm`, which takes a set of paired concentrations and measured responses and fits a particular instantiation of the BRAID model to the data. Bootstrapped confidence intervals on all fit parameters can be constructed by `braidrm`, or afterwards by the function `getBRAIDbootstrap`. Known BRAID surfaces can be evaluated or inverted using the calculating functions `evalBRAIDrsm` and `invertBRAIDrsm`. As analysis of combined dose response is closely linked to single agent dose response, we have provided several simple functions for analyzing such relationships, such as `findBestHill`. Finally, we also provide the more complete analytic functions `findBestBRAID` and `runBRAIDanalysis`, which perform a series of BRAID surface fits (and in the case of `runBRAIDanalysis`, single agent fits) to simplify the process of implementing a BRAID analysis pipeline.

Author(s)

Nathaniel R. Twarog <nathaniel.twarog@stjude.org>

braidrm *BRAID Synergy Dose-Response Modeling*

Description

Calculates the best fit BRAID response surface given the concentrations of two drugs and the response of a measured target to the combination of those drugs.

Usage

```
## Default S3 method:
braidrm(model,data,getCIs=TRUE,fixed="kappa2",startparv=NULL,llims=NULL,ulims=NULL,...)
## S3 method for class 'formula'
braidrm(model,data,...)
## S3 method for class 'braidrm'
print(x, ...)
## S3 method for class 'braidrm'
summary(object, ...)
```

Arguments

model	a two-column array containing concentrations of Drug 1 and Drug 2 in each dose pair, or a symbolic formula (e.g. act ~ conc1+conc2) specifying which variables are to be fit
data	if model is an array, a vector of measurements of response to the concentrations of Drug 1 and Drug 2; if model is a formula, a data frame containing the columns specified in formula
getCIs	determines if confidence intervals will be calculated for all response surface parameters being fit. Parameters are fit using a bootstrapping approach which resamples residuals and refits the new data.
fixed	parameter specifying which parameters of the full BRAID model will be fit. See Details for a full description of this highly critical parameter
startparv	an optional parameter specifying starting parameter values for the optimization. The relationship between startparv and fixed is rather subtle, and is discussed below in the Details section.
llims	a vector of lower limits on parameters being fit. May have length 10, or length equal to the number of free parameters being fit. Any parameters that do not require a limit can have a value of NA. If NULL (the default), llims is calculated from the starting values in startparv (or the values calculated for startparv if startparv is not specified).
ulims	a vector of upper limits on parameters being fit. Follows same behavior as llims.
x	the object of class "braidrm" to be printed
object	the object of class "braidrm" to be summarized
...	Not used

Details

This function is designed to give as much control as is reasonably possible over which parameters are optimized and how the optimization behaves. However, implementing this much control can be quite complicated, and despite our efforts to make the function intuitive and transparent, the interface is still quite unwieldy. A great deal can be accomplished with the function simply using the model nicknames "kappa1" and "kappa2" (which constrain the two maximal effects to be equal or allow them to vary independently, respectively); if these are sufficient for you, feel free to ignore the remainder of this section.

The parameter `fixed` is used to control which parameters are fit by the optimization, and in the case of the three maximal effect parameters, how the parameters are constrained with respect to one another. `fixed` may be specified in one of three forms: a raw vector, an index vector, or a model nickname. There are seven model nicknames; using them is equivalent to using the corresponding index vector as specified in the following table:

kappa1	1,2,3,4,6,7,10
kappa2 (default)	1,2,3,4,6,7,8,9
kappa3	1,2,3,4,6,7,8,9,10
delta1	1,2,3,4,5,7,10
delta2	1,2,3,4,5,7,8,9
delta3	1,2,3,4,5,7,8,9,10
ebraid	1,2,3,4,5,6,7,8,9,10

An index vector specifies which parameters vary in the optimization by listing their indices in the full ten parameter vector; a raw vector specifies which parameters vary by setting the corresponding values equal to `NA`. The remaining values in a raw vector specify the values at which the fixed parameters are fixed, unless these values are overridden by `startparv`

`startparv` specifies the starting values for the optimization; one can input a vector in `startparv` that specifies only the values of varying parameters, but the remaining values must be specified in a raw vector in `fixed`. If a full-length ten-element vector is provided for `starparv`, the values of fixed parameters are taken from that vector, regardless of the type of input in `fixed`.

For parameters one through seven, the presence or absence of each parameter in the optimization is quite simple: either the parameter is fixed at the specified (or calculated) value, or it varies between the specified (or calculated) optimization limits. Parameters eight through ten, however, which specify the maximal effect parameters, are more complicated. The complication is introduced by the fact that the parameters can be fixed *or* constrained to be equal to one another, which introduce the same number of degrees of freedom, but very different optimization behaviors. How the different possibilities for `fixed` and `startparv` influence the optimization behavior (for as many possibilities as we could think of) is described by the following table. In the table W, X, Y, and Z represent arbitrary but distinct valid values while ~ represents any valid (non-`NA`) value; in addition, all `startparv` vectors are assumed to be complete ten-element vectors (incomplete vectors will simply be extended with the corresponding values in `fixed`).

fixed	**startparv**	**Behavior**
Index: (. . .,8,9,10) or Raw: (. . .,NA,NA,NA)	Any	All three maximal effect parameters vary independently, with E_f constrained to be a larger effect than $E_{f,A}$ and $E_{f,B}$.

Index: (. . .,8,9) or Raw: (. . .,NA,NA,~)	NULL	$E_{f,A}$ and $E_{f,B}$ vary independently and E_f is constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,8,9) or Raw: (. . .,NA,NA,~)	(. . ., X, X, X) or (. . ., X, Y, X) or (. . ., Y, X, X)	$E_{f,A}$ and $E_{f,B}$ vary independently and E_f is constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,8,9) or Raw: (. . .,NA,NA,~)	(. . ., X, X, Y) or (. . ., X, Z, Y)	E_f is fixed at the value Y, while $E_{f,A}$ and $E_{f,B}$ vary independently and are constrained to be smaller effects than Y.
Index: (. . .,10) or Raw: (. . .,X,X,NA)	NULL	All maximal parameters are constrained to vary as a single parameter.
Raw: (. . .,X,Y,NA)	NULL	If X is larger than Y, E_f and $E_{f,A}$ are constrained to vary as a single parameter above Y, and $E_{f,B}$ is fixed at Y. Otherwise, the roles of A and B are reversed.
Index: (. . .,10) or Raw: (. . .,~,~,NA)	(. . ., X, X, X)	All maximal parameters are constrained to vary as a single parameter.
Index: (. . .,10) or Raw: (. . .,~,~,NA)	(. . ., X, Y, X)	E_f and $E_{f,A}$ are constrained to vary as a single parameter above Y, and $E_{f,B}$ is fixed at Y.
Index: (. . .,10) or Raw: (. . .,~,~,NA)	(. . ., Y, X, X)	E_f and $E_{f,B}$ are constrained to vary as a single parameter above Y, and $E_{f,A}$ is fixed at Y.
Index: (. . .,10) or Raw: (. . .,~,~,NA)	(. . ., X, Y, Z)	E_f is constrained to vary above the larger effect of X and Y, $E_{f,A}$ is fixed at X, and $E_{f,B}$ is fixed at Y.
Index: (. . .,8,10) or Raw: (. . .,NA,~,NA)	NULL or (. . ., ~, X, X)	$E_{f,A}$ varies freely, and $E_{f,B}$ and E_f are constrained to vary as a single parameter with a larger effect than $E_{f,A}$.
Index: (. . .,8,10) or Raw: (. . .,NA,~,NA)	(. . ., ~, X, Y)	$E_{f,A}$ varies freely, $E_{f,B}$ is fixed at X and E_f varies freely constrained to be a larger effect than $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,9,10) or Raw: (. . .,~,NA,NA)	NULL or (. . ., X, ~, X)	$E_{f,B}$ varies freely, and $E_{f,A}$ and E_f are constrained to vary as a single parameter with a larger effect than $E_{f,B}$.
Index: (. . .,9,10) or Raw: (. . .,~,NA,NA)	(. . ., X, ~, Y)	$E_{f,B}$ varies freely, $E_{f,A}$ is fixed at X and E_f varies freely constrained to be a larger effect than $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,8) or Raw: (. . .,NA,X,X)	NULL	$E_{f,A}$ varies freely, $E_{f,B}$ is fixed at X (or calculated starting value), and E_f is constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Raw: (. . .,NA,X,Y)	NULL	$E_{f,B}$ is fixed at X, E_f is fixed at Y, and $E_{f,A}$ varies freely constrained to be a smaller effect than Y.
Index: (. . .,8) or Raw: (. . .,NA,~,~)	(. . ., X, X, X) or (. . ., Y, X, X) or (. . ., Y, X, Y)	$E_{f,A}$ varies freely, $E_{f,B}$ is fixed at X, and E_f is constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,8) or Raw: (. . .,NA,~,~)	(. . ., X, X, Y) or (. . ., Z, X, Y)	$E_{f,B}$ is fixed at X, E_f is fixed at Y, and $E_{f,A}$ varies freely constrained to be a smaller effect than Y.
Index: (. . .,9) or Raw: (. . .,X,NA,X)	NULL	$E_{f,B}$ varies freely, $E_{f,A}$ is fixed at X (or calculated starting value), and E_f is constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Raw: (. . .,X,NA,Y)	NULL	$E_{f,A}$ is fixed at X, E_f is fixed at Y, and $E_{f,B}$ varies freely constrained to be a smaller effect than Y.
Index: (. . .,9) or	(. . ., X, X, X) or	$E_{f,B}$ varies freely, $E_{f,A}$ is fixed at X, and E_f is

Raw: (. . .,~,NA,~)	(. . ., X, Y, X) or (. . ., X, Y, Y)	constrained to vary with the larger effect of $E_{f,A}$ and $E_{f,B}$.
Index: (. . .,9) or Raw: (. . .,~,NA,~)	(. . ., X, X, Y) or (. . ., X, Z, Y)	$E_{f,A}$ is fixed at X, E_f is fixed at Y, and $E_{f,B}$ varies freely constrained to be a smaller effect than Y.
Index: (. . .) or Raw: (. . .,~,~,~)	Any	All three values are fixed, and do not vary.

Value

An object of class 'braidrm', including the elements

`conc1`	Concentrations of drug 1 used in the fit
`conc2`	Concentrations of drug 2 used in the fit
`act`	Reponse to drugs 1 and 2 used in the fit
`fitted.values`	Value of best-fit response surface at concentrations of drugs 1 and 2 used in the fit
`residuals`	Difference between actual measurements and fitted values
`ostart`	The full 10-parameter starting parameter vector used in the optimization
`fixed`	Vector describing which parameters were fit
`mlims`	Array containing upper and lower parameter bounds used in the optimization
`coefficients`	Estimates of fitted parameters
`fullpar`	Full parameter vector including best-fit parameters
`convergence`	From `optim`, indicates if optimization successfully converged
`message`	From `optim`, describes results of non-linear optimization
`call`	Function call

If `getCIs` is TRUE, then the following elements are also included

`ciPass`	TRUE if a sufficient proportion of bootstrapping trials successfully converged, FALSE otherwise
`ciLevs`	Two-element vector specifying the lower and upper percentiles of the desired confidence interval. Defaults to 0.025 and 0.975 for a 95% confidence interval.
`ciVec`	Assuming 'ciPass' is true, a vector containing the lower and upper bounds on the confidence intervals of all free parameters
`bCoefs`	A matrix containing the best-fit parameter values from all bootstrapping trials. Useful for calculating confidence intervals on other values, such as EC99

Author(s)

Nathaniel R. Twarog

See Also

`getBRAIDbootstrap`, `calcBRAIDconfint`, `evalBRAIDrsm`

Examples

```
data(es8olatmz)
summary(braidrm(act~conc1+conc2,es8olatmz,getCIs=FALSE))
## Not run:
summary(braidrm(cbind(es8olatmz$conc1,es8olatmz$conc2),es8olatmz$act))
summary(braidrm(act~conc1+conc2,es8olatmz,fixed="delta2"))
summary(braidrm(act~conc1+conc2,es8olatmz,fixed=c(1,2,3,4,6,8,9)))
summary(braidrm(act~conc1+conc2,es8olatmz,llims=c(NA,NA,NA,NA,NA,NA,NA,-4,-4,-4)))

## End(Not run)
```

---

calcBRAIDconfint *Calculate General BRAID Confidence Intervals*

Description

Produces confidence intervals for any general calculable property of a BRAID surface, based on a given BRAID surface fit (with bootstrapped coefficients).

Usage

```
calcBRAIDconfint(bfit, parfunc, civals = NULL)
```

Arguments

`bfit`	A BRAID surface fit of class `braidrm`, with bootstrapped coefficients
`parfunc`	A function to calculate a property of a response surface. `parfunc` must take a ten-element BRAID parameter vector as input, and return a single numeric value or a numeric vector of a fixed length as output.
`civals`	Values specifying the lower and upper bound of the confidence interval(s) to be calculated. If `NULL` (the default), the values of the `ciLevs` property of `bfit` will be used. NOTE: using a wider percentile here than was used to estimate the bootstrapped coefficients can lead to an unstable confidence interval.

Value

If `parfunc` produces a single value as output, a vector of three values: the second value is the estimate of the best BRAID surface fit, while the first and third values represent the lower and upper bounds of the calculated confidence interval. If `parfunc` produces a vector as output, an array of width three: the second column is the estimate of the best BRAID surface fit, while the first and third columns represent the lower and upper bounds of the calculated confidence intervals.

Author(s)

Nathaniel R. Twarog

See Also

braidrm,getBRAIDbootstrap

Examples

```
data(es8olatmz)

brd <- braidrm(act~conc1+conc2,es8olatmz,fixed=c(NA,NA,NA,NA,1,NA,NA,-4,-4,-4))
potci <- calcBRAIDconfint(brd,function(parv) invertBRAIDrsm(-1,DA=10^-7,parv=parv))
## Not run:
pconc <- exp(seq(log(10^-6),log(10^-3),length=100))
effci <- calcBRAIDconfint(brd,function(parv) evalBRAIDrsm(10^-7,pconc,parv))

## End(Not run)
```

---

es1data *Results from Ewings-Sarcoma Experiments*

Description

These data frames contain the results of a series of cytotoxicity experiments on cell-lines from the pediatric cancer Ewings' sarcoma. Data frames describe results from several different experiments or sets of experiments:

es1data Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the ES1 cell line.

es1extdata Follow up experiments combining BMN 673 and SN-38 with several additional mechanistic classes in the ES1 cell line.

es8data Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the ES8 cell line.

es8extdata Follow up experiments combining BMN 673 and SN-38 with several additional mechanistic classes in the ES8 cell line.

ew8data Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the EW8 cell line.

Usage

```
data(ewings)
```

Format

Data frames containing the following columns:

plate The number of the plate this measurement was taken from

well The name of the well (e.g. "A01") this measurement was taken from

compound1 The compound drugged in decreasing concentration across this plate (DMSO if none)

compound2 The compound drugged in decreasing concentration down this plate (DMSO if none)

conc1 The concentration (in molar) of the first compound in this well

conc2 The concentration (in molar) of the second compound in this well

act The base-10 logarithm of CellTiter-Glo intensity in this well normalized to intensity in negative control wells on the same plate

date The date on which the experiment using this plate was run

---

`es1plates` *Plate Information from Ewings-Sarcoma Experiments*

Description

These data frames describe the compounds used in each of plates tested in a series of cytotoxicity experiments on cell-lines from the pediatric cancer Ewings' sarcoma. Data frames describe plates from several different experiments or sets of experiments:

es1plates Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the ES1 cell line.

es1extplates Follow up experiments combining BMN 673 and SN-38 with several additional mechanistic classes in the ES1 cell line.

es8plates Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the ES8 cell line.

es8extplates Follow up experiments combining BMN 673 and SN-38 with several additional mechanistic classes in the ES8 cell line.

ew8plates Experiment combining three PARP inhibitors with standard-of-care DNA damaging agents in the EW8 cell line.

Usage

```
data(ewings)
```

Format

Data frames containing the following columns:

plate The number of a plate run in a particular experiment

compound1 The compound drugged in decreasing concentration across the plate (DMSO if none)

compound2 The compound drugged in decreasing concentration down the plate (DMSO if none)

posmean The average base-10 logarithm of CellTiter-Glo intensity in positive control wells on the plate, normalized to intensity in negative control wells on the same plate

date The date on which the experiment using that plate was run

es8olatmz *Results from Ewings-Sarcoma Experiments*

Description

This data frames contain the results of a single combination of the PARP inhibitor olaparib and the DNA-damaging agent temozolomide in a cytotoxicity experiment on the Ewings' sarcoma cell line ES8 run on April 4, 2014.

Usage

```
data(es8olatmz)
```

Format

A data frame containing the following columns:

plate The number of the plate this measurement was taken from

well The name of the well (e.g. "A01") this measurement was taken from

compound1 The compound drugged in decreasing concentration across this plate (DMSO if none)

compound2 The compound drugged in decreasing concentration down this plate (DMSO if none)

conc1 The concentration (in molar) of the first compound in this well

conc2 The concentration (in molar) of the second compound in this well

act The base-10 logarithm of CellTiter-Glo intensity in this well normalized to intensity in negative control wells on the same plate

date The date on which the experiment using this plate was run

evalBRAIDrsm *Calculate BRAID Surface Values*

Description

Calculates the value of the Bivariate Response to Additive Interacting Doses (BRAID) surface model for the given concentration pairs.

Usage

```
evalBRAIDrsm(DA, DB, parv)
```

Arguments

DA	vector of concentrations of drug A
DB	vector of concentrations of drug B
parv	ten-element vector specifying the full set of parameters for the BRAID surface (see Details below)

Details

The full ten-parameter BRAID model, which we refer to as the *extended BRAID* or *eBRAID* model is defined as:

$$E(D_A, D_B) = E_0 + \frac{E_f - E_0}{1 + \tilde{D}_{AB}^{-\delta\sqrt{n_a n_b}}}$$

$$\tilde{D}_{AB} = \tilde{D}_A^{\frac{1}{\delta\sqrt{n_a n_b}}} + \tilde{D}_B^{\frac{1}{\delta\sqrt{n_a n_b}}} + \kappa\sqrt{\tilde{D}_A^{\frac{1}{\delta\sqrt{n_a n_b}}}\,\tilde{D}_B^{\frac{1}{\delta\sqrt{n_a n_b}}}}$$

$$\tilde{D}_A = \frac{\left(\frac{E_{f,A}-E_0}{E_f-E_0}\right)\left(\frac{D_A}{ID_{M,A}}\right)^{n_a}}{1 + \left(1 - \frac{E_{f,A}-E_0}{E_f-E_0}\right)\left(\frac{D_A}{ID_{M,A}}\right)^{n_a}}$$

$$\tilde{D}_B = \frac{\left(\frac{E_{f,B}-E_0}{E_f-E_0}\right)\left(\frac{D_B}{ID_{M,B}}\right)^{n_b}}{1 + \left(1 - \frac{E_{f,B}-E_0}{E_f-E_0}\right)\left(\frac{D_B}{ID_{M,B}}\right)^{n_b}}$$

The parameters of this equation must satisfy $n_a > 0$, $n_b > 0$, $\delta > 0$, $\kappa > -2$, $sign(E_f - E_0) = sign(E_{f,A} - E_0) = sign(E_{f,B} - E_0)$, $|E_f - E_0| \geq |E_{f,A} - E_0|$, and $|E_f - E_0| \geq |E_{f,B} - E_0|$. With this definition, the ten-element parameter vector is [$ID_{M,A}$, $ID_{M,B}$, n_a, n_b, δ, κ, E_0, $E_{f,A}$, $E_{f,B}$, E_f]. The simpler standard BRAID model, as described in Twarog *et al.* is obtained by setting δ equal to 1 and setting E_f such that $|E_f - E_0|$ is equal to the maximum of $|E_{f,A} - E_0|$ and $|E_{f,B} - E_0|$. Assuming that this sets E_f equal to $E_{f,A}$, this causes the equation for $\tilde{D}_A$ to simplify to

$$\tilde{D}_A = \left(\frac{D_A}{ID_{M,A}}\right)^{n_a}$$

Value

A vector of response values corresponding to the pairs of concentrations in `DA` and `DB`

Author(s)

Nathaniel R. Twarog

See Also

`braidrm`, `invertBRAIDrsm`

Examples

```
conc1 <- rep(seq(0,3*10^-6,length=50),each=50)
conc2 <- rep(seq(0,3*10^-6,length=50),times=50)

# Additive surface
act <- evalBRAIDrsm(conc1,conc2,parv=c(10^-6,10^-6,1.5,1.5,1,0,0,100,100,100))
# A BRAID additive surface is not a Loewe additive surface
act <- evalBRAIDrsm(conc1,conc2,parv=c(10^-6,10^-6,1,3,1,0,0,100,100,100))
# BRAID antagonism
act <- evalBRAIDrsm(conc1,conc2,parv=c(10^-6,10^-6,1.5,1.5,1,-1,0,100,100,100))
# delta-BRAID synergy
```

```
act <- evalBRAIDrsm(conc1,conc2,parv=c(10^-6,10^-6,1.5,1.5,1.75,0,0,100,100,100))
# Differing final effects
act <- evalBRAIDrsm(conc1,conc2,parv=c(10^-6,10^-6,1.5,1.5,1,0,0,75,100,100))
```

`evalHillEqn` *Calculate and Invert Hill Equation*

Description

Calculates the response of a four-parameter Hill (or log-logistic) dose-response model, or determines the concentrations which yield the given response.

Usage

```
evalHillEqn(conc, parv)
invertHillEqn(val, parv)
```

Arguments

`conc`	a vector of concentrations to be passed as input into the model
`parv`	a four-element vector specifying the full parameter set of a particular Hill model
`val`	a vector of responses to be inverted in a particular Hill model

Details

There is some ambiguity in how the parameters of a Hill or log-logistic model are specified. We have chosen to use the following equation:

$$E(D) = E_0 + \frac{E_f - E_0}{1 + \left(\frac{D}{ID_M}\right)^{-n}}$$

where the Hill slope n is always positive. Using this equation, the parameter vector for a Hill model is (n,E_0,E_f,$lnID_M$) (note that the fourth parameter value is log-transformed in the parameter vector).

Value

For `evalHillEqn` a vector of responses resulting from the given concentrations. For `invertHillEqn`, a vector of concentrations that produce the given responses; responses beyond the model's maximal effect will produce a value of `Inf`, while responses that do not reach the model's baseline effect produce a value of 0.

Author(s)

Nathaniel R. Twarog

See Also

findBestHill, hillConcCorrect

Examples

```
act <- evalHillEqn(exp(seq(log(10^-8),log(10^-5),length=100)),parv=c(2,0,100,log(10^-6)))
ec90 <- invertHillEqn(90,parv=c(2,0,100,log(10^-6)))
```

---

findBestBRAID *Select Best Fitting BRAID Surface Model*

Description

Fits several BRAID surface models to the given data, and selects the most parsimonious model using the Akaike information criterion.

Usage

```
## Default S3 method:
findBestBRAID(model, data, defaults, startparv=NULL, llims=NULL,
    ulims=NULL, itype=1, getCIs=TRUE, crossval=TRUE, ...)
## S3 method for class 'formula'
findBestBRAID(model, data, ...)
```

Arguments

model	a two-column array containing concentrations of Drug 1 and Drug 2 in each dose pair, or a symbolic formula (e.g. act ~ conc1+conc2) specifying which variables are to be fit
data	if model is an array, a vector of measurements of response to the concentrations of Drug 1 and Drug 2; if model is a formula, a data frame containing the columns specified in formula
defaults	two-element vector specifying the default initial and maximal effects for the response surface. These values will be used in several of the models that are tried (see Details below).
startparv	an optional parameter specifying starting parameter values for the optimization
llims	a ten-element vector of lower limits on parameters being fit. Any parameters that do not require a limit can have a value of NA. If NULL (the default), llims is calculated from the starting values in startparv (or the values calculated for startparv if startparv is not specified).
ulims	a vector of upper limits on parameters being fit. Follows same behavior as llims.
itype	an integer that specifies the type of interaction(s) that is assumed in the models. The default is 1, which assumes that the interaction parameter κ is varying. See details below for other possible values.

`getCIs`	determines if confidence intervals will be calculated for all response surface parameters being fit. Parameters are fit using a bootstrapping approach which resamples residuals and refits the new data.
`crossval`	if `TRUE`, goodness of fit is determined by randomly assigning data points to four blocks, and evaluating goodness of fit on each block by fitting the remaining three. If `FALSE`, all data points are fit once, and goodness of fit is determined using the residuals from that fit. It is advisable not to use cross-validation when a relatively small number of data points are available, especially on the margins (when drug A or drug B has concentration 0).
...	Not used

Details

Because experiments do not reliably capture the full range of responses that a combination can produce, estimation of the initial and maximal effect parameters for a BRAID surface can be highly unstable. This function fits at least 10 distinct BRAID surface models to the given data, and selects the model which best balances simplicity with goodness of fit. For each interaction type (see below), the following 10 models are fit:

- The initial effect parameter E_0 varies freely, and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ vary freely independently of one another.
- The initial effect parameter E_0 is fixed at the default value, and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ vary freely independently of one another.
- The initial effect parameter E_0 the maximal effect parameter $E_{f,A}$ vary freely, and the maximal effect parameter $E_{f,B}$ is fixed at the default value.
- The initial effect parameter E_0 the maximal effect parameter $E_{f,B}$ vary freely, and the maximal effect parameter $E_{f,A}$ is fixed at the default value.
- The initial effect parameter E_0 varies freely, and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ are constrained to vary as a single parameter E_f.
- The initial effect parameter E_0 varies freely, and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ are fixed at the default value.
- The initial effect parameter E_0 the two maximal effect parameter $E_{f,B}$ are fixed at the default values, and the maximal effect parameter $E_{f,A}$ varies freely.
- The initial effect parameter E_0 the two maximal effect parameter $E_{f,A}$ are fixed at the default values, and the maximal effect parameter $E_{f,B}$ varies freely.
- The initial effect parameter E_0 is fixed at the default value,, and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ are constrained to vary as a single parameter E_f.
- The initial effect parameter E_0 and the two maximal effect parameters $E_{f,A}$ and $E_{f,B}$ are all fixed at the default values.

In all models, the potencies of the two drugs (represented by $ID_{M,A}$ and $ID_{M,B}$) and the Hill slopes of both drugs (represented by n_a and n_b) vary freely. Which of the interaction parameters κ and δ varies depends on the parameter `itype`, as follows:

- `itype = 1`: κ varies freely in all models; δ is fixed at 1 (10 models total).
- `itype = 2`: δ varies freely in all models; κ is fixed at 0 (10 models total).

- itype = 3: Either κ or δ (but not both) vary freely in all models (20 models total).
- itype = 4: Either κ or δ or both vary freely in all models (30 models total).
- itype = 5: κ is fixed at 0 and δ is fixed at 1 in all models (10 models total).

Value

An object of the class 'braidrm', with elements as described in braidrm.

Author(s)

Nathaniel R. Twarog

See Also

braidrm, getBRAIDbootstrap, runBRAIDanalysis

Examples

```
data(es8olatmz)
## Not run: summary(findBestBRAID(cbind(es8olatmz$conc1,es8olatmz$conc2),
    es8olatmz$act,defaults=c(0,-2.7)))
## End(Not run)
## Not run: summary(findBestBRAID(act~conc1+conc2,es8olatmz,defaults=c(0,-2.7),itype=2))
summary(findBestBRAID(act~conc1+conc2,es8olatmz,defaults=c(0,-4),getCIs=FALSE))
```

---

findBestHill *Fit Hill Model with Model Selection*

Description

Fits the given concentration and response data with several Hill (or log-likelihood) models, and selects the model that best balances simplicity and accuracy according to the Akaike Information Criterion.

Usage

```
## Default S3 method:
findBestHill(model,data,defaults,startparv=NULL,llims=NULL,ulims=NULL,...)
## S3 method for class 'formula'
findBestHill(model,data,...)
```

Arguments

model	a vector of concentrations, or a symbolic formula (e.g. act ~ conc) specifying which variables are to be fit
data	if model is a vector, a vector of measurements of response to the concentrations set down in conc; if model is formula, a data frame containing the columns specified in formula

defaults	two-element vector specifying the default initial and maximal effects for the Hill model. These values will be used in several of the models that are tried (see Details below).
startparv	an optional parameter specifying starting parameter values for the optimization. Any parameters that do not have a starting value can have a value of NA (and will be estimated from the concentrations and responses.
llims	a four-element vector of lower limits on parameters being fit. Any parameters that do not require a limit can have a value of NA. If NULL (the default), llims is calculated from the starting values in startparv (or the values calculated for startparv if startparv is not specified).
ulims	a vector of upper limits on parameters being fit. Follows same behavior as llims.
...	Not used

Details

As many dose response experiments fail to capture the full sigmoidal behavior of a compound, estimation of the initial and maximal effect parameters for a four-parameter Hill equation (see evalHillEqn for a description of the equation used in this package) can be quite unstable. To avoid estimation of wildly divergent parameters and overfitting, this function fits a four-parameter Hill equation to the given data four times, with the initial and maximal effect parameters either varying freely or fixed to the provided default values. The most parsimonious model is selected from the four fits according to the Akaike Information Criterion. The model also fits a constant flat function to the data; though this fifth "model" is not included in the model selection, it is included in the output for reference, as a check that the given data shows any signal at all.

Value

A list describing the results of the model fits, with the following components:

conc	The vector of concentrations used in the fit
act	The vector of responses used in the fit
bestModIdx	The index of the model among those fit which yielded the most parsimonious fit
bestModName	The name of the best-fit model. The names of the models are 1. "m2p": A model in which both the initial and maximal effect parameters are fixed to the default values 2. "m3puc": A model in which the initial effect parameter varies freely 3. "m3plc": A model in which the maximal effect parameter varies freely 4. "m4p": A model in which both the initial and maximal effect parameters vary freely The fifth "model", a constant linear fit, is named "Lin", but will not be selected by the model selection.
mlims	A matrix of upper and lower limits applied to the four Hill parameters in the fit
allfits	A list of the five fits (including the trivial linear fit) applied to the data. Each fit is described by a list with the following components:

- coefficients: The full four-element parameter vector describing the best-fit Hill equation in this model (or the single parameter for the constant linear fit)
- parv: The vector of best-fit values for only those parameters which varied freely in the model
- pinds: A vector of parameter indices indicating which of the four Hill parameters varied in this model
- AIC: The Akaike Information Criterion value for this model

Author(s)

Nathaniel R. Twarog

See Also

evalHillEqn, getHillBootstrap, runBRAIDanalysis

Examples

```
data(es8olatmz)
drv <- es8olatmz$compound2=="DMSO"
hfit <- findBestHill(es8olatmz$conc1[drv],es8olatmz$act[drv],defaults=c(0,-2.7))
coef(hfit$allfits[[hfit$bestModIdx]])
hfit <- findBestHill(act~conc1,es8olatmz[drv,],defaults=c(0,-4))
coef(hfit$allfits[[hfit$bestModIdx]])
```

---

getBRAIDbootstrap *Bootstrap BRAID Parameter Confidence Intervals*

Description

Estimate bootstrapped confidence intervals of a BRAID surface fit's parameters, by sampling residuals and refitting.

Usage

```
getBRAIDbootstrap(bfit, ciLevs = c(0.025, 0.975),numBoot=NULL)
```

Arguments

bfit	a BRAID surface fit of class 'braidrm'
ciLevs	a two-element vector specifying the lower and upper percentiles of the desired confidence interval. The default value of c(0.025, 0.975) results in a 95% confidence interval.
numBoot	the number of bootstrapped coefficients to be calculated. If NULL (the default), number of bootstrapped coefficients is determined from the width of the interval.

Details

This function constructs bootstrapped confidence intervals for objects of class `braidrm`. If the input already has bootstrapped confidence intervals (or has tried and failed to generate them), this function will throw a warning and return the input model.

Value

An object of class `braidrm`, with the new elements:

`ciPass`	TRUE if a sufficient proportion of bootstrapping trials successfully converged, FALSE otherwise
`ciLevs`	Two-element vector specifying the lower and upper percentiles of the desired confidence interval. Defaults to 0.025 and 0.975 for a 95% confidence interval.
`ciVec`	Assuming 'ciPass' is true, a vector containing the lower and upper bounds on the confidence intervals of all free parameters
`bCoefs`	Array containing the best-fit parameter values from all bootstrapping trials. Useful for calculating confidence intervals on other values.

Author(s)

Nathaniel R. Twarog

See Also

`braidrm`, `calcBRAIDconfint`

Examples

```
data(es8olatmz)

brd <- braidrm(act~conc1+conc2,es8olatmz,getCIs=FALSE,fixed=c(NA,NA,NA,NA,1,NA,NA,-4,-4,-4))
summary(getBRAIDbootstrap(brd))
## Not run: summary(getBRAIDbootstrap(brd,ciLevs=c(0.05,0.95)))
```

`getHillBootstrap` *Bootstrap Hill Equation Parameter Confidence Intervals*

Description

Uses sampling residuals with replacement to bootstrap a distribution of Hill equation parameters for the construction of confidence intervals on those parameters.

Usage

```
getHillBootstrap(hfitl, ciLevs = c(0.025,0.975), mi = NULL)
```

Arguments

hfitl	a list of Hill equation model fits produced by findBestHill
ciLevs	a two-element vector specifying the lower and upper percentiles of the desired confidence intervals. Default is c(0.025,0.975), which yields a 95% confidence interval.
mi	index of the desired model to bootstrap and construct confidence intervals on. If NULL (the default), the value of bestModIdx in hfitl will be used.

Value

A list of the same form as produced by findBestHill, with the following additional components:

ciPass	TRUE if at least 50% of bootstrapping trials result in a converged fit
ciLevs	Equal to the input parameter ciLevs
ciMInd	Index of the model used to construct the bootstrapped confidence interval
ciVec	Vector of lower and upper bounds of confidence intervals on all freely varying parameters, in order
bCoefs	Four column array containing complete Hill equation parameters for all resampled fits

Author(s)

Nathaniel R. Twarog

See Also

findBestHill, evalHillEqn

Examples

```
data(es8olatmz)
drv <- es8olatmz$compound2=="DMSO"
hfit <- findBestHill(act~conc1,es8olatmz[drv,],defaults=c(0,-2.7))
hfit1 <- getHillBootstrap(hfit)

# Setting 'mi' to 4 constructs a confidence interval on the four-
# parameter Hill model, overriding the 'best' model index
hfit2 <- getHillBootstrap(hfit,ciLevs=c(0.05,0.95),mi=4)
```

`hillConcCorrect` *Hill-Based Concentration Correction*

Description

Estimates actual underlying concentrations leading to a given set of response measurements based on the assumption that actual concentrations are log-normally distributed around target concentrations, response errors are normally distributed, and the actual underlying relationship between concentration and response is represented by the given Hill dose-response model.

Usage

```
hillConcCorrect(conc, act, parv, sigr = 1)
```

Arguments

`conc`	a vector of expected or target concentrations, around which actual concentrations are assumed to be log-normally distributed
`act`	a vector of response values
`parv`	a four-parameter vector specifying a Hill model as described in `evalHillEqn` which is assumed to be the actual relationship between concentration and response
`sigr`	the estimated ratio of the noises in response- and log (base10) concentration-space

Details

Suppose that $\hat{c}$ is a given target concentration, and c is the actual concentration in given well, plate, or condition. Suppose also that y is the actual response that would result from the concentration in the given Hill dose-response model, and $\hat{y}$ is the measured response value. This function assumes that

$$\hat{y} \sim N(y, \sigma)$$

$$\log_{10} c \sim N(\log_{10} \hat{c}, \frac{\sigma}{r})$$

for some σ, where N is a normal distribution, and r is the ratio specified by the parameter `sigr`. Based on these assumptions, the function uses Bayes' rule to calculate the maximum likelihood estimate of c for every given value of $\hat{c}$ and $\hat{y}$.

Value

A vector of concentrations representing the maximum likelihood estimates of the actual concentrations which produced the given responses.

Author(s)

Nathaniel R. Twarog

See Also

evalHillEqn, runBRAIDanalysis

Examples

```
data(es8olatmz)
drv <- es8olatmz$compound2=="DMSO"
hfit <- findBestHill(act~conc1,es8olatmz[drv,],defaults=c(0,-2.7))
drvpos <- drv & es8olatmz$conc1>0
cconc <- hillConcCorrect(es8olatmz$conc1[drvpos],es8olatmz$act[drvpos],
   coef(hfit$allfits[[hfit$bestModIdx]]),sigr=1)
```

---

invertBRAIDrsm *Invert BRAID Response Function*

Description

Determines which drug concentrations, when paired with the given input concentrations, produce the given response values. Useful for estimating potentiation of one compound by presence of the other.

Usage

```
invertBRAIDrsm(val, DA=NULL, DB=NULL, parv)
```

Arguments

val	Effect value or values to be inverted. If a single value, it will be repeated for all values of DA or DB which are input.
DA	Concentrations of drug A to be inverted. If NULL (the default), the concentrations in DB will be inverted and the concentrations of drug A will be estimated.
DB	Concentrations of drug B to be inverted. If NULL (the default), the concentrations in DA will be inverted and the concentrations of drug B will be estimated.
parv	10-element vector specifying the full set of parameters for the BRAID surface

Value

A vector of concentrations which, when paired with the given concentrations, produce the given values. If DA is NULL, the vector contains the concentrations of drug A corresponding to the concentrations of drug B given in DB. If concB is null, the vector contains the concentrations of drug B corresponding to the concentrations of drug A give in DA. Exactly one of these two inputs must be NULL.

Author(s)

Nathaniel R. Twarog

See Also

braidrm, evalBRAIDrsm

Examples

```
level <- 90
doses <- c(0,10^-8,10^-7,10^-6)
ec90A <- invertBRAIDrsm(level,DB=doses,parv=c(10^-6,10^-6,1.5,1.5,1,1.8,0,100,100,100))
```

---

runBRAIDanalysis *Run Full BRAID Analysis*

Description

Performs a complete analysis of combined action data using the BRAID model. The behaviors of both drugs alone (if such doses are included) are fit using a Hill equation; if desired, the resulting Hill equation fits can be used to correct uncertain or noisy input concentrations for use in combination analysis. All measurements of the two drugs (in isolation or in combination) are then fit using the BRAID equation.

Usage

```
runBRAIDanalysis(data, defaults, llims = NULL, ulims = NULL, itype = 1,
     compounds = NULL, corrconc = FALSE, corrsigr = 1)
```

Arguments

data	a data frame containing all measurements to be fit. Must conatin columns named "conc1", "conc2", and "act"; other columns may be required (see Details below).
defaults	a two-element vector containing default values for the initial and maximal effects of the combination; used by findBestBRAID to select best BRAID model
llims	a ten-element vector of lower limits on parameters being fit. Any parameters that do not require a limit can have a value of NA.
ulims	a vector of upper limits on parameters being fit. Follows same behavior as llims.
itype	an integer that specifies the type of interaction(s) that is assumed in the models. Used by findBestBRAID
compounds	an optional parameter specifying which compounds are to be fit; this requires that compounds be identified in the input data frame data.
corrconc	boolean specifying whether input concentrations will be corrected according to individual dose response using hillConcCorrect
corrsigr	the ratio of noise in measurement to noise the base-10 logarithm of concentration; used by hillConcCorrect if corrconc is TRUE

Details

This convenience function is intended as a blueprint for a complete BRAID model analysis. Though users of this package can develop their own approaches to applying the BRAID model to combined action data (indeed we encourage it), this function encapsulates our overall strategy for fitting the BRAID equation. The input parameter `data` must contain all measurements to be fit, with the columns "conc1", "conc2", and "act" containing the corresponding inputs and outputs. If no other columns required (based on the values of other parameters), "conc1" is assumed to describe the concentration of drug A, and "conc2" the concentration of drug B, with values of 0 for "conc1" or "conc2" corresponding to conditions with Drug B or Drug A alone, respectively.

If `data` contains measurements from multiple combinations, the identity of the compounds to be analyzed must be specified in the parameter `compounds`. Further, the input data frame `data` must contain columns named "compound1" and "compound2" that specify which two compounds were used in each measurement. Measurements will be used in which "compound1" is equal to the first element of `compounds` or "conc1" is equal to 0, and in which "compound2" is equal to the second element of `compounds` or "conc2 is equal to 0. Note that this means measurements in which "conc1" and "conc2" are both equal to 0 will be used in any analysis.

For concentrations to be corrected in the combination analysis, there must be a correspondence between concentrations of each drug alone and concentrations used in combination; in our implementation, this is assumed to be the result of a shared drugging plate, such that measurements corresponding to the same well are given the same actual concentration of drug. As a result, if `corrconc` is set to `TRUE`, each measurement must be associated with a well specified in a column named "well".

Value

A list with the following componenents:

`concs`	A two-column array containing the concentrations of the first and second drugs used in the analysis. This will only inlcude measurements from the input frame `data` that contained drug A alone, drug B alone, or drug A and B in combination. Even if `corrconc` is `TRUE`, this array will contain the original, uncorrected concentrations. To access the corrected concentrations, use the `conc1` and `conc2` elements of the component `braidFit` (described below).
`act`	The measured response values corresponding to the concentration pairs in `concs`.
`corrconc`	The value of the input parameter `corrconc`.
`corrsigr`	The value of the input parameter `corrsigr`.
`braidFit`	The best BRAID model fit as determined by `findBestBRAID`.
`hfit1`	A set of Hill model fits to the behavior of drug A in isolation, as output by `findBestHill`. If no data points for drug A in isolation are included, this component is `NULL`.
`hfit2`	A set of Hill model fits to the behavior of drug B in isolation, as output by `findBestHill`. If no data points for drug B in isolation are included, this component is `NULL`.

Author(s)

Nathaniel R. Twarog

See Also

braidrm, findBestBRAID, findBestHill, hillConcCorrect

Examples

```
data(es8olatmz)
# Note that 'es8olatmz' contains all necessary column names, including
# 'conc1','conc2','act','compound1','compound2', and, for concentration
# correction, 'well'
## Not run: brdAnalysis <- runBRAIDanalysis(es8olatmz,defaults=c(0,-2.7),corrconc=TRUE)
brdAnalysis <- runBRAIDanalysis(es8olatmz,defaults=c(0,-4))
summary(brdAnalysis$braidFit)
```

Index