Package 'MCPMod'

July 21, 2025

Type Package

Title Design and Analysis of Dose-Finding Studies

Version 1.0-10.1

Date 2017-08-05

Author Bjoern Bornkamp, Jose Pinheiro and Frank Bretz

Maintainer Bjoern Bornkamp <bornkamp@statistik.tu-dortmund.de>

Depends mvtnorm, lattice, R (>= 2.4.1)

Description Implements a methodology for the design and analysis of dose-response studies that combines aspects of multiple comparison procedures and modeling approaches (Bretz, Pinheiro and Branson, 2005, Biometrics 61, 738-748, <doi:10.1111/j.1541-0420.2005.00344.x>).
The package provides tools for the analysis of dose finding trials as well as a variety of tools necessary to plan a trial to be conducted with the MCP-Mod methodology.
Please note: The 'MCPMod' package will not be further developed, all future development of the MCP-Mod methodology will be done in the 'DoseFinding' R-package.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-09 15:09:13 UTC

Contents

`MCPMod-package` *Design and Analysis of Dose-Finding Studies*

Description

This package implements a methodology for dose-response analysis that combines aspects of multiple comparison procedures and modeling approaches (Bretz, Pinheiro and Branson, (2005)). The package provides tools for the analysis of dose finding trials as well as a variety of tools necessary to plan a trial to be conducted with the MCPMod methodology. **Note: The MCPMod package will not be further developed, all future development of the MCP-Mod methodology will be done in the DoseFinding R-package, which already contains an extended version of MCP-Mod, and additional functions useful for planning and analysing dose-finding trials.**

Details

Package:	MCPMod
Type:	Package
Version:	1.0-9
Date:	2016-11-24
License:	GPL-3

Author(s)

Bjoern Bornkamp, Jose Pinheiro and Frank Bretz

Maintainer: Bjoern Bornkamp <bornkamp@statistik.tu-dortmund.de>

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009), MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Bretz, F., Pinheiro, J. C., and Branson, M. (2005), Combining multiple comparisons and modeling techniques in dose-response studies, *Biometrics*, **61**, 738–748

Pinheiro, J. C., Bornkamp, B., and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

Pinheiro, J. C., Bretz, F., and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

Examples

```
# detailed information regarding MCP-Mod methodology
# and R-package available via vignette("MCPMod")
## Not run:
# planning a trial for MCPMod
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential = c(85),
               betaMod = matrix(c(0.33, 2.31, 1.39, 1.39), byrow=TRUE,nrow=2))
plotModels(models, doses, base = 0, maxEff = 0.4, scal = 200)
sSize <- sampSize(models, doses, base = 0, maxEff = 0.4, sigma = 1,
          upperN = 80, scal = 200, alpha = 0.05)
sSize
plM <- planMM(models, doses, n = rep(sSize$samp.size,6), scal=200, alpha = 0.05)
plM
plot(plM)

# analysing a trial
data(biom)
models <- list(linear = NULL, linlog = NULL, emax = 0.2,
            exponential = c(0.279,0.15), quadratic = c(-0.854,-1))

dfe <- MCPMod(biom, models, alpha = 0.05, dePar = 0.05, pVal = TRUE,
           selModel = "maxT", doseEst = "MED2", clinRel = 0.4, off = 1)
# detailed information is available via summary
summary(dfe)
# plots data with selected model function
plot(dfe, complData = TRUE, cR = TRUE)

## End(Not run)
```

betaMod *Beta model*

Description

The beta model is defined as

$$f(d,\theta) = E_0 + E_{max} B(\delta_1,\delta_2)(d/scal)^{\delta_1}(1-d/scal)^{\delta_2}$$

where

$$B(\delta_1,\delta_2) = (\delta_1+\delta_2)^{\delta_1+\delta_2}/(\delta_1^{\delta_1}\delta_2^{\delta_2})$$

Usage

```
betaMod(dose, e0, eMax, delta1, delta2, scal)
```

Arguments

dose	Dose variable
e0	Placebo effect
eMax	Maximum effect
delta1	delta1 parameter
delta2	delta2 parameter
scal	Scale parameter (not estimated in the code)

Details

The beta model is intended to capture non-monotone dose-response relationships and is more flexible than the quadratic model. The kernel of the beta model function consists of the kernel of the density function of a beta distribution on the interval [0,scal]. The parameter scal is not estimated but needs to be set to a value larger than the maximum dose via the argument scal.

Value

Response value

See Also

logistic, sigEmax, linlog, linear, quadratic, emax, exponential

biom *Biometrics Dose Response data*

Description

An example data set for dose response studies. This data set was used in Bretz et al. (2005) to illustrate the MCPMod methodology.

Usage

```
data(biom)
```

Format

A data frame with 100 observations on the following 2 variables.

`resp` a numeric vector containing the response values

`dose` a numeric vector containing the dose values

Source

Bretz, F., Pinheiro, J. C., and Branson, M. (2005), Combining multiple comparisons and modeling techniques in dose-response studies, *Biometrics*, **61**, 738–748

critVal *Calculate critical value for multiple contrast test*

Description

This function calculates the critical value for a multiple contrast test via numerical integration (using the methods implemented in the `mvtnorm` package).

Usage

```
critVal(cMat, n, alpha = 0.025, control = mvtnorm.control(),
        twoSide = FALSE, corMat = NULL)
```

Arguments

`cMat`	A matrix with the contrasts in the columns
`n`	A vector giving the sample size per group. If only one number is specified it is assumed that the sample sizes are balanced.
`alpha`	Level of significance (defaults to 0.025)
`control`	A list of options for the `pmvt` and `qmvt` functions as produced by `mvtnorm.control`
`twoSide`	Logical indicating if two or one-sided critical value should be calculated.
`corMat`	Optional: correlation of contrasts matrix can be specified

Value

Critical value

References

Bretz, F., Pinheiro, J. and Branson, M. (2005), Combining Multiple Comparisons and Modeling Techniques in Dose-Response Studies, *Biometrics*, **61**, 738–748

Hothorn, T., Bretz, F., and Genz, A. (2001), On multivariate t and Gauss probabilities in R, *R News*, **1**, 27–29

See Also

planMM, mvtnorm.control

Examples

```
# Calculation of critical value for Dunnett test
# Set up contrast matrix (3 active doses)
CM <- rbind(-1,diag(3))
# 30 patients per group, one-sided alpha 0.05
critVal(CM, n=30, alpha = 0.05)
# Example from R News 1(2) p. 28, 29
CM <- c(1, 1, 1, 0, 0, -1, 0, 0, 1, 0, 0, -1, 0, 0,
       1, 0, 0, 0, -1, -1, 0, 0, -1, 0, 0)
CM <- t(matrix(CM, ncol = 5))
critVal(CM, n=c(26, 24, 20, 33, 32), alpha = 0.05,
    twoSide = TRUE)
```

---

emax *Emax model*

Description

The model function for the Emax model is defined as

$$f(d, \theta) = E_0 + E_{max} \frac{d}{ED_{50} + d}$$

Usage

```
emax(dose, e0, eMax, ed50)
```

Arguments

dose	Dose variable
e0	Placebo effect
eMax	Asymptotic maximum change from placebo effect
ed50	Dose giving half of the asymptotic maximum effect

Details

The emax model is used to represent monotone, concave dose-response shapes. To distinguish it from the more general sigmoid emax model it is sometimes also called hyperbolic emax model.

Value

Response value

References

MacDougall, J. (2006). Analysis of dose-response studies - Emax model,*in* N. Ting (ed.), *Dose Finding in Drug Development*, Springer, New York, pp. 127–145

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

`sigEmax`, `logistic`, `betaMod`, `linlog`, `linear`, `quadratic`, `exponential`

`exponential` *Exponential (power) model*

Description

The model function for the exponential model is defined as

$$f(d, \theta) = E_0 + E_1(exp(d/\delta) - 1)$$

Usage

```
exponential(dose, e0, e1, delta)
```

Arguments

`dose`	Dose variable
`e0`	Placebo effect
`e1`	E1 parameter
`delta`	Delta parameter, controlling the convexity of the model.

Details

This model is intended to capture a possible sub-linear or a convex dose-response relationship.

Value

Response value

References

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

logistic, sigEmax, linlog, linear, quadratic, emax, betaMod

`fullMod` *Calculate location and scale parameters for candidate set of models*

Description

Calculates location and scale parameters for all models in the candidate set using the maximum approach from Pinheiro et al. (2006). This is done by repeatedly calling the `getPars` function.

Usage

```
fullMod(models, doses, base, maxEff, off = 0.1*max(doses), scal = 1.2 * max(doses))
```

Arguments

`models`	A list specifying the candidate models. The names of the list entries should be equal to the names of the model functions. The list entries should be equal to the guesstimates
`doses`	Dose levels to be administered
`base`, `maxEff`	Baseline effect and maximum change from baseline to be used for calculating the location and scale parameters of the model
`off`	Offset parameter for the linear in log model
`scal`	Scale parameter for the beta model

Value

Returns an object of class fullMod, containing all parameter values for the models in a list.

References

Bornkamp B., Pinheiro J. C., Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

getPars, sampSize, powerMM, plotModels, LP

Examples

```
doses <- c(0, 10, 25, 50, 100, 150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential = c(85),
               betaMod = matrix(c(0.33, 2.31, 1.39, 1.39), byrow=TRUE,nrow=2))
fMod <- fullMod(models, doses, base = 0, maxEff = 0.4, scal = 200)
plot(fMod) # automatically calls the plotModels function
```

---

genDFdata *Simulate dose-response data*

Description

The function simulates normally distributed dose-response data, according to a prespecified dose-response model (or mean vector) and a given standard deviation.

Usage

```
genDFdata(model, argsMod, doses, n, sigma, mu = NULL)
```

Arguments

model	Character string giving the name of a model function. The first argument of the model function should be the dose variable.
argsMod	A vector with the arguments for the model function.
doses	Dose levels to be used.
n	Group sample sizes.
sigma	Standard deviation.
mu	If model is not specified mu is used to determine the mean vector of the observations.

Value

A data frame with two columns called dose and resp, corresponding to the dose and simulated response values.

Examples

```
# use emax model
genDFdata("emax", c(e0 = 0.2, eMax = 1, ed50 = 0.05), c(0,0.05,0.2,0.6,1), 20, 1)
# use fixed mean vector
genDFdata(mu = 1:5, doses = 0:4, n = c(20, 20, 10, 5, 1), sigma = 0.2)
```

getPars *Calculate location and scale parameters*

Description

Given the baseline, the maximum effect and the standardized model parameters this function calculates the location and scale parameters in the model function using the maximum approach, see Pinheiro et al. (2006) for the basic idea.

Usage

```
getPars(model, doses, initEstim, base, maxEff,
        off = 0.1 * max(doses), scal = 1.2 * max(doses))
```

Arguments

model	A character string with the model name. Built-in models have their full parameterization derived internally. For user-defined models, it is assumed that a function named as "Par" appended to end of model name exists (e.g., for model = "cubic", it is assumed that there is function "cubicPar" that calculates the necessary parameters; this function is assumed to have arguments "doses", "initEstim", "base", and "maxEff", in that order (see below for an example).
doses	Doses to be used in design
initEstim	Vector of guesstimates
base	Expected baseline effect
maxEff	Expected maximum change from baseline
off	Offset parameter for the linear in log model (default 1).
scal	Scale parameter for the beta model (default: 20 perc. larger than maximum dose).

Value

Vector containing all model parameters.

References

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

fullMod

Examples

```
doses <- c(0, 10, 25, 50, 100, 150)
getPars("emax", doses, 25, 0, 0.4)
getPars("logistic", doses, c(50, 10.88111), 0, 0.4) # compare JBS 16, p.650
getPars("betaMod", doses, initEstim = c(0.33, 2.31), base = 0,
        maxEff = 0.4)
#example for user model
userMod <- function(dose, e0, eMax, ed50, h){
  e0 + eMax * ( dose^h / (ed50^h + dose^h) )
}
# function to return location and scale parameters
userModPar <- function(dose, initEstim, base, maxEff){
  # function to get linear parameters
  # ed50 parameter assumed to be first in initEstim
  ed50 <- initEstim[1]
  h <- initEstim[2]
  dmax <- max(dose)
  emax <- maxEff*(ed50^h+dmax^h)/dmax^h
  c(base, emax, initEstim)
}
getPars("userMod", doses, initEstim = c(50,2), base = 0, maxEff = 1)
```

---

guesst *Calculate guesstimates based on prior knowledge*

Description

Calculates guesstimates for standardized model parameter(s) using the general approach described in Pinheiro et al. (2006).

Usage

```
guesst(d, p, model = c("emax", "exponential", "logistic", "quadratic",
       "betaMod", "sigEmax"), less = TRUE,  local = FALSE,
       dMax, Maxd, scal)
```

Arguments

d	Vector containing dose value(s).
p	Vector of expected percentages of the maximum effect achieved at d.
model	Character string. Should be one of "emax", "exponential", "quadratic", "betaMod", "sigEmax".
less	Logical, only needed in case of quadratic model. Determines if d is smaller (less=TRUE) or larger (less=FALSE) than dopt (see Pinheiro et al. (2006) for details).
local	Logical indicating whether local or asymptotic version of guesstimate should be derived (defaults to FALSE). Only needed for emax, logistic and sigEmax model. When local=TRUE the maximum dose must be provided via Maxd.

dMax	Dose at which maximum effect occurs, only needed for the beta model
Maxd	Maximum dose to be administered in the trial
scal	Scale parameter, only needed for the beta model

Details

Calculates guesstimates for the parameters of the standardized model function based on the prior expected percentage of the maximum effect at certain dose levels. Note that this function should be used together with the `plotModels` function to ensure that the guesstimates are reflecting the prior beliefs.

For the logistic and sigmoid emax models at least two pairs (d,p) need to be specified.

For the beta model the dose at which the maximum effect occurs (dMax) has to be specified in addition to the (d,p) pair.

For the exponential model the maximum dose administered (Maxd) needs to be specified in addition to the (d,p) pair.

For the quadratic model one (d,p) pair is needed. It is advisable to specify the location of the maximum within the dose range with this pair.

For the emax, sigmoid Emax and logistic model one can choose between a local and an asymptotic version. In the local version one explicitly forces the standardized model function to pass through the specified points (d,p). For the asymptotic version it assumed that the standardized model function is equal to 1 at the largest dose (this is the approach described in Pinheiro et al. (2006)). If the local version is used, convergence problems with the underlying nonlinear optimization can occur.

Value

Returns a numeric vector containing the guesstimates.

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bretz, F., and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.), *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

`emax`, `logistic`, `betaMod`, `sigEmax`, `quadratic`, `exponential`, `plotModels`

Examples

```
# Emax model
# Expected percentage of maximum effect: 0.8 is associated with
# dose 0.3 (d,p)=(0.3, 0.8), dose range [0,1]
emx1 <- guesst(d=0.3, p=0.8, model="emax")
# local approach
emx2 <- guesst(d=0.3, p=0.8, model="emax", local = TRUE, Maxd = 1)
# plot models
models <- list(emax=c(emx1, emx2))
```

```
plotModels(models, c(0,1), base= 0, maxEff = 1)

# Logistic model
# Select two (d,p) pairs (0.2, 0.5) and (0.6, 0.95)
lgc1 <- guesst(d = c(0.2, 0.5), p = c(0.6, 0.95), "logistic")
# local approach
lgc2 <- guesst(d = c(0.2, 0.5), p = c(0.6, 0.95), "logistic",
               local = TRUE, Maxd = 1)
# plot models
models <- list(logistic = matrix(c(lgc1, lgc2), byrow = TRUE, nrow = 2))
plotModels(models, c(0,1), base= 0, maxEff = 1)

# Beta Model
# Select one pair (d,p): (0.5,0.5)
# dose, where maximum occurs: 0.8
bta <- guesst(d=0.5, p=0.5, model="betaMod", dMax=0.8, scal=1.2)
# plot
models <- list(betaMod = bta)
plotModels(models, c(0,1), base= 0, maxEff = 1)

# Sigmoid Emax model
# Select two (d,p) pairs (0.2, 0.5) and (0.6, 0.95)
sgE1 <- guesst(d = c(0.2, 0.5), p = c(0.6, 0.95), "sigEmax")
# local approach
sgE2 <- guesst(d = c(0.2, 0.5), p = c(0.6, 0.95), "sigEmax",
               local = TRUE, Maxd = 1)
models <- list(sigEmax = matrix(c(sgE1, sgE2), byrow = TRUE, nrow = 2))
plotModels(models, c(0,1), base= 0, maxEff = 1)

# Quadratic model
# For the quadratic model it is assumed that the maximum effect occurs at
# dose 0.7
quad <- guesst(d = 0.7, p = 1, "quadratic")
models <- list(quadratic = quad)
plotModels(models, c(0,1), base= 0, maxEff = 1)

# exponential model
# (d,p) = (0.8,0.5)
expo <- guesst(d = 0.8, p = 0.5, "exponential", Maxd=1)
models <- list(exponential = expo)
plotModels(models, c(0,1), base= 0, maxEff = 1)
```

IBS *Irritable Bowel Syndrome Dose Response data*

Description

A subset of the data used by (Biesheuvel and Hothorn, 2002). Except for the dose and the response variable all variables from the original data set are removed. The data are part of a dose ranging trial on a compound for the treatment of the irritable bowel syndrome with four active treatment arms,

corresponding to doses 1,2,3,4 and placebo. Note that the original dose levels have been blinded in this data set for confidentiality. The primary endpoint was a baseline adjusted abdominal pain score with larger values corresponding to a better treatment effect. In total 369 patients completed the study, with nearly balanced allocation across the doses.

Usage

```
data(IBS)
```

Format

A data frame with 369 observations on the following 2 variables.

`dose` a numeric vector

`resp` a numeric vector

Source

Biesheuvel, E. and Hothorn, L. A. (2002). Many-to-one comparisons in stratified designs, *Biometrical Journal*, **44**, 101–116

`linear` *Linear Model*

Description

The model function for the linear model is defined as

$$f(d, \theta) = E_0 + \delta d$$

Usage

```
linear(dose, e0, delta)
```

Arguments

`dose`	Dose variable
`e0`	Placebo effect
`delta`	Slope parameter

Value

Response value

References

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

logistic, sigEmax, linlog, exponential, quadratic, emax, betaMod

---

`linlog` *Linear in log dose Model*

Description

The model function for the linear in log model is defined as

$$f(d, \theta) = E_0 + \delta \log(d + off)$$

where off is an offset parameter not estimated in the code.

Usage

```
linlog(dose, e0, delta, off = 1)
```

Arguments

`dose`	Dose variable
`e0`	Placebo effect
`delta`	Slope parameter
`off`	Offset value to avoid problems with dose=0 (treated as a fixed value in the code)

Details

The linear in log-dose model is intended to capture concave shapes. The parameter `off` is not estimated in the code but set to a pre-specified value.

Value

Response value

References

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

logistic, sigEmax, linear, exponential, quadratic, emax, betaMod

`logistic` *Logistic Model*

Description

The model function for the logistic model is defined as

$$f(d, \theta) = E_0 + E_{\max} / \left\{1 + \exp\left[\left(ED_{50} - d\right) / \delta\right]\right\}$$

Usage

```
logistic(dose, e0, eMax, ed50, delta)
```

Arguments

`dose`	Dose variable
`e0`	Left-asymptote parameter, corresponding to a basal effect level (not the placebo effect, though).
`eMax`	Asymptotic maximum change in effect from the basal level.
`ed50`	Dose giving half of the asymptotic maximum effect.
`delta`	Parameter controlling determining the steepness of the curve.

Details

The logistic model is intended to capture general monotone, sigmoid dose-response relationships.

Value

Response value

References

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

`betaMod`, `logistic`, `sigEmax`, `linlog`, `linear`, `quadratic`, `exponential`

LP *Sensitivity analysis for misspecification of standardized model parameters*

Description

Calculates the loss in power associated with misspecification of the standardized model parameters for a specific model.

Usage

```
LP(models, model, type = c("both", "LP1", "LP2"), paramRange,
   doses, base, maxEff, sigma, n, len = c(10, 1), nr = 1,
   alpha = 0.025, twoSide = FALSE, off = 0.1 * max(doses),
   scal = 1.2 * max(doses), control = mvtnorm.control())
```

Arguments

models	A list specifying the candidate models. This can also be a fullMod object, then the arguments base, maxEff, off and scal are ignored
model	Character string giving the model for which the sensitivity should be investigated.
type	Character string: One of "LP1", "LP2" or "both".
paramRange	Numeric of length two, giving lower and upper limits for standardized model parameter values when the model has just one standardized model parameter. For models with two standardized model parameters a 2x2 matrix with the boundaries for each standardized model parameter in the rows. See examples for details.
doses	Dose levels to be administered
base	Baseline effect
maxEff	Maximum change from baseline
sigma	Standard deviation
n	Numeric vector of sample sizes per group. In case just one number is specified, it is assumed that all group sample sizes are equal to this number.
len	Number of points in the standardized model parameter range on which LP is calculated. Has to be of length 2 in case of models with 2 standardized model parameters.
nr	Numeric giving the number of the model (in the order given in the model argument) in case there is more than one model from one model class in the candidate set (e.g. two emax models).
alpha	Level of significance (default: 0.025)
twoSide	Logical indicating whether a two sided or a one sided test is performed (defaults to one-sided).

off	Offset parameter for the linear in log model (default 10 perc. of maximum dose).
scal	Scale parameter for the beta model (default 20 perc. larger than maximum dose).
control	A list of options for the pmvt and qmvt functions as produced by mvtnorm.control

Details

For a given set of candidate models the power-sensitivity of the multiple contrast test with respect to misspecification of the guesstimates is investigated. Two measures to measure loss in power ("LP1" or "LP2") can be used. Roughly LP1 can be interpretated as the difference between the power that "was intended" (nominal power), when designing the study and "what one actually gets" (actual power).
LP2 can be interpreted as the difference between "what could be achieved knowing the true value of the parameter in advance" (potential power) and "what one actually gets". For a detailed definition see the reference below. The power values are calculated on a number of points specified by the len argument. The calculation of LP2 is computationally more demanding as the optimal contrasts and the critical value need to be recalculated for each point in the standardized model parameter space.

Value

An object of class LP, i.e. a matrix containing the different alternative standardized model parameters, associated potential/actual power values and the loss in power values.

References

Bornkamp B., Pinheiro J. C., Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

plot.LP, guesst

Examples

```
## Not run:
doses <- c(0,10,25,50,100,150)
models <- list(linear=NULL, emax=c(25),
           logistic=c(50,10.88111), exponential=c(85),
           betaMod=matrix(c(0.33,2.31,1.39,1.39),byrow=TRUE,nrow=2))

# Examples from JBS paper, p.654
LPobj <- LP(models, model = "emax", type = "both", paramRange = c(10,70),
          doses = doses, base = 0, maxEff = 0.4, sigma = 1, n = 60,
          alpha = 0.05, len = 15, scal = 200)
print(LPobj)
plot(LPobj)

# for exponential model with fullMod and LP1:
```

```
fMod <- fullMod(models, doses, base = 0, maxEff = 0.4, scal=200)
LPobj <- LP(fMod, "exponential", "LP1", c(50, 120), sigma = 1,
            alpha = 0.05, len = 20, n = 60)
plot(LPobj)

# Examples for models with two standardized model parameters
LP(models, "betaMod", "LP1",
   paramRange = matrix(c(0.3,1.9,0.4,2.5),nrow=2),
   doses, 0, 0.4, 1, 60, alpha=0.05, len=c(10,4), scal=200)
#  Time consuming example
LPobj <- LP(models, "logistic", "both",
            paramRange = matrix(c(40,5,60,15),nrow=2),
            doses, 0, 0.4, 1, 60, alpha=0.05, len=c(10,4), scal=200)
plot(LPobj)

## End(Not run)
```

MCPMod *Perform MCPMod analysis of a data set*

Description

Tests for a dose-response effect with a model-based multiple contrast test and estimates a target dose with regression techniques. For details see Bretz et al. (2005) or the enclosed vignette, available via the command `vignette("MCPMod")`.

Usage

```
MCPMod(data, models = NULL, contMat = NULL, critV = NULL, resp = "resp",
       dose = "dose", off = NULL, scal = NULL, alpha = 0.025,
       twoSide = FALSE, selModel = c("maxT", "AIC", "BIC", "aveAIC",
       "aveBIC"), doseEst = c("MED2", "MED1", "MED3", "ED"), std = TRUE,
       start = NULL, uModPars = NULL, addArgs = NULL, dePar = NULL,
       clinRel = NULL, lenDose = 101, pW = NULL,
       control = list(maxiter = 100, tol = 1e-06, minFactor = 1/1024),
       signTtest = 1, pVal = FALSE, testOnly = FALSE,
       mvtcontrol = mvtnorm.control(), na.action = na.fail, uGrad = NULL)
```

Arguments

`data`	Data frame containing the dose and the response data. The code assumes the columns to be named "dose" and "resp". Other names can be handed over via the `dose` and `resp` argument see below.
`models`	A list specifying the candidate models. The names of the list entries should be equal to the names of the model functions. The list entries should be equal to the guesstimates. See the Examples (ii) for details on this topic. If the `contMat` argument is specified, this argument is ignored, see Examples (iv).

`contMat` Optional matrix containing the optimal contrasts in the columns. The names of the columns should be equal to the underlying model function names. If specified the code does not calculate the optimal contrasts.

`critV` Optional numeric specifying the critical value to be used in the multiple contrast test.

`resp` Character string giving the name of the response column for the data frame specified in `data` (default: "resp").

`dose` Character string giving the name of the dose column for the data frame specified in `data` (default: "dose").

`off` Fixed offset parameter needed when the linear in log model is used. See also documentation of the linear in log model: `linlog`. When `off = NULL` by default (maximum dose)*0.1 is used for `off`.

`scal` Fixed scale parameter needed when the beta model is used. See also documentation of the beta model: `betaMod`. When `scal = NULL` by default (maximum dose)*1.2 is used for `scal`.

`alpha` Level of significance for the multiple contrast test (defaults to 0.025)

`twoSide` Optional logical value deterimining whether two-sided or one-sided testing should be performed. Defaults to FALSE, so one-sided testing.

`selModel` Optional character vector specifying the model selection criterion for dose estimation. Possible values are
"maxT": Selects the model corresponding to the largest t-statistic (this is the default).
"AIC": Selects model with smallest AIC
"BIC": Selects model with smallest BIC
"aveAIC": Uses a weighted average of the models corresponding to the significant contrasts. The model weights are chosen by the formula: wi = exp(-0.5AICi)/sum(exp(-0.5AICi)). See Buckland et al. (1997) for details.
"aveBIC": Same as "aveAIC", but the BIC is used to calculate the model weights.

`doseEst` Determines which dose to estimate and which dose estimator to use, possible values are "MED2", "MED1", "MED3" and "ED". See Bretz et al. (2005) for the definition of MED1-MED3. If ED is specified, the dose that gives a prespecified percentage of the maximum effect is returned.

`std` Optional logical value determining, whether standardized versions should be assumed for calculation of the optimal contrasts. If FALSE all model parameters need to be specified in the models argument (also location and scale parameters).

`start` List containing starting values for the nls fitting algorithm. The names of the list elements need to be equal to the names of the model functions. The names of the starting vector should equal the corresponding names for the model parameters. For built-in models starting values need to be provided only for the non-linear parameters. In case of a user model (not built in) starting values for the all parameters need to be supplied. (see Examples (iii) for details).

`uModPars` Optional character vector with names/expressions of user-defined model parameters (names(start) used by default).

addArgs	Optional character vector with names of additional arguments (variables) to user-defined model.
dePar	Numeric, defining parameter used for dose estimators. For the MED-type estimators dePar determines the confidence level gamma used in the estimator. The used confidence level is given by 1-2*dePar. The default for dePar for MED-type estimators is 0.1. For ED-type estimators dePar determines which effective dose is estimated. Specifying 0.95 for example results in an estimate of the ED95. If the ED estimator is used the default for dePar is 0.5.
clinRel	Numeric specifying the clinical relevance threshold.
lenDose	Numeric vector specifying the number of points in the dose-range to search for the dose estimate, defaults to 101.
pW	Optional vector specifying prior weights for the different models. Should be a named vector with names matching the names of the models list.
control	List of parameters to be used in the calls to the nls function. See also nls.control function.
signTtest	Optional numeric vector multiplied with the test statistics.
pVal	Optional logical determining whether p-values should be calculated, defaults to FALSE. If the critical value is supplied, p-values will not be calculated.
testOnly	Logical value determining, whether only the multiple comparisons test should be performed. See Examples (v) below.
mvtcontrol	A list specifying additional control parameters for the qmvt and pmvt calls in the code, see also mvtnorm.control for details.
na.action	A function which indicates what should happen when the data contain NAs.
uGrad	Function to return the gradient of a user defined model, see Examples (iii) below.

Details

This function performs the multiple comparisons and modelling (MCPMod) procedure presented in Bretz et al. (2005). The method consists of two steps:
(i) MCP step: The function calculates the optimal contrasts (if not supplied) and the contrast test statistics. In the calculation of the critical value and p-values multiplicity is taken into account.
(ii) Modelling step: If there is at least one significant contrast, one model or a combination of models is chosen (depending on the selModel argument) for dose estimation. In case of non-convergence of certain non-linear models the remaining significant models are used. Finally the target dose is estimated.

Built in models are the linear, linear in log, emax, sigmoid emax, logistic, exponential, quadratic and beta model (for their definitions see their help files or Bretz et al. (2005), Pinheiro et al. (2006)). Users may hand over their own model functions for details have a look at the Example (iii).

Value

An object of class MCPMod, with the following entries:

signf	Logical indicating, whether multiple contrast test is significant

model1	Model with largest contrast test statistic
model2	Model(s) used for estimation of target doses
input	A list with entries equal to the input parameters for the function: models, resp, dose, off, scal, alpha, twoSide, first entry of selModel, doseEst, std, dePar, uModArgs, addArgs, start, uGrad, clinRel, lenDose, signTtest, pVal, testOnly
data	The data set.
contMat	The contrast matrix.
corMat	The correlation matrix.
cVal	The critical value for the multiple contrast test.
tStat	The contrast test-statistics. If 'pVal=TRUE' the p-values are also attached.
fm	List containing the dose-response model(s) used for dose-estimation. WARNING: The model fitting is for computational efficiency done based on the group means and for positive non-linear parameters (e.g. the ED50 parameter in the Emax model) the estimate in fm is on log scale. The summary.MCPMod method shows the parameters on their original scale. Hence some care is hence needed when extracting the fitted model objects from fm (in particular when interest is in standard deviations of predictions, parameter estimates...).
tdose	Estimated target dose, in case of model averaging the dose estimates under the individual models are attached.

Note: If testOnly=TRUE, or no model is significant, the object does not contain fm and tdose entries

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Bretz, F., Pinheiro, J. C., and Branson, M. (2005), Combining multiple comparisons and modeling techniques in dose-response studies, *Biometrics*, **61**, 738–748

Pinheiro, J. C., Bornkamp, B., and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

Pinheiro, J. C., Bretz, F., and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

Bretz, F., Pinheiro, J. C., and Branson, M. (2004), On a hybrid method in dose-finding studies, *Methods of Information in Medicine*, **43**, 457–460

Buckland, S. T., Burnham, K. P. and Augustin, N. H. (1997). Model selection an integral part of inference, *Biometrics*, **53**, 603–618

See Also

logistic, sigEmax, linlog, linear, quadratic, emax, betaMod, exponential, plot.MCPMod, mvtnorm.control

Examples

```
## Not run:
# (i)
# example from Biometrics paper p. 743
data(biom)
models <- list(linear = NULL, linlog = NULL, emax = 0.2,
            exponential = c(0.279,0.15), quadratic = c(-0.854,-1))
dfe <- MCPMod(biom, models, alpha = 0.05, dePar = 0.05, pVal = TRUE,
           selModel = "maxT", doseEst = "MED2", clinRel = 0.4, off = 1)
# detailed information is available via summary
summary(dfe)
# plots data with selected model function
plot(dfe)

# example with IBS data
data(IBS)
models <- list(emax = 0.2, quadratic = -0.2, linlog = NULL)
dfe2 <- MCPMod(IBS, models, alpha = 0.05, pVal = TRUE,
             selModel = "aveAIC", clinRel = 0.25, off = 1)
dfe2
# show more digits in the output
print(dfe2, digits = 8)
summary(dfe2, digits = 8)
plot(dfe2, complData = TRUE, cR = TRUE)
plot(dfe2, CI = TRUE)

# simulate dose-response data
dfData <- genDFdata(model = "emax", argsMod = c(e0 = 0.2, eMax = 1,
          ed50 = 0.05), doses = c(0,0.05,0.2,0.6,1), n=20, sigma=0.5)
models <- list(emax = 0.1, logistic = c(0.2, 0.08),
             betaMod = c(1, 1))
dfe3 <- MCPMod(dfData, models, clinRel = 0.4, critV = 1.891,
           scal = 1.5)

## End(Not run)
# (ii) Example for constructing a model list

# Contrasts to be included:
# Model             guesstimate(s) for stand. model parameter(s) (name)
# linear            -
# linear in log     -
# Emax              0.2 (ED50)
# Emax              0.3 (ED50)
# exponential       0.7 (delta)
# quadratic        -0.85 (delta)
# logistic          0.4  0.09 (ED50, delta)
# logistic          0.3  0.1 (ED50, delta)
# betaMod           0.3  1.3 (delta1, delta2)
# sigmoid Emax      0.5  2 (ED50, h)
#
# For each model class exactly one list entry needs to be used.
# The names for the list entries need to be written exactly
```

```
# as the model functions ("linear", "linlog", "quadratic", "emax",
# "exponential", "logistic", "betaMod", "sigEmax").
# For models with no parameter in the standardized model just NULL is
# specified as list entry.
# For models with one parameter a vector needs to be used with length
# equal to the number of contrasts to be used for this model class.
# For the models with two parameters in the standardized model a vector
# is used to hand over the contrast, if it is desired to use just one
# contrast. Otherwise a matrix with the guesstimates in the rows needs to
# be used. For the above example the models list has to look like this

list(linear = NULL, linlog = NULL, emax = c(0.2, 0.3),
     exponential = 0.7, quadratic = -0.85, logistic =
     matrix(c(0.4, 0.3, 0.09, 0.1), nrow = 2),
     betaMod = c(0.3, 1.3), sigEmax = c(0.5, 2))

# Additional parameters (who will not be estimated) are the "off"
# parameter for the linlog model and the "scal" parameter for the
# beta model, which are not handed over in the model list.


# (iii) example for incorporation of a usermodel
# simulate dose response data
dats <- genDFdata("sigEmax", c(e0 = 0, eMax = 1, ed50 = 2, h = 2),
             n = 50, sigma = 1, doses = 0:10)
# define usermodel
userMod <- function(dose, a, b, d){
  a + b*dose/(dose + d)
}
# define gradient
userModGrad <-
    function(dose, a, b, d) cbind(1, dose/(dose+d), -b*dose/(dose+d)^2)
# name starting values for nls
start <- list(userMod=c(a=0, b=1, d=2))
models <- list(userMod=c(0, 1, 1), linear = NULL)
MM1 <- MCPMod(dats, models, clinRel = 0.4, selModel="AIC", start = start,
         uGrad = userModGrad)

# (iv) Contrast matrix and critical value handed over and not calculated
# simulate dose response data
dat <- genDFdata(mu = (0:4)/4, n = 20,
                       sigma = 1, doses = (0:4)/4)
# construct optimal contrasts and critical value with planMM
doses <- (0:4)/4
mods <- list(linear = NULL, quadratic = -0.7)
pM <- planMM(mods, doses, 20)
MCPMod(dat, models = NULL, clinRel = 0.3, contMat = pM$contMat,
     critV = pM$critVal)
## Not run:
# (v) Using MCPMod for mutiple contrast tests only
mu1 <- c(1, 2, 2, 2, 2)
mu2 <- c(1, 1, 2, 2, 2)
mu3 <- c(1, 1, 1, 2, 2)
```

```
mMat <- cbind(mu1, mu2, mu3)
dimnames(mMat)[[1]] <- doses
pM <- planMM(muMat = mMat, doses = doses, n = 20, cV = FALSE)
# calculate p-values
fit <-MCPMod(dat, models = NULL, clinRel = 0.3, contMat = pM$contMat,
        pVal = TRUE, testOnly = TRUE)
summary(fit)

## End(Not run)
```

`modelMeans` *Calculate mean vectors for a given candidate set*

Description

Calculates the mean or standardized mean vectors for a candidate set of models. This function is mainly for internal use.

Usage

```
modelMeans(models, doses, std = TRUE, off = 0.1 * max(doses),
           scal = 1.2 * max(doses))
```

Arguments

`models`	A list of candidate models, or the output of the `fullMod` function (depending on the value of `std`).
`doses`	A numeric vector giving the doses to be administered.
`std`	Logical indicating whether standardized or non-standardized version of model function should be used.
`off`	Offset parameter for linear in log model.
`scal`	Scale parameter for beta model.

Value

Matrix with standardized or non-standardized model means.

Examples

```
doses <- c(0, 10, 25, 50, 100, 150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential = c(85),
               betaMod = matrix(c(0.33, 2.31, 1.39, 1.39), byrow=TRUE,nrow=2))
modelMeans(models, doses, std = TRUE)

# now non-standardized means
Models <- fullMod(models, doses, base = 0, maxEff = 0.4, scal = 200)
modelMeans(Models, doses, std = FALSE)
```

mvtnorm.control *Control options for pmvt and qmvt functions*

Description

Returns a list with control parameters (an object of class GenzBretz) for the `pmvt` and `qmvt` functions from the `mvtnorm` package, see the corresponding documentation for more information.

Usage

```
mvtnorm.control(maxpts = 30000, abseps = 0.001,
                interval = NULL, releps = 0)
```

Arguments

`maxpts`	Maximum number of function values as integer.
`abseps`	Absolute error tolerance as double.
`interval`	A vector containing the end-points of the interval to be searched for the critical value.
`releps`	Relative error tolerance as double.

See Also

`pmvt`, `qmvt`

planMM *Calculate planning quantities for MCPMod*

Description

Calculates the optimal model contrasts, the critical value and the contrast correlation matrix, i.e. the quantities necessary to conduct the multiple contrast test for a given candidate set of dose-response models.

Usage

```
planMM(models, doses, n, off = 0.1 * max(doses), scal = 1.2 * max(doses),
       std = TRUE, alpha = 0.025, twoSide = FALSE,
       control = mvtnorm.control(), cV = TRUE, muMat = NULL)
```

Arguments

models	A list of candidate models
doses	A numeric vector giving the doses to be administered.
n	The vector of sample sizes per group. In case just one number is specified, it is assumed that all group sample sizes are equal to this number.
off	Offset parameter for the linear in log model (default 10 perc of the maximum dose).
scal	Scale parameter for the beta model (default 20 perc. larger than maximum dose).
std	Optional logical indicating, whether standardized version of the models should be assumed.
alpha	Level of significance (default: 0.025)
twoSide	Logical indicating whether a two sided or a one-sided test should be performed. By default FALSE, so one-sided testing.
control	A list of options for the pmvt and qmvt functions as produced by mvtnorm.control
cV	Logical indicating whether critical value should be calculated
muMat	An optional matrix with means in the columns and given dimnames (dose levels and names of contrasts). If specified the models argument should not be specified, see examples below.

Value

An object of class planMM with the following components:

contMat	Matrix of optimal contrasts.
critVal	The critical value for the test (if calculated)
muMat	Matrix of (non-normalized) model means
corMat	Matrix of the contrast correlations.

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Bretz, F., Pinheiro, J., and Branson, M. (2005), Combining Multiple Comparisons and Modeling Techniques in Dose-Response Studies, *Biometrics*, **61**, 738–748

Pinheiro, J. C., Bornkamp, B., and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

critVal

Examples

```
# Example from JBS paper
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = 25,
               logistic = c(50, 10.88111), exponential= 85,
               betaMod=matrix(c(0.33,2.31,1.39,1.39), byrow=TRUE, nrow=2))
plM <- planMM(models, doses, n = rep(50,6), alpha = 0.05, scal=200)
plot(plM)

## Not run:
# example, where means are directly specified
# doses
dvec <- c(0, 10, 50, 100)
# mean vectors
mu1 <- c(1, 2, 2, 2)
mu2 <- c(1, 1, 2, 2)
mu3 <- c(1, 1, 1, 2)
mMat <- cbind(mu1, mu2, mu3)
dimnames(mMat)[[1]] <- dvec
planMM(muMat = mMat, doses = dvec, n = 30)

## End(Not run)
```

`plot.fullMod` *Plot method for fullMod objects*

Description

Plot method for fullMod objects.

Usage

```
## S3 method for class 'fullMod'
plot(x, ...)
```

Arguments

`x`	`fullMod` object
`...`	Additional arguments for the `plotModels` function

See Also

`plotModels`, `fullMod`

plot.LP *Plot method for LP objects*

Description

Graphically displays an LP object.

Usage

```
## S3 method for class 'LP'
plot(x, line = TRUE, type = NULL, spldf = 5, ...)
```

Arguments

x	LP object as obtained from a call to the LP function
line	Logical indicating whether the power values should be smoothed.
type	One of "LP1", "LP2" or "both", availability depending on whether the corresponding values have been calculated in the call to the LP function.
spldf	Numeric determining the degrees of freedom for the smoothing spline which is plotted if line==TRUE. Note that spldf should be larger than 1 and smaller than len (default: 5).
...	Additional arguments.

Details

The function produces a trellis display of the loss in power for different values of the standardized model parameter. A smoothing spline (with spldf degrees of freedom) is fit to these points to give a smooth impression of the loss in power curve. For models with two prior parameters a trellis display is shown with the number of panels equal to len[2]. The number of points on which the power is evaluated is equal to len[1] in each panel, where len is an argument of the LP function.

See Also

LP

Examples

```
## Not run:
doses <- c(0,10,25,50,100,150)
models <- list(linear=NULL, emax=c(25),
           logistic=c(50,10.88111), exponential=c(85),
           betaMod=matrix(c(0.33,2.31,1.39,1.39),byrow=TRUE,nrow=2))

# Examples from JBS paper, p.654
LPobj <- LP(models, model = "emax", type = "both", paramRange = c(10,70),
           doses = doses, base = 0, maxEff = 0.4, sigma = 1, n = 60,
           alpha = 0.05, len = 15, scal = 200)
```

```
plot(LPobj)
plot(LPobj, line = FALSE, type = "LP1")
plot(LPobj, type = "LP1", spldf = 9)

## End(Not run)
```

`plot.MCPMod` *Plot MCPMod model fits*

Description

The function plots the model(s) used for dose estimation.

Usage

```
## S3 method for class 'MCPMod'
plot(x, complData = FALSE, CI = FALSE, clinRel = FALSE, doseEst = FALSE,
     gamma = NULL, models = "all", nrDoseGam = 1,
     colors = c("black", "blue", "black", "gray", "blue"),
     uGrad = NULL, ...)
```

Arguments

`x`	A MCPMod object.
`complData`	Logical indicating whether complete data set or group means should be plotted.
`CI`	Logical indicating whether a confidence interval should be plotted along the model fit(s).
`clinRel`	Logical indicating, whether clinical relevance threshold should be included in plot.
`doseEst`	Logical determining whether dose estimate should be included in plot.
`gamma`	Numeric giving the value for the 1-2*gamma pointwise CI around the predicted mean. if equal to NULL the value determined in the MCPMod call is used. In case a vector of gamma values was used `nrDoseGam` determines which is used.
`models`	Character vector determining, which of the used models should be plotted (only available if model averaging was used)
`nrDoseGam`	In case a vector is specified for `dePar` in the MCPMod function (and gamma in the plot.MCPMod function is NULL), nrDoseGam determines which of these values should be used for the conf. interval and the dose estimate (if doseEst = T).
`colors`	Vector of length 5 with the names of the colors for: predictions, CI, data, clinical relevance threshold, dose estimator
`uGrad`	If a user defined model has been used for dose estimation, the gradient function needs to be handed over via uGrad.
`...`	Additional arguments to xyplot.

See Also

MCPMod

---

plot.planMM *Plotting a planMM object*

Description

This function displays the contrasts or model means obtained from a planMM object.

Usage

```
## S3 method for class 'planMM'
plot(x, superpose = TRUE, xlab = "Dose",
     ylab = NULL, resp = c("contrasts", "means"), ...)
```

Arguments

x	A planMM object.
superpose	Logical, indicating if lines should be superposed.
xlab	Label for x-axis
ylab	Label for y-axis
resp	One of "contrasts" or "means". Determines, whether contrasts or normalized means are plotted.
...	Additional arguments to the xyplot function call.

See Also

planMM

Examples

```
## Not run:
doses <- c(0, 10, 25, 50, 100, 150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential = c(85),
               betaMod = matrix(c(0.33, 2.31, 1.39, 1.39),
                byrow=TRUE,nrow=2))
pM <- planMM(models, doses, 50, scal = 200)
plot(pM)
plot(pM, superpose=FALSE, xlab="Different axis name")
plot(pM, resp = "means")
# example with muMat
dvec <- c(0, 10, 50, 100)
mu1 <- c(1, 2, 2, 2)
mu2 <- c(1, 1, 2, 2)
mu3 <- c(1, 1, 1, 2)
```

```
mMat <- cbind(mu1, mu2, mu3)
dimnames(mMat)[[1]] <- dvec
pM <- planMM(muMat = mMat, doses = dvec, n = 30)
plot(pM)
plot(pM, superpose=FALSE, xlab="Different axis name")

## End(Not run)
```

`plot.powerMM` *Plot method for powerMM objects*

Description

This function plots the result of the `powerMM` function call in a trellis display.

Usage

```
## S3 method for class 'powerMM'
plot(x, superpose = TRUE, line.at = NULL, models = "all",
     summ = NULL, perc = FALSE, xlab = NULL,
     ylab = ifelse(perc, "Power (%)", "Power"), ...)
```

Arguments

`x`	A powerMM object, i.e. a matrix with power values for different sample sizes and models
`superpose`	Logical, indicating if lines should be superposed.
`line.at`	A value, or a vector of values, between 0 and 1, to be drawn as horizontal line in the plot (default: not drawn).
`models`	Character determining which of the models should be included in the plot, "all" and "none" are accepted, else names (or numbers) of models.
`summ`	Summaries to be included in plot; by default the mean, the minimum and the maximum value are displayed.
`perc`	Logical indicating if power values should be in percentage.
`xlab`	Label for x-axis.
`ylab`	Label for y-axis.
`...`	Additional arguments for the `xyplot` function.

References

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

powerMM

Examples

```
## Not run:
# Example from JBS paper
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = 25,
               logistic = c(50, 10.88111), exponential= 85,
               betaMod=matrix(c(0.33,2.31,1.39,1.39), byrow=TRUE, nrow=2))
pM <- powerMM(models, doses, base = 0, maxEff = 0.4, sigma = 1,
               lower = 10, upper = 100, step = 20, scal = 200)
pM
plot(pM)
plot(pM, line.at = 0.8, model = c("emax", "linear"), summ = "mean")
plot(pM, line.at = 0.8, model = "none", summ = c("median", "min"))

## End(Not run)
```

---

plotModels *Plot candidate models*

Description

Produces a trellis display of the model functions in the candidate set. The location and scale parameters of the models are determined by the `base` and `maxEff` arguments.

Usage

```
plotModels(models, doses, base, maxEff, nPoints = 200,
           off = 0.1 * max(doses), scal = 1.2 * max(doses),
           superpose = FALSE, ylab = "Model means",
           xlab = "Dose", ...)
```

Arguments

`models`	A list specifying the candidate models. This can also be a fullMod object, then the arguments `base`, `maxEff`, `off` and `scal` are ignored.
`doses`	Dose levels to be administered
`base`	Expected baseline effect
`maxEff`	Expected maximum change from baseline
`nPoints`	Number of points for plotting
`off`	Offset parameter for the linear in log model (default: 10 percent of maximum dose)

scal	Scale parameter for the beta model (default: 20 percent larger than maximum dose)
superpose	Logical determining, whether model plots should be superposed
ylab, xlab	Label for y-axis and x-axis.
...	Additional arguments to the xyplot call.

References

Bornkamp B., Pinheiro J. C., Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

guesst, fullMod

Examples

```
# JBS example
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = c(25),
            logistic = c(50, 10.88111), exponential = c(85),
            betaMod = matrix(c(0.33, 2.31, 1.39, 1.39),
             byrow=TRUE, nrow=2))
plotModels(models, doses, base = 0, maxEff = 0.4, scal = 200)
# all models in one panel
plotModels(models, doses, base = 0, maxEff = 0.4, scal = 200,
           superpose = TRUE)

# plotModels can also be called using a fullMod object
fM <- fullMod(models, doses, base = 0, maxEff = 0.4, scal = 200)
plotModels(fM)
# or even easier
plot(fM)
```

---

powCalc *Calculate the power for the multiple contrast test*

Description

Given the optimal contrasts, the sample size and a certain 'alternative' (i.e. a mean vector and sigma), the function calculates the power to detect this alternative. See Pinheiro et al. (2006) for details. The function is the building block for the functions powerMM, sampSize and LP. Numerical integration routines from the mvtnorm package are used to calculate the underlying multivariate integrals.

Usage

```
powCalc(cMat, n, alpha = 0.025, delta = NULL, mu = NULL,
        sigma = NULL, cVal = NULL, corMat = NULL,
        twoSide = FALSE, control = mvtnorm.control())
```

Arguments

cMat	Matrix with the contrasts in the columns
n	Numeric vector of sample sizes per group. In case just one number is specified, it is assumed that all group sample sizes are equal to this number
alpha	Level of significance (defaults to 0.025)
delta	Non-centrality vector of the distribution of the test statistic under the alternative.
mu	Mean vector under the alternative. The function then calculates the non-centrality vector itself. Ignored if delta is specified.
sigma	Expected standard deviation of the response. Only necessary if the non-centrality vector is to be calculated by the function (i.e. if delta is NULL).
cVal	Optional numeric vector giving the critical value, if specified the argument alpha is ignored.
corMat	An optional matrix giving the correlations of the contrasts specified in cMat.
twoSide	Logical indicating whether a two sided or a one sided test should be performed (defaults to one-sided)
control	A list of options for the pmvt and qmvt functions as produced by mvtnorm.control.

Value

The function returns the power value.

References

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

planMM, LP, sampSize, powerMM

Examples

```
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential=c(85),
               betaMod=matrix(c(0.33,2.31,1.39,1.39), byrow=TRUE, nrow=2))

# calculate optimal contrasts and critical value
plMM <- planMM(models, doses, 50, scal = 200, alpha = 0.05)
```

```
# calculate mean vectors
compMod <- fullMod(models, doses, base = 0, maxEff = 0.4, scal = 200)
muMat <- modelMeans(compMod, doses, FALSE, scal = 200)

# calculate power to detect mean vectors
# Power for linear model
powCalc(plMM$contMat, 50, mu = muMat[,1], sigma = 1, cVal = plMM$critVal)
# Power for emax model
powCalc(plMM$contMat, 50, mu = muMat[,2], sigma = 1, cVal = plMM$critVal)
# Power for logistic model
powCalc(plMM$contMat, 50, mu = muMat[,3], sigma = 1, cVal = plMM$critVal)
# compare with JBS 16, p. 650
```

---

powerMM *Calculate power for different sample sizes*

Description

Calculates the power under the assumed candidate set for different sample sizes.

Usage

```
powerMM(models, doses, base, maxEff, sigma, lower, upper, step,
        sumFct = c("min", "mean", "max"), off = 0.1 * max(doses),
        scal = 1.2 * max(doses), alpha = 0.025, twoSide = FALSE,
        control = mvtnorm.control(), muMat = NULL, alRatio = NULL,
        typeN = c("arm", "total"), ...)
```

Arguments

models	A list specifying the candidate models. This can also be a fullMod object, then the arguments base, maxEff, off and scal are ignored.
doses	Dose levels to be administered
base	Expected baseline effect
maxEff	Expected maximum change from baseline
sigma	Expected standard deviation
lower, upper	Maximum and minimum group sample size for which the power is calculated.
step	Stepsize for the sample size at which the power is calculated. It is calculated at seq(lower,upper,by=step).
sumFct	A character vector giving the names of the summary functions used to combine the power values into one value. By default the minimum, the mean and the maximum are used.
off	Offset parameter for the linear in log model (default 10 perc. of maximum dose).
scal	Scale parameter for the beta model (default 20 perc. larger than maximum dose).
alpha	Level of significance (default: 0.025)

twoSide	Logical indicating whether a two sided or a one-sided test should be performed. By default FALSE, so one-sided testing.
control	A list of options for the pmvt and qmvt functions as produced by mvtnorm.control.
muMat	An optional matrix with means in the columns, dimnames should be given (dose levels and names of contrasts), if specified the the models argument should not be specified, see examples below.
alRatio	Vector describing the relative patient allocations to the dose groups. See examples below, e.g. c(1,2,2) corresponds to allocating twice as many patients in dose groups two and three. Per default balanced allocations are assumed.
typeN	One of "arm" or "total". Determines, whether the sample size in the smallest arm or the total sample size is iterated in bisection search algorithm. See examples below.
...	Possible additional arguments for sumFct.

Details

Given the candidate set of models and associated guesstimates the function calculates the power to detect every model in the candidate set for different group sample sizes. Additionally summary functions can be specified to calculate the combined power (by default the minimum, mean and maximum). The location and scale parameters are determined by forcing the model function to go through (0,base) and (dmax,maxEff), see Pinheiro et al. (2006) for details. There exists a plot method for the output of the powerMM function. See the examples below.

Value

A powerMM object, i.e. a matrix containing the power values for different sample sizes and models

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bornkamp, B. and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

plot.powerMM, powCalc

Examples

```
## Not run:
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = 25,
               logistic = c(50, 10.88111), exponential= 85,
               betaMod=matrix(c(0.33,2.31,1.39,1.39), byrow=TRUE, nrow=2))
pM <- powerMM(models, doses, base = 0, maxEff = 0.4, sigma = 1,
            alpha = 0.05, lower = 10, upper = 100, step = 20, scal = 200)
```

```
pM
# a graphical display provides plot method
plot(pM)
# reproduces plot in JBS 16, p.651
plot(pM, line.at = 0.8, models = "none")


# the same with fullMod object and default alpha
fMod <- fullMod(models, doses, base = 0, maxEff = 0.4, scal=200)
pM <- powerMM(fMod, sigma = 1, lower = 10, upper = 100,
              step = 20, scal = 200)
pM

# using unbalanced (but fixed) allocations
pM <- powerMM(models, doses, base = 0, maxEff = 0.4, sigma = 1,
               lower = 10, upper = 100, step = 20, scal = 200,
               alRatio = c(3, 2, 2, 1, 1, 1), typeN = "arm")
plot(pM, summ = "mean")

# example, where means are directly specified
# doses
dvec <- c(0, 10, 50, 100)
# mean vectors
mu1 <- c(1, 2, 2, 2)
mu2 <- c(1, 1, 2, 2)
mu3 <- c(1, 1, 1, 2)
mMat <- cbind(mu1, mu2, mu3)
dimnames(mMat)[[1]] <- dvec
pM <- powerMM(muMat = mMat, doses = dvec, sigma = 2, lower = 10,
              upper = 100, step = 20)
pM

## End(Not run)
```

`quadratic` *Quadratic model*

Description

The model function for the quadratic model is defined as

$$f(d, \theta) = E_0 + \beta_1 d + \beta_2 d^2$$

Usage

```
quadratic(dose, e0, b1, b2)
```

Arguments

dose	Dose variable
e0	Placebo effect
b1	beta1 parameter
b2	beta2 parameter (controls, whether model is convex or concave)

Details

This model is intended to capture a possible non-monotonic dose-response relationship.

Value

Response value

References

Pinheiro, J. C., Bretz, F. and Branson, M. (2006). Analysis of dose-response studies - modeling approaches, *in* N. Ting (ed.). *Dose Finding in Drug Development*, Springer, New York, pp. 146–171

See Also

logistic, sigEmax, linlog, linear, exponential, emax, betaMod

---

sampSize *Sample size calculations for MCPMod*

Description

Given a candidate set, the baseline effect, the maximum effect and the standard deviation, the sampSize function returns the smallest sample size achieving a certain combined power value. See Pinheiro et al. (2006) for details.

Usage

```
sampSize(models, doses, base, maxEff, sigma, upperN,
         lowerN = floor(upperN/2), power = 0.8, alRatio = NULL,
         sumFct = mean, off = 0.1*max(doses), scal = 1.2 * max(doses),
         alpha = 0.025, twoSide = FALSE, tol = 0.001, verbose = FALSE,
         control = mvtnorm.control(), muMat = NULL,
         typeN = c("arm", "total"), ...)
```

Arguments

models	A list specifying the candidate models. This can also be a fullMod object, then the arguments base, maxEff, off and scal are ignored
doses	Dose levels to be administered
base	Expected baseline effect
maxEff	Expected maximum change from baseline
sigma	Expected standard deviation
upperN, lowerN	Upper and lower bound for the target sample size. lowerN defaults to floor(upperN/2).
power	Desired combined power value, defaults to 0.8.
alRatio	Vector describing the relative patient allocations to the dose groups. See Examples below.
sumFct	A function to combine the power values under the different models into one value. By default the arthmetic mean is used.
off	Offset parameter for the linear in log model (default 10 perc. of maximum dose).
scal	Scale parameter for the beta model (default 20 perc. larger than maximum dose).
alpha	Level of significance (default: 0.025)
twoSide	Logical indicating whether a two sided or a one-sided test is performed. By default FALSE, so one-sided testing.
tol	A positive numeric value specifying the tolerance level for the bisection search algorithm.
verbose	Logical value indicating if a trace of the iteration progress of the bisection search algorithm should be displayed.
control	A list of options for the pmvt and qmvt functions as produced by mvtnorm.control
muMat	An optional matrix with means as columns and given dimnames (dose levels and names of contrasts). If specified the the models argument should not be specified, see examples below.
typeN	One of "arm" or "total". Determines, whether the sample size in the smallest arm or the total sample size is iterated in bisection search algorithm. See examples below.
...	Possible additional arguments for sumFct

Details

Calculates the sample size necessary to achieve a desired combined power value for the multiple contrast test. A summary function is used to combine the individual power values. The allocation ratios for the dose groups need to be predefined and fixed (by default balanced allocations are assumed).

The function implements a simple bisection search algorithm to determine the target sample size. In case the upper and lower bound (upperN, lowerN) do not contain the target sample size the algorithm automatically adjusts these boundaries, but outputs a warning message.

Value

An object of class sampSize, with the following components:

`samp.size`	Vector of target sample size(s)
`approx.power`	Combined Power achieved under the assumed scenario and sample size.

References

Bornkamp B., Pinheiro J. C., and Bretz, F. (2009). MCPMod: An R Package for the Design and Analysis of Dose-Finding Studies, *Journal of Statistical Software*, **29**(7), 1–23

Pinheiro, J. C., Bornkamp, B., and Bretz, F. (2006). Design and analysis of dose finding studies combining multiple comparisons and modeling procedures, *Journal of Biopharmaceutical Statistics*, **16**, 639–656

See Also

powCalc, powerMM

Examples

```
## Not run:
# example from JBS paper p.651
doses <- c(0,10,25,50,100,150)
models <- list(linear = NULL, emax = c(25),
               logistic = c(50, 10.88111), exponential=c(85),
               betaMod=matrix(c(0.33,2.31,1.39,1.39), byrow=TRUE, nrow=2))
sampSize(models, doses, base = 0, maxEff = 0.4, sigma = 1,
         upperN = 80, scal = 200, alpha = 0.05)
# with different summary function

sampSize(models, doses, base = 0, maxEff = 0.4, sigma = 1,
         upperN = 90, scal = 200, sumFct = median, alpha = 0.05)

# with unbalanced allocations (twice as many patients in placebo group
# than in active dose groups)
sampSize(models, doses, base = 0, maxEff = 0.4, sigma = 1,
       alpha = 0.05, upperN = 80, scal = 200, alRatio=c(2,1,1,1,1,1))
# iterates total sample size instead of sample size in smallest arm
# in this case no big difference
sampSize(models, doses, base = 0, maxEff = 0.4, sigma = 1,
         alpha = 0.05, upperN = 500, scal = 200, typeN = "total",
         alRatio=c(2,1,1,1,1,1))


# sample size calculation for general matrix of means
dvec <- c(0, 10, 50, 100)
mu1 <- c(1, 2, 2, 2)
mu2 <- c(1, 1, 2, 2)
mu3 <- c(1, 1, 1, 2)
mMat <- cbind(mu1, mu2, mu3)
dimnames(mMat)[[1]] <- dvec
```

```
sampSize(muMat = mMat, doses = dvec, sigma = 1,
       alpha = 0.05, upperN = 10, alRatio=c(2,2,1,1))

## End(Not run)
```

`sigEmax` *Sigmoid Emax Model*

Description

The model function for the sigmoid Emax model is defined as

$$f(d, \theta) = E_0 + E_{max} \frac{d^h}{ED_{50}^h + d^h}$$

Usage

```
sigEmax(dose, e0, eMax, ed50, h)
```

Arguments

`dose`	Dose variable
`e0`	Placebo effect
`eMax`	Asymptotic maximum change from placebo effect
`ed50`	Dose giving half of the asymptotic maximum effect
`h`	Hill parameter, determining the steepness of the model at the ED50

Details

The sigmoid Emax model is an extension of the (hyperbolic) Emax model by introducing an additional parameter h, that determines the steepness of the curve at the ed50 value. The sigmoid Emax model describes monotonic, sigmoid dose-response relationships.

Value

Response value

References

MacDougall, J. (2006). Analysis of dose-response studies - Emax model,*in* N. Ting (ed.), *Dose Finding in Drug Development*, Springer, New York, pp. 127–145

See Also

`emax`, `logistic`, `betaMod`, `linlog`, `linear`, `quadratic`, `exponential`

Index