Package 'braidrm'

July 22, 2025

Title Fitting Combined Action with the BRAID Response Surface Model

Version 1.0.3

Description Contains functions for evaluating, analyzing, and fitting combined action dose response surfaces with the Bivariate Response to Additive Interacting Doses (BRAID) model of combined action, along with tools for implementing other combination analysis methods, including Bliss independence, combination index, and additional response surface methods.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.1

Config/Needs/website rmarkdown

Depends R (>= 3.10)

LazyData true

Suggests braidReports (>= 1.0.0), knitr, rmarkdown

VignetteBuilder knitr

Imports basicdrm, stats

NeedsCompilation no

Author Anang Shelat [aut],
Nathaniel R. Twarog [aut, cre]

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

Repository CRAN

Date/Publication 2024-09-26 21:00:41 UTC

Contents

`additiveExample` *Example Additive Surface*

Description

A synthetically generated response surface using a additive parameter vector. The surface was generated with IDMA of 1, IDMB of 1, na of 3, nb of 3, kappa of 0, E0 of 0, EfA of 1, EfB of 1, and Ef of 1. Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
additiveExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

antagonisticExample *Example Antagonistic Surface*

Description

A synthetically generated response surface using a antagonistic parameter vector. The surface was generated with IDMA of 1, IDMB of 1, na of 3, nb of 3, kappa of -1, E0 of 0, EfA of 1, EfB of 1, and Ef of 1. Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
antagonisticExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

braidrm *BRAID Response Surface Fitting*

Description

Fits a BRAID response surface model to data.

Usage

```
braidrm(
  formula,
  data,
  model = "kappa2",
  links = NULL,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE
```

```
)

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

## S3 method for class 'summary.braidrm'
print(x, ...)

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

## S3 method for class 'formula'
braidrm(
  formula,
  data,
  model = "kappa2",
  links = NULL,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE
)

## Default S3 method:
braidrm(
  formula,
  data,
  model = "kappa2",
  links = NULL,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE
)
```

Arguments

formula	Either an object of class formula such as would be provided to a modeling function like stats::lm(), or a width-2 numeric array vector of concentration pairs (including 0 or Inf). A formula should specify a single output as a function of two inputs, eg. activity ~ conc1 + conc2.
data	If forumula is a symbolic formula, a data frame containing the specified values.

	If `formula` is a numeric array of concentrations, a numeric vector of response values, the same length as the number of rows of `formula`.
`model`, `links`	Parameters `model` and `links` are used to specify which variant of the BRAID model is fit to data. Model may be one of the following character strings: "kappa1", "kappa2", or "kappa3" (see Details), or a subset of the numbers 1 through 9 specifying which of the nine BRAID response surface parameters is allowed to vary when fitting. `links` allows the user to further specify constraints on the three BRAID maximal effect parameters (see Details for more). If `model` is one of the supported character strings, the parameter `links` will be ignored.
`weights`	A vector of weights (between 0 and 1) the same length as the data which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If `NULL` (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in `data` if `formula` is a symbolic formula
`start`	A BRAID parameter vector specifying the first guess where the non-linear optimization should begin. May be a length 7, 8, or 9 vector, though a full length vector is always preferable. If `NULL` (the default), it will be estimated from the data.
`direction`	Determines the possible directionality of the BRAID model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effects are all *greater* than or equal to the minimal effect. If less than 0, the fitting will require that all maximal effect is *less* than or equal to the minimal effect.
`lower`	A numeric vector of lower bounds on the fitted parameter values. May be the same length as the number of fitted parameters, or a full, length-9 vector. Missing or unspecified lower bounds may be included as `NA` or `Inf`; if unspecified, lower bounds on the first five parameters (IDMA, IDMB, na, nb, and kappa) will be automatically estimated from the data. Bounds on the minimal and maximal effect parameters however (E0, EfA, EfB, and Ef) will be assumed to be infinite unless specified. A value of `NULL`, the default, will be treated as all lower parameter bounds being unspecified.
`upper`	A numeric vector of upper bounds on the fitted parameter values. Used in the same way as `lower`.
`prior`	A character string specifying the desired Bayesian prior term for kappa, or an object of class `kappaPrior` genererated by the function `kappaPrior()`. Allowed strings are "mild", "moderate" (the default), "high", or "none". If a string is given, the kappa prior object will be estimated from the data using an initial ten-parameter fit to approximate measurement noise.
`getCIs`	Should bootstrapped confidence intervals be estimated and added to the BRAID fit object. Default value is `TRUE`.
`object`	An object of class `braidrm` to be summarized
`...`	Further arguments passed to or from other methods.
`x`	An object of class `braidrm` or `summary.braidrm` to be printed

Details

One of the hairiest and most confusing aspects of fitting a combined response surface model is handling the relationships between maximal effects. Unlike a simple dose response model such as those fit in `basicdrm` in which all parameters can be treated as fairly independent, response surface models are often considered with constraints that cannot be expressed as simply one parameter being fixed a particular value. Many response surface models assume that the two drugs in combination (and the overall combination) share a single common maximal effect; others might assume that the maximal effects of the two drugs should differ but that the overall maximal effect must be equal to one of these; still others may wish to fit with no constraints on maximal effect at all beyond guaranteeing that they lie above a fixed minimal effect. All these approaches are valid, and creating a functional interface to support them all is a challenge. The parameters given here represent our best effort to balance ease-of-use with flexibility.

The primary interface for model selection and customization is the paired paramters `model` and `links`. For the first six parameters of the BRAID surface, `model` is the only relevant control, and operates much as it would in any fitting function. If a given parameter (say IDMB) is included in `model` (as index 2) then it will vary freely within the provided bounds to best fit the data. If it is not, the value will be fixed at the value given in `start` (or if `start` is `NULL`, estimated from the data), and will remain fixed at that value in the best fit surface.

Parameters EfA, EfB, and Ef (the maximal effect parameters) require slightly more care. Relationships between these values is represented by the `links` parameter, which can take on one of the following five values:

- "AB": Indicating that the overall Ef parameter is the driver, both EfA and EfB are constrained to be equal to Ef, whatever its value. Can be used when indices 7 and 8 (EfA and EfB) are absent from `model`, but index 9 (Ef) is present
- "F": Indicating instead that the individual parameters EfA and EfB are the drivers, and EfA is constrained to be equal to larger magnitude of the two. Can be used when indices 7 and 8 are present `model`, but index 9 is absent
- "A": Specifies that the overall maximal effect is equal to that of the first trug (and consequentially that the effect of drug A must be of greater or equal magnitude to that of drug B). Can be used when index 7 is absent in `model` and index 9 is present; index 8 may be present or absent
- "B": Specifies that the overall maximal effect is equal to that of the second drug. Can be used when index 8 is absent in `model` and index 9 is present; index 7 may be present or absent
- "" (the empty string): Indicates no equality between maximal effects. Parameters that are present in `model` vary freely in fitting, those that are absent are fixed at constant values.

For example, if the maximal effects *should* be fit, but should be constrained to be all equal, then it is the parameter Ef that varies freely in the fitting; the fact that EfA and EfB are always equal to this value is represented by setting the `links` parameter to "AB". Contrast this with the (admittedly much less common) scenario in which the `links` parameter is set to the empty string "", representing no link between maximal effects. In this case the parameter Ef will indeed vary freely in the fitting, but EfA and EfB will instead always be held at the constant initial values in the starting parameter vector. On the other hand, if we were to assume that the two individual maximum effect parameters can vary independently, but that the overall maximal effect should be equal to the larger of the two, then indices 7 and 8 (representing EfA and EfB) would be included in `model`; index 9 (representing Ef) would be excluded, and the `links` parameter would be set to "F" indicating that Ef is tied to the larger of the two.

Note also that the default value for `links` is *not* the empty string, but instead `NULL`. By default the value of `links` will be guessed from the model vector, based on the scenarios that we have encountered most often. If `model` inclues Ef (index 9) but not EfA or EfB (indices 7 and 8), `links` is assumed to be "F"; if EfA and EfB are present but not Ef, `links` is set to "AB". In the vast majority of cases, you will not need to specify this parameter yourself. This is especially true when `model` is specified with one of the model strings, in which any provided value for `links` will be discarded and replaced witht the following preset models:

- kappa1: Model vector includes (1, 2, 3, 4, 5, 6, 9) and `links` is set to "AB"
- kappa2: Model vector includes (1, 2, 3, 4, 5, 6, 7, 8) and `links` is set to "F"
- kappa3: Model vector includes (1, 2, 3, 4, 5, 6, 7, 8, 9) and `links` is set to the empty string

Value

An object of class `braidrm` containing the following elements:

- `concs`: A width-two array of the concentration pairs fit by the model
- `act`: A vector of responses fit by the model
- `weights`: The vector of weights (the same length as `act`) specifying the relative weight of each measurement
- `coefficients`: A full length-9 named BRAID parameter vector representing the best fit BRAID surface for the data
- `fitted.values`: A vector of responses (the same length of `act`) given by the best fit response surface as a function of the concentrations in `concs`
- `residuals`: The fitting errors of the best fit model, equal to the `fitted.values` subtracted from `act` `scenario`: A character string specifying one of 32 distinct fitting scenarios determined from the parameters `model`, `links`, and `start`. Used in bootstrapping confidence intervals
- `model`: The model vector (a subset of values between 1 and 9) specifying which BRAID parameters were varying freely in the fit
- `start`: The length-9 starting BRAID parameter vector used in non-linear optimization
- `direction`: Like the input parameter, a value of -1, 0, or 1 specifying the constraint on the directionality of the fitted surface
- `pbounds`: A 2-by-k array of values specifying the lower and upper bounds of all varying parameters (where k is the number of free parameters).
- `kweight`: A numeric value summarizing the relative Bayesian influence of the BRAID parameter kappa on optimized objective function.

Fit objects with bootstrapped confidence intervals include several additional elements derived from that; see `calcBraidBootstrap()` for details.

Examples

```
bfit1 <- braidrm(measure ~ concA + concB, additiveExample)
summary(bfit1)

bfit2 <- braidrm(measure ~ concA + concB, synergisticExample,
```

```
                  model = c(1,2,3,4,5,6,9),
                  lower = c(NA,NA,NA,NA,NA,0,0),
                  prior = "none",
                  getCIs = FALSE)
summary(bfit2)
```

`calcBraidBootstrap` *BRAID Parameter Confidence Intervals*

Description

Uses residuals-based bootstrapping to estimate confidence intervals on a BRAID fit's response surface parameters

Usage

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

Arguments

`bfit`	A BRAID fit object of class `braidrm`. If this object already has bootstrapped coefficients, a warning will be given, and they will be overwritten
`ciLevs`	The lower and upper quantiles at which the confidence intervals should be estimated. Default is 0.025 and 0.975, producing 95% confidence intervals
`numBoot`	The number of bootstrapped coefficient values to estimate. Defaults to a number large enough that at least 10 measurement should lie outside the estimated interval; but held to a minimum value of 100 and a maximum value of 1000.

Value

An object of class `braidrm` but with three additional elements:

- `ciLevs`: The two quantiles at which the confidence intervals are set
- `ciCoefs`: An array of bootstrapped coefficients. The number of rows is the number of *successful* bootstrapped fits; there is one column for each the nine BRAID parameters
- `ciMat`: An array of confidence intervals on the fitted parameters. The rows correspond to the *free* parameters included in the fit, and are named for those parameters; the firt column contains the lower bound of the confidence intervals, the second column the upper bound.

Examples

```
bfit <- braidrm(measure ~ concA + concB, synergisticExample, getCIs = FALSE)
summary(bfit)

bfit_ci <- calcBraidBootstrap(bfit)
summary(bfit_ci)
```

`calcBraidConfInt` *Generic BRAID confidence intervals*

Description

Generates confidence intervals on derived BRAID response surface values

Usage

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

Arguments

`bfit`	A BRAID fit object of class `braidrm` which contains a full set of bootstrapped response surface coefficients
`parfunc`	A function that takes a full-length BRAID parameter vector as an input and gives a single numeric value or numeric vector as an output. If the function produces a vector, it must produce the same length vector for all inputs
`civals`	If given, the lower and upper quantile values at which the confidence intervals are set. Defaults to the `ciLevs` parameter of the bootstrapped BRAID fit

Details

In come cases, it is desirable to estimate a confidence interval on a value derived from or dependent on a BRAID surface model that is not a parameter of the model itself. For example, one might want a confidence interval on a given index of achievable efficacy value, or the predicted effect at a certain set of dose pairs. This function replicates confidence interval calculations on any such derive values

Value

An n-by-3 array, where n is the length of the output produced by `parfunc`. The first column is the lower bound of the confidence interval; the second column is the derived value for the best fit coefficients; and the third column is the upper bound of the confidence intervals. Note that is possible for the lower bound of the confidence interval to lie above the central value, or for the upper bound to lie below it; though this is only likely to occur in the case of a poorly determined fit.

Examples

```
bfit <- braidrm(measure ~ concA + concB, synergisticExample, getCIs=TRUE)

calcBraidConfInt(bfit, function(p) evalBraidModel(10, 10, p))
calcBraidConfInt(bfit, function(p) estimateIAE(p, c(0.5, 0.9), c(10, 10)))
```

coactiveExample *Example Coactive Surface*

Description

A synthetically generated response surface using a flipped "coactive" parameter vector. The surface was generated with IDMA of 0.5, IDMB of 0.5, na of 3, nb of 3, kappa of 0, E0 of 0, EfA of 0, EfB of 0, and Ef of 1; the surface was flipped along both drug axes (so `flip` was set to "both"). Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
coactiveExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

deviationSurface *Non-interacting Reference Surfaces*

Description

Estimate the best fitting non-interacting reference surface according to multiple methods, including Loewe, Bliss, and HSA

Usage

```
deviationSurface(concs, act, method = "Bliss", ...)

referenceSurface(concs, act, method = "Bliss", ...)

blissDeviation(concs, act, range, clip = "none")

blissReference(concs, act, range, clip = "none")

hsaDeviation(concs, act, increasing)
```

```
hsaReference(concs, act, increasing)

loeweDeviation(concs, act, weights = NULL, limits = c(NA, NA))

loeweReference(concs, act, weights = NULL, limits = c(NA, NA))

zipDeviation(concs, act, range, weights = NULL, limits = c(NA, NA))

zipReference(concs, act, range, weights = NULL, limits = c(NA, NA))

zipSmoothed(concs, act, range, weights = NULL, limits = c(NA, NA))
```

Arguments

concs	A width-two array of concentrations representing all measured dose pairs
act	A vector of measured activity or effect values
method	A string specifying which model of non-interaction should be used; possible values are "Bliss" (the default), "HSA", "Loewe", and "ZIP"
...	Additional parameters to be passed to the method-specific deviation or reference surface functions
range	For Bliss calculations, the range of effects assumed by Bliss independence; a two-element numeric vector containing the minimal effect and the maximal effect, in that order. For ZIP calculations, the initial estimate of the minimal and maximal effects used in fitting the individual dose response curves.
clip	Clipping method for Bliss reference and deviation calculations. Possible values are "pre", "post", and "none". See details for specifics.
increasing	For HSA calculations, is the effect increasing (TRUE) meaning the "highest" single agent activity is numerically greater; or decreasing (FALSE), meaning the "highest" single agent activity is numerically lower. The latter may be appropriate when the modeled response is target growth or survival.
weights	A vector of weights (between 0 and 1) the same length as the data which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If NULL (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in data if formula is a symbolic formula
limits	For Loewe and ZIP calculations, the fixed values of the minimal and maximal effects of the drugs and the combination. By default, both values are set to NA; any value set to NA will fit from the data.

Details

This collection of functions can be used to implement a family of combination analysis methods known as "deviation" methods. The details of the methods differ, but the core strategy is common to all of them: estimate what a given response measurement *would be* based on individual behaviors and some model of non-interaction, and use the deviation of the measured response from that expected response as a measure of the degree of synergy or antagonism.

Bliss independence is the most widely used of these, and can be described as the assumption that any response represents a fraction of the target population being unaffected, and that a combined response corresponds to the product of these fractions for both drugs (Bliss, 1939). It is extremely simple to calculate, and relies on the intuitive model of probabilisticalyly independent events. Because it treats responses as a scaled representation of probabilities, it requires that all values be expressed relative to two limiting values: the response seen when **all** targets are unaffected (the minimal effect) and the response seen when **none** of the targets remain unaffected (the maximal effect). For a Bliss independent surface to be estimated, these two values must be provided, using the parameter `range`. Further, because values outside of this range would represent proportions above 1 or below 0, most Bliss calculations involve some adjustment of the data to ensure they always lie within the specified range. The Bliss functions support two ways of doing this to generate a reference surface: "pre" will clip all values to the range immediately; "post" will clip all calculated responses to the given range after they have been combined. A third option, "none", performs no clipping at all, and allows for proportions outside of the 0 to 1 range.

The highest-single-agent, or HSA model, is even simpler than Bliss. The effect of a combined pair of doses is simply the "greater" of the individual effects produced by the two drugs at those levels. The word "greater" is placed in quotes here as taking the larger response value is only appropriate when a numerically larger measurement corresponds to greater activity. For survival or growth inhibition studies, the reverse may be true; the parameter `increasing` allows the user to specify this directionality.

Loewe additivity is one of the oldest models of non-interaction, and the inspiration for BRAID additivity (Loewe and Muischnek, 1926). According to Loewe additivity, the combined response to a pair of doses the the effect such that the two individual doses represent complementary fractions of the individual doses of both drugs that produce the same effect in isolation. It is considered the gold standard of non-interaction by many researchers, but = has several significant pragmatic disadvantages. It requires that the full dose response behavior of both drugs is known, and that they produce an identical range of effects. The `loeweReference` and `loeweDeviation` functions perform basic dose response fitting with the additional constraint of matching minimal and maximal effects, either fixed by the `limits` parameter or estimated directly from the data.

The zero-interaction potency, or ZIP, model is a variant of Bliss independence that uses smoothing to give more robust values (Wooten *et al.* 2015). The reference surface is calculated by fitting the dose response of the individual drugs and then combining them using Bliss independence; the method then adds additional robustness and smooths the measured surface itself by fitting each constant-dose set of data points as its own dose response curve, constrained to match the other drug's effect when the first drug is zero. Fitting these partial dose-response curves in either direction produces a smoothed version of the original measurements (which can be accessed directly using the function `zipSmoothed`), from which the reference surface is subtracted to get the deviation (or "delta") surface.

Value

For the deviation functions (`deviationSurface`, `blissDeviation`, `hsaDeviation`, `loeweDeviation`, and `zipDeviation`), a vector of values the same length as `act` and/or `concs` representing the deviation of the measurement from the specified reference surface. For the reference functions (`referenceSurface`, `blissReference`, `hsaReference`, `loeweReference`, and `zipReferences`), a vector of values the same length as `act` and/or `concs` containing the appropriate non-interacting reference surface itself. For `zipSmoothed`, the smoothed measurement surface given by ZIP's dose-response-based smoothing method (see Details).

References

Bliss, Chester I. 1939. "The Toxicity of Poisons Applied Jointly 1." *Annals of Applied Biology* **26** (3): 585–615.

Loewe, S, and H Muischnek. 1926. "Uber Kombinationswirkungen." Naunyn. Schmiedebergs. Arch. Pharmacol. 114: 313–26.

Yadav, Bhagwan, Krister Wennerberg, Tero Aittokallio, and Jing Tang. 2015. "Searching for Drug Synergy in Complex Dose–Response Landscapes Using an Interaction Potency Model." *Computational and Structural Biotechnology Journal* **13**: 504–13.

Examples

```
surface <- additiveExample
concs1 <- cbind(surface$concA,surface$concB)
act1 <- surface$measure

sum(deviationSurface(concs1,act1,"Bliss",range=c(0,1)))
sum(deviationSurface(concs1,act1,"Loewe"))
surface$hsa <- hsaReference(concs1,act1,increasing=TRUE)

surface <- synergisticExample
concs2 <- cbind(surface$concA,surface$concB)
act2 <- surface$measure

sum(deviationSurface(concs2,act2,"ZIP",range=c(0,1)))
sum(deviationSurface(concs2,act2,"Loewe"))
surface$smooth <- zipSmoothed(concs2,act2,range=c(0,1))
```

---

estimateCombinationIndices

Combination Index

Description

Estimates the combination index using the median effet method of Chou and Talalay (1984) or the more robust method of non-linear optimization.

Usage

```
estimateCombinationIndices(
  concs,
  act,
  level,
  weights = NULL,
  limits = c(NA, NA)
)

estimateCombinationIndex(dr1, dr2, drc, ratio, level, limits = c(NA, NA))
```

```
estimateChouIndices(concs, act, level, range, excess = "clip")

estimateChouIndex(dr1, dr2, drc, ratio, level, range, excess = "clip")
```

Arguments

concs	A width-two array of concentrations representing all measured dose pairs
act	A vector of measured activity or effect values
level	A numeric vector of one or more effect levels at which to estimate the combination index
weights	A vector of weights (between 0 and 1) the same length as the data which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If NULL (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in data if formula is a symbolic formula
limits	The fixed values of the minimal and maximal effects of the drugs and the combination. By default, both values are set to NA; any value set to NA will fit from the data.
dr1	A data frame with two columns, conc and act reflecting the dose response behavior of the first drug alone
dr2	A data frame with two columns, conc and act reflecting the dose response behavior of the second drug alone
drc	A data frame with two columns, conc and act reflecting the dose response behavior of a constant ratio combination; conc should be the combined concentrations of the two drugs
ratio	The ratio of the two drugs in the constant ratio combination (dose A to dose B)
range	The range of effects assumed by the median effect model; a two-element numeric vector containing the minimal effect and the maximal effect, in that order.
excess	For estimateChouIndices and estimateChouIndex, what should be done with values outside the expected range. If "clip" (the default), values will be clipped to 0.1% and 99.9% of the expected range; if "drop", values outside the range will be dropped

Details

The combination index is a peculiar method, as it does not produce values corresponding to each measurement (as the deviation methods do), nor does it produce a value shared by the entire surface (as response surface methods do). It instead, produces a value associated with a particular effect level and a particular *dose ratio*. This makes implementing the method consistently for a wide range of possible data sources quite tricky; nevertheless, we have attempted to do so here. In brief, the combination index method involves fitting the dose response of both individual drugs and and a constant ratio combination of the drugs (treated as a virtual third drug). It the compares the potency of the constant-ratio combination (estimated a particular effect level) with the expected potency according to Loewe additivity, and returns the degree to which the combination is *more* potent (with a combination index less than 1) or *less* potent (with a combination greater than one)

than expected by additivity. This method is also in turns known as the sum of fractional inhibitory coefficients (FICs), observed-over-expected, or originally as Berenbaum's interaction index.

Because the method was originally built for three distinct sets of dose response measurements, we have included the `estimateCombinationIndex` and `estimateChouIndex` functions which operate on three separate data frames. However, in most cases, it will be easier to use the `estimateCombinationIndices` and `estimateChouIndices` functions, which operate on an array of concentrations and a vector of responses, just like the numerous other functions in this package, and generate a set of combination index values with level and ratio information included.

The only difference between the `estimateCombination*` and `estimateChou*` functions is the way in which they perform dose response fitting. The `estimateCombination*` functions use non-linear least squares optimization (based on the package `basicdrm`) to estimate dose-response parameters. The `estimateChou*` functions use the median-effect method described by Chou and Talalay in their 1984 paper, which linearized all measurements and performs linear regression. We *do not* recommend using these methods, as they are much less reliable than the non-linear optimization approach and extremely susceptible to noise at extreme values.

Value

For `estimateCombinationIndices` and `estimateChouIndices`, a data frame with the following columns:

- `ratio`: The ratio of doses (dose A to dose B) along which the index was estimated
- `level`: The effect level at which the index was estimated
- `ci`: The estimated combination index at that dose ratio and effect level

Combination index estimates will be included for all provided effect levels and all dose ratios present. For `estimateCombinationIndex` and `estimateChouIndex`, a vector of estimated combinatoin indices the same length as `level`.

References

Berenbaum, MC. 1978. "A Method for Testing for Synergy with Any Number of Agents." *Journal of Infectious Diseases* **137** (2): 122–30.

Berenbaum, MC. 1989. "What Is Synergy." *Pharmacol Rev* **41**: 93–141.

Chou, Ting-Chao, and Paul Talalay. 1984. "Quantitative Analysis of Dose-Effect Relationships: The Combined Effects of Multiple Drugs or Enzyme Inhibitors." *Advances in Enzyme Regulation* **22**: 27–55.

Examples

```
surface <- synergisticExample
concs1 <- cbind(surface$concA, surface$concB)
act1 <- surface$measure

estimateCombinationIndices(concs1,act1,c(0.5))

dr1 <- surface[surface$concB==0, c("concA","measure")]
names(dr1) <- c("conc","act")
dr2 <- surface[surface$concA==0, c("concB","measure")]
```

```
names(dr2) <- c("conc","act")
drc <- surface[surface$concA==surface$concB,]
drc$conc <- drc$concA+drc$concB
drc <- drc[,c("conc","measure")]
names(drc) <- c("conc","act")

estimateChouIndex(dr1,dr2,drc,ratio=1,
                  level=c(0.5,0.9,0.99),
                  range=c(0,1))
```

`estimateIAE` *BRAID Response Surface Combined Potency*

Description

Estimates the BRAID index of achievable efficacy (or IAE) for a given response surface

Usage

```
estimateIAE(bpar, levels, limits, lowerLimits = c(0, 0))

## S3 method for class 'braidrm'
estimateIAE(bpar, levels, limits, lowerLimits = c(0, 0))

## S3 method for class 'braidrmflip'
estimateIAE(bpar, levels, limits, lowerLimits = c(0, 0))

## Default S3 method:
estimateIAE(bpar, levels, limits, lowerLimits = c(0, 0))

estimateFlippedIAE(bpar, flip, levels, limits, lowerLimits = c(0, 0))
```

Arguments

`bpar`	The response surface to be evaluated. Can be a numeric vector (which will be treated as a standard BRAID parameter vector), a BRAID fit object of class `braidrm`, or a flipped BRAID fit object of class `braidrmflip`
`levels`	The effect level or levels at which the index is to be estimated
`limits`	The upper concentration limits beneath which the IAE is to be estimated. Could be known toxicity thresholds, limits on pharmacokinetic availability, or simply a convenient and consistent reference concentration
`lowerLimits`	By default, the IAE is calculated by comparing the area of dose space below which a given effect level is not reached with the total achievable dose space specified by `limits`. However, in some cases, it is not desirable to allow the sub-threshold area to become infinitesimally small. If lowerLimits is included, doses that lie below both lower limits will always be included in the sub-threshold area, placing an effective upper bound on possible IAE values

flip — String specifying the direction or directions of the surface's flip. Must be one of "A", "B", or "both".

Details

The index of achievable efficacy is an aggregate measure of combined potency, and a useful first pass for quantifying the efficacy of a given response surface. Formally, it is equal to the area of achievable dose pairs divided by the area of achievable doses below which a desired effect level is not reached (then passed through a square root to give a more dimensionless result). If the surface is more potent, the area of sub-threshold achievable doses is smaller, and the IAE is larger. If the surface is less efficacious, the doses at which a desired effect is reach will be larger, the sub-threshold area will increase, and the IAE will decrease. By default, the IAE can range from 1 to infinity, but upper bounds can be placed by setting the values in `lowerLimits` to concentrations above 0. For convenience, the function takes BRAID parameter vectors, `braidrm` fit objects, and `braidrmflip` flipped BRAID fit objects. However, flipped BRAID response surface parameters cannot be passed to the function as is, so the function `estimateFlippedIAE` is also included specifically for flipped parameter vectors.

Value

A numeric vector, the same length as `levels` with the estimated index of achievable efficacy (IAE) values for each of those levels.

Examples

```
estimateIAE(c(1,1,3,3,0,0,100),c(50,90),c(5,5))

bfit <- braidrm(measure ~ concA + concB, synergisticExample, getCIs = FALSE)
estimateIAE(bfit, c(0,0.25,0.5,0.75,1), c(10,1), lowerLimits=c(0.01,0.01))
```

---

evalBraidModel *Evaluate the BRAID response surface model*

Description

Evaluate the BRAID response surface model

Usage

```
evalBraidModel(DA, DB, bpar, calcderivs = FALSE)
```

Arguments

DA — A vector of concentrations of drug A in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DB`.

DB — A vector of concentrations of drug B in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DA`.

bpar	A BRAID response surface parameter vector (see Details)
calcderivs	Primarily used by fitting functions for non-linear optimization. If FALSE (the default), the function returns a vector of response values; if TRUE, it returns a list including the partial derivatives of the BRAID parameters.

Details

The BRAID response model is, in total, described by nine response surface parameters. A BRAID parameter vector should uniquely determine all these values. They are

- IDMA: The dose of median effect of drug A, also called the EC50
- IDMB: The dose of median effect of drug B
- na: The Hill slope, or sigmoidicity, of drug A
- nb: The Hill slope of drug B
- kappa: The BRAID interaction parameter, indicating additivity (kappa = 0), antagonism (2 < kappa < 0), or synergy (kappa > 0)
- E0: The minimal effect, the effect observed when neither drug is present
- EfA: The maximal effect of drug A, the effect theoretically observed when drug B is absent and drug A is present at infinite concentration
- EfB: The maximal effect of drug B,
- Ef: The maximal effect of the combination, theoretically observed when both drugs are present at infinite concentration. It may be (but often is not) further from E0 than either EfA or EfB.

In many cases, however, it is easier to specify only some of the final three parameters. braidrm functions therefore support BRAID parameter vectors of length 7 (in which the sixth and seventh values are assumed to be E0 and Ef, and EfA and EfB are assumed to be equal to Ef), length 8 (in which the seventh and eighth values are EfA and EfB, and Ef is assumed to be equal to whichever of these two values is further from E0), or the full length 9 parameter vector.

Value

If calcderivs is FALSE, a numeric vector the same length as DA and/or DB with the predicted BRAID response surface values. If calcderivs is TRUE, a list with two elements: value, containing the response surface values, and derivatives, a matrix with as many rows as value has elements, and nine columns containing the partial derivatives of the response surface with respect to the nine BRAID response surface parameters

Examples

```
concentrations <- c(0, 2^(-3:3))
surface <- data.frame(
    concA = rep(concentrations,each=length(concentrations)),
    concB = rep(concentrations,times=length(concentrations))
)

surface$additive <- evalBraidModel(
    surface$concA,
    surface$concB,
```

```
    c(1, 1, 3, 3, 0, 0, 100, 100, 100)
)

surface$synergy <- evalBraidModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, 2, 0, 80, 90)
)

surface$antagonism <- evalBraidModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, -1, 0, 100)
)

head(surface)
```

`evalFlippedBraidModel` *Evaluate Flipped BRAID Surfaces*

Description

Evaluate Flipped BRAID Surfaces

Usage

```
evalFlippedBraidModel(DA, DB, bpar, flip)
```

Arguments

`DA`	A vector of concentrations of drug A in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DB`.
`DB`	A vector of concentrations of drug B in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DA`.
`bpar`	Flipped-BRAID parameter of the flipped response surface. See details for more information on specifying atypical surfaces
`flip`	String specifying the direction or directions of the surface's flip. Must be one of "A", "B", or "both".

Details

While the BRAID model generates a fairly versatile range of combined behaviors, the traditional model is still strictly constrained in certain respects. Response surfaces must exhibit a change in response to both drugs and this change must be in the same direction. Furthermore, the model is only suited to surfaces in which the change resulting from single drugs is larger than the additional effect of the combination. However, modifying the model equation by inverting one or both of the slope parameters can produce a set of qualitatively distinct "flipped" surfaces, allowing the model to produce a much wider range of behaviors. For example, flipping the model along the axis

representing drug A can produce a surface in which drug B has no effect in isolation, but attenuates or eliminates the effect of drug A, a pattern we call a "protective" surface. See `fitBraidFlipped()` for the possible range of surfaces.

An important note: a flipped BRAID surface, like a traditional BRAID surface, is represented by a parameter vector of up to 9 values. While these functions will attempt to fill a 7- or 8- value parameter factor to a full 9-element vector, it is strongly recommended that you specify the response surface with the full 9-element vector, as the precise ordering with which implicit values are arranged can be extremely confusing. Note also that flipped parameter vectors, regardless of the underlying mathematical representation, should always be specified in the same order as in a traditional vector. So in a full 9-element vector:

- Parameter 6 (E0) should always specify the expected effect when both drugs are absent
- Parameter 7 (EfA) should always specify the expected effect at high concentrations of drug A when drug B is absent
- Parameter 8 (EfB) should always specify the expected effect at high concentrations of drug B when drug A is absent
- Parameter 9 (Ef) should always specify the expected effect when both drugs are present at high concentrations

Value

A numeric vector the same length as `DA` and/or `DB` with the predicted flipped BRAID response surface values.

Examples

```
concentrations <- c(0, 2^(-3:3))
surface <- data.frame(
    concA = rep(concentrations,each=length(concentrations)),
    concB = rep(concentrations,times=length(concentrations))
)

surface$protective <- evalFlippedBraidModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, 0, 0, 100, 0, 10),
    flip="A"
)
surface$coactive <- evalFlippedBraidModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, 0, 0, 0, 0, 100),
    flip="both"
)

head(surface)
```

`evalMusycModel` *Evaluate MuSyC Response Surfaces*

Description

Evaluates the Mulitdimensional Synergy of Combinations (MuSyC) model of combined action for the given values and parameters (Wooten *et al.* 2021).

Usage

```
evalMusycModel(DA, DB, mupar, calcderivs = FALSE)
```

Arguments

`DA`	A vector of concentrations of drug A in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DB`.
`DB`	A vector of concentrations of drug B in a combination (values 0 and `Inf` are permitted). Must be length 1 or the same length as `DA`.
`mupar`	A MuSyC response surface parameter vector; may be length 8, 10, or 12 (see details for specifics of MuSyC parameters)
`calcderivs`	Primarily used by fitting functions for non-linear optimization. If `FALSE` (the default), the function returns a vector of response values; if `TRUE`, it returns a list including the partial derivatives of the BRAID parameters.

Details

The multi-dimensional synergy of combinatoins, or MySyC, model is a parametric response surface model introduced by Wooten et al. in 2021. The method models the effect of combination by simulating occupancy in four compartments in which compounds are affected or unaffected by either drug. The full MuSyC model can be specified by a total of twelve parameters:

- `IDMA`: dose of median effect of first drug
- `IDMB`: dose of median effect of second drug
- `na`: Hill slope of first drug
- `nb`: Hill slope of second drug
- `alpha12`: factor by which first drug potentiates the second
- `alpha21`: factor by which second drug potentiates the first
- `gamma12`: factor by which first drug increases second drug's Hill slope
- `gamma21`: factor by which second drug increases first drug's Hill slope
- `E0` - the observed effect when unaffected by either drug
- `EfA` - the observed effect when affected by drug 1 but not drug 2
- `EfB` - the observed effect when affected by drug 2 but not drug 1
- `Ef` - the observed effect when affected by both drugs

In practice, gamma12 and gamma21 are rarely used, so a ten-element parameter vector specifies the other 10 values and assumes that gamma12 and gamma21 are both equal to 1. In some cases it is even useful to specify a MuSyC surface with no interaction at all with an eight-element vector, in which case alpha12, alpha21, gamma12, and gamma21 are all set equal to 1.

Value

If calcderivs is FALSE, a numeric vector the same length as DA and/or DB with the predicted MuSyC response surface values. If calcderivs is TRUE, a list with two elements: value, containing the response surface values, and derivatives, a matrix with as many rows as value has elements, and all columns containing the partial derivatives of the response surface with respect to the fitted MuSyC response surface parameters

References

Wooten, David J, Christian T Meyer, Alexander LR Lubbock, Vito Quaranta, and Carlos F Lopez. 2021. "MuSyC Is a Consensus Framework That Unifies Multi-Drug Synergy Metrics for Combinatorial Drug Discovery." *Nature Communications* **12** (1): 4607.

Examples

```
efficacyPar <- c(
    1, 1, 3, 3,
                      # Omitted shape synergy parameters assume to be 1
    0, 100, 100, 125  # Elevated Ef indicates efficacy synergy
)
potencyPar <- c(
    1, 1, 3, 3,
    10, 15,           # alphas above 1 indicate potency synergy
    0, 100, 100, 100  # No efficacy synergy
)

concentrations <- c(0, 2^(-3:3))
surface <- data.frame(
    concA = rep(concentrations,each=length(concentrations)),
    concB = rep(concentrations,times=length(concentrations))
)
surface$efficacy <- evalMusycModel(surface$concA, surface$concB, efficacyPar)
surface$potency  <- evalMusycModel(surface$concA, surface$concB, potencyPar)

head(surface)
```

---

evalUrsaModel *Evaluate URSA response surface model*

Description

Numerically estimates the universal response surface approach (URSA) model for the given data and parameters (Greco, Park, and Rustum 1990).

Usage

```
evalUrsaModel(DA, DB, upar)
```

Arguments

DA	A vector of concentrations of drug A in a combination (values 0 and Inf are permitted). Must be length 1 or the same length as DB.
DB	A vector of concentrations of drug B in a combination (values 0 and Inf are permitted). Must be length 1 or the same length as DA.
upar	A length seven URSA response surface parameter vector (see Details)

Details

The URSA model is described by the following seven values

- IDMA: The dose of median effect of drug A, also called the EC50
- IDMB: The dose of median effect of drug B
- na: The Hill slope, or sigmoidicity, of drug A
- nb: The Hill slope of drug B
- alpha: The URSA interaction parameter, indicating additivity (alpha = 0), antagonism (alpha < 0), or synergy (alpha > 0)
- E0: The minimal effect, the effect observed when neither drug is present
- Ef: The maximal effect of the drugs, theoretically observed when either drug is present at infinite concentration

Value

A numeric vector the same length as DA and/or DB with the predicted URSA response surface values.

References

Greco, William R, Hyoung Sook Park, and Youcef M Rustum. 1990. "Application of a New Approach for the Quantitation of Drug Synergism to the Combination of Cis-Diamminedichloroplatinum and 1-b-d-Arabinofuranosylcytosine." *Cancer Research* **50** (17): 5318–27.

Examples

```
concentrations <- c(0, 2^(-3:3))
surface <- data.frame(
    concA = rep(concentrations,each=length(concentrations)),
    concB = rep(concentrations,times=length(concentrations))
)

surface$uadditive <- evalUrsaModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, 0, 0, 100)
)
```

```
surface$usynergy <- evalUrsaModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, 5, 0, 80)
)

surface$uantagonism <- evalUrsaModel(
    surface$concA,
    surface$concB,
    c(1, 1, 3, 3, -0.5, 0, 100)
)

head(surface)
```

`findBestBraid` *Select Best BRAID Response Fit*

Description

Picks the most parsimonious BRAID fit from a standard set of commonly used variants

Usage

```
findBestBraid(
  formula,
  data,
  defaults,
  extended = FALSE,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE,
  useBIC = TRUE
)

## S3 method for class 'formula'
findBestBraid(
  formula,
  data,
  defaults,
  extended = FALSE,
  weights = NULL,
  start = NULL,
  direction = 0,
```

```
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE,
  useBIC = TRUE
)

## Default S3 method:
findBestBraid(
  formula,
  data,
  defaults,
  extended = FALSE,
  weights = NULL,
  start = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL,
  prior = "moderate",
  getCIs = TRUE,
  useBIC = TRUE
)
```

Arguments

formula	Either an object of class `formula` such as would be provided to a modeling function like `stats::lm()`, or a width-2 numeric array vector of concentration pairs (including 0 or Inf). A formula should specify a single output as a function of two inputs, eg. `activity ~ conc1 + conc2`.
data	If `forumula` is a symbolic formula, a data frame containing the specified values. If `formula` is a numeric array of concentrations, a numeric vector of response values, the same length as the number of rows of `formula`.
defaults	Default minimal and maximal effect values used to fix effect parameters during model selection.
extended	Should models with an additional freely varying Ef parameter be included. If `FALSE` (the default), ten models in which the maximal effect parameter Ef is constrained to be equal to one or more of the two individual maximal effect parameters will be tested; if `TRUE`, an additional two models in which Ef varies freely will be included.
weights	A vector of weights (between 0 and 1) the same length as the data which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If `NULL` (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in `data` if `formula` is a symbolic formula
start	A BRAID parameter vector specifying the first guess where the non-linear optimization should begin. May be a length 7, 8, or 9 vector, though a full length vector is always preferable. If `NULL` (the default), it will be estimated from the data.

`direction` Determines the possible directionality of the BRAID model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effects are all *greater* than or equal to the minimal effect. If less than 0, the fitting will require that all maximal effect is *less* than or equal to the minimal effect.

`lower` A numeric vector of lower bounds on the fitted parameter values. May be the same length as the number of fitted parameters, or a full, length-9 vector. Missing or unspecified lower bounds may be included as `NA` or `Inf`; if unspecified, lower bounds on the first five parameters (IDMA, IDMB, na, nb, and kappa) will be automatically estimated from the data. Bounds on the minimal and maximal effect parameters however (E0, EfA, EfB, and Ef) will be assumed to be infinite unless specified. A value of `NULL`, the default, will be treated as all lower parameter bounds being unspecified.

`upper` A numeric vector of upper bounds on the fitted parameter values. Used in the same way as `lower`.

`prior` A character string specifying the desired Bayesian prior term for kappa, or an object of class `kappaPrior` genererated by the function `kappaPrior()`. Allowed strings are "mild", "moderate" (the default), "high", or "none". If a string is given, the kappa prior object will be estimated from the data using an initial ten-parameter fit to approximate measurement noise.

`getCIs` Should bootstrapped confidence intervals be estimated and added to the BRAID fit object. Default value is `TRUE`.

`useBIC` If `TRUE` (the default), the best (read: most parsimonious) model will be selected from all tested models using the Bayesian information criterion (Schwarz 1978). If `FALSE` the function will follow the convention of earlier versions of the `braidrm` package and use the Akaike information criterion (Akaike 1974).

Details

When fitting real experimental data, it is not uncommon for a measured surface to contain such incomplete or noisy data that many of the parameters are highly underdetermined. Unfortunately, in such cases, non-linear optimization can often resort to wildly implausible values to explain small variations in the data. To address this, this function runs multiple BRAID response fits, including some in which the minimal and maximal effect parameters are constrained to reasonable default values, to test if additional free parameters offer sufficiently improved fits to be included.

When the parameter `extended` is set to `FALSE`, the function runs ten BRAID scenarios: five in which the minimal effect parameter is allowed to vary freely, and five in which it is fixed at the first default value. The five tested models in each set represent five distinct configurations of the maximal effect parameters:

- Both maximal effects are fixed the same value (the second default)
- Maximal effect EfA (and when it is larger, Ef) varies freely, but effect EfB is fixed at the second default
- Maximal effect EfB (and when it is larger, Ef) varies freely, but effect EfA is fixed at the second default
- The maximal effect Ef varies freely, and both EfA and EfB are constrained to be equal to it

- The maximal effects EfA and EfB both vary freely, and Ef is constrained to be equal to the larger of the two

When `extended` is `TRUE`, two additional models (one with E0 fixed and one in which it varies freely) are included, in which all three maximal effect parameters are allowed to vary freely and independently.

Value

An object of class `braidrm`. It will contain all the fields of a standard `braidrm` object, and also an additional field, `allfits` containing a summary of the best fit model from each of the 10 or 12 candidate models tested.

References

Akaike, Hirotugu. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* **19** (6): 716–23.

Schwarz, Gideon. 1978. "Estimating the Dimension of a Model." *The Annals of Statistics*, 461–64.

Examples

```
bfit1 <- findBestBraid(measure ~ concA + concB, additiveExample,
                       defaults=c(0,1))
summary(bfit1)
length(bfit1$allfits)

bfit2 <- findBestBraid(measure ~ concA + concB, additiveExample,
                       defaults=c(0,2), extended=TRUE, getCIs = FALSE)
summary(bfit2)
length(bfit2$allfits)
```

`fitBraidFlipped` *Fit Flipped BRAID Surfaces*

Description

Functions to fit protective, oppoistional and coactive BRAID surfaces, and more specific flipped surfaces if necessary.

Usage

```
fitBraidFlipped(formula, data, flip, model, links = NULL, ...)

fitProtectiveBraid_A(formula, data, ...)

fitProtectiveBraid_B(formula, data, ...)

fitOppositionalBraid_A(formula, data, ...)
```

```
fitOppositionalBraid_B(formula, data, ...)

fitCoactiveBraid_pure(formula, data, ...)

fitCoactiveBraid_partial(formula, data, ...)
```

Arguments

formula	Either an object of class formula such as would be provided to a modeling function like stats::lm(), or a width-2 numeric array vector of concentration pairs (including 0 or Inf). A formula should specify a single output as a function of two inputs, eg. activity ~ conc1 + conc2.
data	If forumula is a symbolic formula, a data frame containing the specified values. If formula is a numeric array of concentrations, a numeric vector of response values, the same length as the number of rows of formula.
flip	String specifying the direction or directions of the surface's flip. Must be one of "A", "B", or "both".
model, links	Parameters model and links are used to specify which variant of the BRAID model is fit to data. Model may be one of the following character strings: "kappa1", "kappa2", or "kappa3" (see Details), or a subset of the numbers 1 through 9 specifying which of the nine BRAID response surface parameters is allowed to vary when fitting. links allows the user to further specify constraints on the three BRAID maximal effect parameters (see Details for more). If model is one of the supported character strings, the parameter links will be ignored.
...	Additional parameters to be passed to braidrm

Details

Though fitBraidFlipped offers the option of fitting any flipped BRAID surface model specified by flip, this is not recommended, as the interplay between flipping paramteers and parameter constraints becomes very confusing very quickly. In nearly all cases, it is preferable to use one of the pre-defined flipped fitting functions.

fitProtectiveBraid_A and fitProtectiveBraid_B fit "protective" surfaces in which one drug has no effect in isolation, but attenuates or eliminates the effect of the other. fitProtectiveBraid_A generates a surface in which drug A is active and is attenuated by drug B; fitProtectiveBraid_B generates the reverse.

fitOppositinalBraid_A and fitOppositionalBraid_B produce "oppositional" surfaces in which a second drug produces an effect that is in the opposite direction to the first drug, but which is then overwhelmed by the effect of the first drug at higher concentrations. fitProtectiveBraid_A generates a surface in which the maximal effect of drug A dominates a high concentrations, fitProtectiveBraid_B generates the reverse. Note that the A and B in the function names specify which compound's effect is dominant, not the direction of the underlying flip; in actuality the surfaces generated by fitProtectiveBraid_A are produced by flipping along the B axis.

fitCoactiveBraid_pure and fitCoactiveBraid_partial produce "coactive" surfaces, in which both drugs have no or minimal effect in isolation, but produce a pronounced effect when both are present. fitCoactiveBraid_pure generates surfaces in which both drugs have no effect at all in

isolation; `fitCoactiveBraid_partial` generates surfaces in which either drug may have a smaller partial effect in isolation.

Value

A fit object of class `braidrmflip`. This structure contains the exact same elements as an object of class `braidrm` (see `braidrm()` for details) along with one additional element: `flip`, a character value specifying the direction that the surface is flipped. The object's `coefficients` and `flip` fields can be used to evaluate and invert the best fit response surface using `evalFlippedBraidModel()` and `invertFlippedBraidModel()`.

Examples

```
fbfit1 <- fitProtectiveBraid_A(measure ~ concA + concB,
                               protectiveExample, getCIs=FALSE)
coef(fbfit1)

fbfit2 <- fitOppositionalBraid_A(measure ~ concA + concB,
                                 oppositionalExample, getCIs=FALSE)
coef(fbfit2)

fbfit3 <- fitCoactiveBraid_pure(measure ~ concA + concB,
                                coactiveExample, getCIs=FALSE)
coef(fbfit3)
```

---

fitMusycModel *MuSyC Response Surface Fitting*

Description

Fits the Multidimensional Synergy of Combinations (MuSyC) model of combined action to the given data (Wooten *et al.* 2021).

Usage

```
fitMusycModel(
  formula,
  data,
  variant = "standard",
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)

## S3 method for class 'formula'
fitMusycModel(
  formula,
```

```
  data,
  variant = "standard",
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)

## Default S3 method:
fitMusycModel(
  formula,
  data,
  variant = "standard",
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)
```

Arguments

formula	Either an object of class formula such as would be provided to a modeling function like stats::lm(), or a width-2 numeric array vector of concentration pairs (including 0 or Inf). A formula should specify a single output as a function of two inputs, eg. activity ~ conc1 + conc2.
data	If forumula is a symbolic formula, a data frame containing the specified values. If formula is a numeric array of concentrations, a numeric vector of response values, the same length as the number of rows of formula.
variant	String specifying which variant of the MuSyC model is to be fit to the data. If "standard" (the default), all MuSyC parameters except gamma12 and gamma21 will be fit (these will be fixed at 1). If "independent", the four individual dose-response parameters (IDMA, IDMB, na, and nb) and the four maximal effect parameters (E0, EfA, EfB and Ef) will be fit, while the four interaction parameters (alpha12, alpha21, gamma12, and gamma12) will all be fixed at 1. If "full", the full twelve-parameter MuSyC vector will be fit.
weights	An optional vector of weights the same length as act. If NULL (the default), will be set to 1 for all measurements
direction	Determines the possible directionality of the BRAID model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effects are all *greater* than or equal to the minimal effect. If less than 0, the fitting will require that all maximal effect is *less* than or equal to the minimal effect.
lower	An optional set of lower bounds on the fitted MuSyC response parameters. Any values set to NA will be filled with default calculated bounds. May be length 4 (will be treated as a set of lower bounds on the minimal and maximal effect parameters only), length 8 (will be treated as lower bounds on the four individual dose response parameters and the four minimal and maximla effect parameters),

the same length as the space of parameters being optimized (8 for "independent", 10 for "standard", or 12 for "full"), or length 12.

`upper` An optional set of lower bounds on the fitted MuSyC response parameters. Behaves the same as `lower`.

Value

An object of class `braidAltFit` with the following values:

- `concs`: The array of concentrations passed to the functions
- `act`: The vector of measurements associated with the given dose pairs
- `weights`: The vector of weights for the given measurements, set to 1 for all measurements by default
- `method`: Specifying the alternate surface model being used (in this case "MuSyC")
- `variant`: A string specifying which MuSyC variant was fit: "independent", "standard", or "full"
- `coefficients`: A parameter vector of the appropriate length for `variant` specifying the best fit response surface
- `fitted.values`: The predicted response surface value for the given dose pairs and best-fit response surface
- `residuals`: The difference between the predicted and measured values for the given dose pairs, always equal to "measured minus predicted"
- `direction`: The direction value passed to the function
- `pbounds'`: A 2-by-k array of bounds on the MuSyC parameters used in the parameter optimization

References

Wooten, David J, Christian T Meyer, Alexander LR Lubbock, Vito Quaranta, and Carlos F Lopez. 2021. "MuSyC Is a Consensus Framework That Unifies Multi-Drug Synergy Metrics for Combinatorial Drug Discovery." *Nature Communications* **12** (1): 4607.

Examples

```
mfit1 <- fitMusycModel(measure ~ concA + concB, synergisticExample)
coef(mfit1)

mfit2 <- fitMusycModel(measure ~ concA + concB, oppositionalExample,
                       variant = "independent")
coef(mfit2)
```

`fitUrsaModel` *URSA Response Surface Fitting*

Description

Fits the universal response surface approach (URSA) model to the given data (Greco, Park, and Rustum 1990)

Usage

```
fitUrsaModel(
  formula,
  data,
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)

## S3 method for class 'formula'
fitUrsaModel(
  formula,
  data,
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)

## Default S3 method:
fitUrsaModel(
  formula,
  data,
  weights = NULL,
  direction = 0,
  lower = NULL,
  upper = NULL
)
```

Arguments

`formula`	Either an object of class `formula` such as would be provided to a modeling function like `stats::lm()`, or a width-2 numeric array vector of concentration pairs (including 0 or Inf). A formula should specify a single output as a function of two inputs, eg. `activity ~ conc1 + conc2`.

`data`	If `forumula` is a symbolic formula, a data frame containing the specified values. If `formula` is a numeric array of concentrations, a numeric vector of response values, the same length as the number of rows of `formula`.
`weights`	A vector of weights (between 0 and 1) the same length as the data which determines the weight with which each measurement will impact the the sum of squared errors. Weights will be multiplied by errors *before* squaring. If `NULL` (the default) all weights will be set to 1. Can be a numeric vector, or the name of a column in `data` if `formula` is a symbolic formula
`direction`	Determines the possible directionality of the BRAID model. If 0 (the default) no additional constraints are placed on the parameters. If greater than 0, the fitting will require that the maximal effects are all *greater* than or equal to the minimal effect. If less than 0, the fitting will require that all maximal effect is *less* than or equal to the minimal effect.
`lower`	An optional set of lower bounds on the seven URSA response parameters. Any values set to NA will be filled with default calculated bounds.
`upper`	An optional set of upper bounds on the seven URSA response parameters. Any values set to NA will be filled with default calculated bounds.

Value

An object of class `braidAltFit` with the following values:

- `concs`: The array of concentrations passed to the functions
- `act`: The vector of measurements associated with the given dose pairs
- `weights`: The vector of weights for the given measurements, set to 1 for all measurements by default
- `method`: Specifying the alternate surface model being used (in this case "URSA")
- `coefficients`: A length-seven parameter vector specifying the URSA response surface
- `fitted.values`: The predicted response surface value for the given dose pairs and best-fit response surface
- `residuals`: The difference between the predicted and measured values for the given dose pairs, always equal to "measured minus predicted"
- `direction`: The direction value passed to the function
- `pbounds'`: A 2-by-7 array of bounds on the URSA parameters used in the parameter optimization

References

Greco, William R, Hyoung Sook Park, and Youcef M Rustum. 1990. "Application of a New Approach for the Quantitation of Drug Synergism to the Combination of Cis-Diamminedichloroplatinum and 1-b-d-Arabinofuranosylcytosine." *Cancer Research* **50** (17): 5318–27.

Examples

```
ufit1 <- fitUrsaModel(measure ~ concA + concB, additiveExample)
coef(ufit1)

ufit2 <- fitUrsaModel(measure ~ concA + concB, synergisticExample,
                      direction = 1, lower=c(NA, NA, NA, NA, NA, 0, 0))
coef(ufit2)
```

`incompleteExample` *Example Partial or Incomplete Surface*

Description

A synthetically generated response surface using parameter vector describing a surface with one only barely detectable effect. The surface was generated with IDMA of 1, IDMB of 100, na of 3, nb of 3, kappa of 0, E0 of 0, EfA of 1, EfB of 0.1, and Ef of 1. Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
incompleteExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

---

`invertBraidModel` *Invert a BRAID Response Surface Model*

Description

Given a particular effect and one of the doses in a combined action response surface, this function calculates the other dose that will produce the desired effect. Used in the estimation of the IAE (see `estimateIAE()`) but also useful for calculating something like the IC50 of one drug in the presence of various doses of the other. `invertBraidModelA` and `invertBraidModelB` are convenience wrapper functions that set `DA` or `DB` to `NULL` to estimate the necessary concentrations of drug A and drug B respectively.

Usage

```
invertBraidModel(
  DA = NULL,
  DB = NULL,
  effect,
  bpar,
  invalidNA = FALSE,
  lowerBound = FALSE
)

invertBraidModel_A(DB, effect, bpar, invalidNA = FALSE, lowerBound = FALSE)

invertBraidModel_B(DA, effect, bpar, invalidNA = FALSE, lowerBound = FALSE)
```

Arguments

DA	If not NULL, a vector of doses of drug A. Must be length 1 or the same length as effect. Only one of DA and DB may be not null.
DB	If not NULL, a vector of doses of drug B. Must be length 1 or the same length as effect. Only one of DA and DB may be not null.
effect	A vector of desired effect values to be reached. Must be length 1 or the same length as whichever of DA or DB is not null.
bpar	A BRAID response surface parameter vector (see evalBraidModel() for details)
invalidNA	Specifies what to do with values that are outside the range of the given BRAID model or doses. If FALSE (the default), values "below" the given range will be set to zero, and values "above" the given range will be set to Inf. If TRUE, all invalid values will be set to NA.
lowerBound	Primarily used by estimateIAE(). If set to TRUE, will return the lowest non-negative dose that produces an effect no greater than the specified effect, rather than the highest

Value

A vector of concentrations the same length as either DA or DB (whichever is not NULL) and/or effect, representing the concentration of the other drug producing the specified effect in combination with the given dose of the provided drug

Examples

```
baseIC <- invertBraidModel_A(
    DB=0,
    effect=seq(10,90,by=10),
    bpar=c(1, 1, 3, 3, 2, 0, 100, 100, 100)
)

potentiatedIC <- invertBraidModel_A(
    DB=1,
```

```
    effect=seq(10,90,by=10),
    bpar=c(1, 1, 3, 3, 2, 0, 100, 100, 100)
)
```

`invertFlippedBraidModel`

Invert Flipped BRAID Surfaces

Description

Given a particular effect and one of the doses in a flipped combined action response surface, this function calculates the other dose that will produce the desired effect.`invertFlippedBraidModelA` and `invertFlippedBraidModelB` are convenience wrapper functions that set `DA` or `DB` to `NULL` to estimate the necessary concentrations of drug A and drug B respectively.

Usage

```
invertFlippedBraidModel(
  DA = NULL,
  DB = NULL,
  effect,
  bpar,
  flip,
  invalidNA = FALSE,
  lowerBound = FALSE
)

invertFlippedBraidModel_A(
  DB,
  effect,
  bpar,
  flip,
  invalidNA = FALSE,
  lowerBound = FALSE
)

invertFlippedBraidModel_B(
  DA,
  effect,
  bpar,
  flip,
  invalidNA = FALSE,
  lowerBound = FALSE
)
```

Arguments

DA	If not NULL, a vector of doses of drug A. Must be length 1 or the same length as effect. Only one of DA and DB may be not null.
DB	If not NULL, a vector of doses of drug B. Must be length 1 or the same length as effect. Only one of DA and DB may be not null.
effect	A vector of desired effect values to be reached. Must be length 1 or the same length as whichever of DA or DB is not null.
bpar	Flipped-BRAID parameter of the flipped response surface. See evalFlippedBraidModel() for more information on specifying atypical surfaces
flip	String specifying the direction or directions of the surface's flip. Must be one of "A", "B", or "both".
invalidNA	Specifies what to do with values that are outside the range of the given BRAID model or doses. If FALSE (the default), values "below" the given range will be set to zero, and values "above" the given range will be set to Inf. If TRUE, all invalid values will be set to NA.
lowerBound	Primarily used by estimateIAE(). If set to TRUE, will return the lowest non-negative dose that produces an effect no greater than the specified effect, rather than the highest

Value

A vector of concentrations the same length as either DA or DB (whichever is not NULL) and/or effect, representing the concentration of the other drug producing the specified effect in combination with the given dose of the provided drug

Examples

```
fbfit <- fitProtectiveBraid_A(measure ~ concA + concB,
                              protectiveExample, getCIs=FALSE)

invertFlippedBraidModel_A(DB=0, effect=0.5, coef(fbfit), fbfit$flip)
invertFlippedBraidModel_A(DB=0.75, effect=0.5, coef(fbfit), fbfit$flip)
```

---

kappaPrior *Braid kappa Bayesian Prior*

Description

Generates a Bayesian prior object on the BRAID parameter kappa to stabilize parameter fitting

Usage

```
kappaPrior(spread, strength = "moderate")
```

Arguments

spread	Rough estimate of the standard deviation of measurement noise or errors expected in a given data set. Commonly used values are standard deviation of negative/positive controls or root mean squared error of a preliminary surface fit.
strength	String indicating the influence of the BRAID prior on the resulting fit. Must be one of "mild", "moderate" (the default), "high", or "none".

Value

An object of class `kappaPrior` containing two numeric elements, `spread`, and `strength`. Used in BRAID fitting functions to stabilize the parameter kappa

Examples

```
prior <- kappaPrior(0.05,"mild")

bfit <- braidrm(measure ~ concA + concB, incompleteExample,
                prior=prior, getCIs=FALSE)
summary(bfit)
```

`oppositionalExample` *Example Oppositional Surface*

Description

A synthetically generated response surface using a flipped "oppositional" parameter vector. The surface was generated with IDMA of 1, IDMB of 1, na of 3, nb of 3, kappa of -0.5, E0 of 0, EfA of 1, EfB of -0.5, and Ef of 1; the surface was flipped along the drug B axis (so `flip` was set to "B"). Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
oppositionalExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

protectiveExample *Example Protective Surface*

Description

A synthetically generated response surface using a flipped "protective" parameter vector. The surface was generated with IDMA of 0.5, IDMB of 2, na of 3, nb of 3, kappa of 2, E0 of 0, EfA of 1, EfB of 0, and Ef of 0; the surface was flipped along the drug A axis (so `flip` was set to "A"). Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
protectiveExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

synergisticExample *Example Synergistic Surface*

Description

A synthetically generated response surface using a synergistic parameter vector. The surface was generated with IDMA of 1, IDMB of 1, na of 3, nb of 3, kappa of 2, E0 of 0, EfA of 1, EfB of 1, and Ef of 1. Every pair of concentrations is sampled once, with concentrations of 0 and a seven-point two-fold dilution from 0.125 to 8. "Measurements" were sampled from a normal noise distribution around ground truth values with a standard deviation of 0.07.

Usage

```
synergisticExample
```

Format

A data frame with 64 rows and 4 columns

concA The concentration of drug A

concB The concentration of drug B

truth The true response surface value at the given dose pair

measure The sampled noisy measurement of the response surface at the given dose pair

Index