Package ‘clinDR’

April 12, 2021

Version 2.3.5
Date 2021-04-11
Title Simulation and Analysis Tools for Clinical Dose Response Modeling
Maintainer Neal Thomas <snthomas99@gmail.com>
Depends R (>= 3.5.0), rstan (>= 2.17.3), shiny, waiter
Imports foreach, graphics, ggplot2, DoseFinding, stats, utils, parallel, doParallel, magrittr, purrr, tibble, dplyr, tidyr, glue
License GPL (>= 2)
NeedsCompilation no
Author Neal Thomas [aut, cre] (<https://orcid.org/0000-0002-1915-8487>), Jing Wu [aut], Mike K. Smith [aut]
Repository CRAN
Date/Publication 2021-04-12 05:40:03 UTC

R topics documented:

- clinDR-package .......................................................... 3
- "Extract.emaxsim" ....................................................... 4
- "Extract.emaxsimB" .................................................... 6
- checkMonoEmax ........................................................ 8
- coefEmax ............................................................. 10
- compileStanModels ................................................. 12
### R topics documented:

- `DRDensityPlot` .................................................. 12
- `emaxalt` .......................................................... 14
- `emaxfun` .......................................................... 16
- `emaxPrior.control` ............................................... 18
- `emaxsim` .......................................................... 20
- `emaxsimB` ........................................................ 24
- `emaxsolve` ........................................................ 29
- `fitEmax` .......................................................... 30
- `fitEmaxB` .......................................................... 32
- `FixedMean` ...................................................... 35
- `mcmc.control` .................................................... 36
- `metaData` ......................................................... 37
- `nllogis` .......................................................... 39
- `plot.emaxsim` ..................................................... 41
- `plot.emaxsimB` ................................................... 42
- `plot.emaxsimBobj` ............................................... 43
- `plot.fitEmax` ..................................................... 47
- `plot.fitEmaxB` ................................................... 50
- `plot.plotB` ....................................................... 52
- `plotB` ............................................................. 54
- `plotBdensity` ..................................................... 58
- `plotD` ............................................................. 60
- `predict.emaxalt` ................................................ 61
- `predict.emaxsim` ............................................... 63
- `predict.emaxsimB` .............................................. 64
- `predict.emaxsimBobj` ......................................... 66
- `predict.emaxsimobj` ........................................... 67
- `predict.fitEmax` ............................................... 69
- `predict.fitEmaxB` ............................................... 71
- `print.emaxsim` .................................................. 72
- `print.emaxsimB` ............................................... 74
- `print.emaxsimBobj` ........................................... 75
- `print.emaxsimobj` ............................................. 76
- `print.fitEmax` .................................................. 77
- `print.fitEmaxB` ................................................ 78
- `prior.control` ................................................... 78
- `RandEmax` ....................................................... 80
- `runShiny` ......................................................... 81
- `SeEmax` .......................................................... 82
- `selEstan` ......................................................... 84
- `showStanModels` ................................................. 85
- `sigmaEmax` ...................................................... 86
- `startEmax` ....................................................... 87
- `summary.emaxsim` .............................................. 88
- `summary.emaxsimB` ............................................ 90
- `summary.emaxsimBobj` ....................................... 91
- `summary.emaxsimobj` ......................................... 93
Bayesian and maximum likelihood Emax model fitting, graphics and simulation for clinical dose response.

Description

The functions `fitEmax` and `fitEmaxB` fit an Emax model to binary or continuous data using maximum likelihood or Bayesian estimation. They have several generic supporting functions. Functions to produce plots associated with dose response analyses are (`plotD`, `plotB`, `plot.fitEmax`, `plot.fitEmaxB`). The functions `emaxsim` and `emaxsimB` perform simulations of 4- and 3-parameter Emax ML or Bayesian estimation. The ML estimates are replaced with alternative model fits when the primary estimation fails. Several supporting functions are supplied to analyze the output of `emaxsim` and `emaxsimB`, including analyses for specific simulated data sets. All of the data sets from dose response meta analyses are included in `metaData`.

Details

The function `compileStanModels` must be executed once after the package is installed to create compiled STAN Emax models before the Bayes functions in the package can be executed. This requires 3-10 minutes to complete on most machines. The compiled code is 32-bit or 64-bit specific, and both must be created if both versions of R are used.

The Bayesian computations use the R package `rstan`. It can be installed from CRAN. Windows users should check the instructions for `rstan` at the [https://mc-stan.org](https://mc-stan.org) and [https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started](https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started). Note that `Rtools` must be installed, which is a simple, but often overlooked step. Instructions for its installation are given in the second URL.

Author(s)

Neal Thomas [aut, cre], Jing Wu[aut]

See Also

`DoseFinding`
Extract a simulation from the output of emaxsim

Description

Extract a simulated data set from the output of emaxsim. Data are re-created using the stored random number seed.

Usage

```r
## S3 method for class 'emaxsim'
x[i, ...]
```

Arguments

- `x`: Output object from `emaxsim`
- `i`: Simulation replication to extract
- `...`: Parameters passed to other functions (none currently)

Details

Re-creates the ith simulated data set for subsequent analyses. Also returns all analyses done for the ith data set in `emaxsim`

Value

A list is returned with class(emaxsimobj) containing:

- `y`: Response vector
- `dose`: Doses corresponding to `y`
- `pop`: Population parameters; type of parameter depends on constructor function generating study data.
- `popSD`: Vector containing the population SD used to generate continuous data. NULL for binary data.
- `init`: Starting Emax parameters
- `est4`: 4-paramemeter Emax fit (ed50,lambda,emax,e0). NA if failed to converge or 3-parameter model requested.
- `est3`: 3-paramemeter Emax fit (ed50,emax,e0). NA if failed to converge or 4-parameter model successfully fit.
- `estA`: Alternative parameter estimates. NA if Emax model fit successfully
- `vc`: The variance-covariance matrix of the model parameters for the selected model.
- `residSD`: The residual SD based on the selected model.
- `bigC`: bigC = TRUE if the primary fit (from modType) yielded an ED50 > ED50 upper limit.
negC = TRUE if the primary fit (from modType) yielded a negative ED50 estimate < ED50 lower limit

modType
When modType=4, the fitting begins with the 4 parameter model. If estimation fails or modType=3, the 3-parameter estimation is applied. If it fails, a best-fitting model linear in its parameters is selected.

fit
Output of model determined by fitType

fitType
Character vector with "4", "3", "L", "LL", or "E" for 4-Emax, 3-Emax, linear, log-linear, or exponential when an alternative model is selected.

ed50cutoff
Upper allowed limit for ED50 estimates.

ed50lowcutoff
Lower allowed limit for the ED50 estimates.

switchMod
If switchMod is TRUE, the algorithm substitutes a simpler model if (1) convergence is not achieved, (2) the information matrix is not positive definite at the converged values, (3) the ED50 estimates are outside the cutoff bounds. If switchMod is F, only conditions (1) or (2) cause a simpler model to be used.

PL
T if the 'plinear' algorithm in nls converged

predpop
Population means for each dose group

dm
Vector containing dose group means

dsd
Vector containing dose group SDs

fitpred
Dose groups means estimated from the model

sepred
SEs for estimates in fitpred

sedif
SEs for model-based estimates of difference with placebo

pVal, selContrast
P-value and contrast selected from MCP-MOD test

idmax
Index of default dose group for comparison to placebo

Note
Extraction from a simulation object requires re-creation of the simulated data set. If the extracted object is to be used more than once, it is more efficient to save the extracted object than reuse [].

Author(s)
Neal Thomas

See Also
emaxsim, print.emaxsimobj, plot.emaxsimobj, update.emaxsimobj

Examples

```r
## Not run:
## code change random number seed

nsim<-50
```
### population parameters for simulation

```r
e0 <- 2.465375
ed50 <- 67.481113
dtarget <- 100
diftarget <- 9.032497
edmax <- solveEmax(diftarget, dtarget, log(ed50), 1, e0)
sdy <- 7.967897
pop <- c(log(ed50), edmax, e0)
meanlev <- emaxfun(doselev, pop)
```

### FixedMean is specialized constructor function for emaxsim

```r
gen.parm <- FixedMean(n, doselev, meanlev, sdy)
```

```r
D1 <- emaxsim(nsim, gen.parm, modType = 3)
e49 <- D1[49]  # extract 49th simulation
```

## End (Not run)

---

**Extract.emaxsimB**

Extract a simulation from the output of emaxsimB

### Description

Extract a simulated data set from the output of emaxsimB. Data are re-created using the stored random number seed.

### Usage

```r
# S3 method for class 'emaxsimB'
x[i, ...]
```

### Arguments

- `x`: Output object from emaxsimB
- `i`: Simulation replication to extract
- `...`: Parameters passed to other functions (none currently)

### Details

Re-creates the ith simulated data set for subsequent analyses. Also returns all analyses done for the ith data set in emaxsimB
Value

A list is returned with class(emaxsimBobj) containing:

- **y**  
  Response vector

- **dose**  
  Doses corresponding to y

- **pop**  
  Population parameters; type of parameter depends on constructor function generating study data.

- **popSD**  
  Vector containing the population SD used to generate continuous data. NULL for binary data.

- **binary**  
  When TRUE, binary data modeled on the logit scale

- **modType**  
  modType=3,4, for the hyperbolic and sigmoidal Emax models.

- **predpop**  
  Population means for each dose group

- **dm**  
  Vector containing dose group means

- **dsd**  
  Vector containing dose group SDs

- **fitpred**  
  Posterior means of the dose groups means

- **sepred**  
  SE (posterior SD) corresponding to the estimates in fitpred

- **sedif**  
  SE (posterior SD) for the differences with placebo

- **bfit**  
  Bayesian fitted model of class fitEmaxB.

- **prior, mcmc**  
  See fitEmax for documentation.

- **pVal, selContrast**  
  P-value and contrast selected from MCP-MOD test

- **idmax**  
  Index of default dose group for comparison to placebo

Note

Extraction from a simulation object requires re-creation of the simulated data set. If the extracted object is to be used more than once, it is more efficient to save the extracted object than reuse [].

Author(s)

Neal Thomas

See Also

emaxsimB, print.emaxsimBobj, plot.emaxsimBobj

Examples

```r
## Not run:

save.seed<-.Random.seed
set.seed(12357)

nsim<-50
```
### checkMonoEmax

**Bayes posterior predictive test for Emax (monotone) model fit**

**Description**

Bayes posterior predictive test for an Emax (monotone) model fit comparing the best response from lower doses to the response from the highest dose.

**Usage**

```r
checkMonoEmax(y, dose, parm, sigma2, nvec=rep(1, length(dose)), xbase=NULL,
```

---

```r
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
eax<-solveEmax(diftarget, dtarget, log(ed50), 1, e0)
sdy<-7.967897
pop<-c(log(ed50), emax, e0)
meanlev<-emaxfun(doselev, pop)

### FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n, doselev, meanlev, sdy)
prior<-emaxPrior.control(epmu=0, epsca=30, difTargetmu=0, difTargetsc=30, dTarget=100, p50=50, sigmalow=0.1, sigmaup=30, parMD=5)
mcmc<-mcmc.control(chains=1, warmup=500, iter=5000, seed=53453, propInit=0.15, adapt_delta = 0.95)
D1 <- emaxsimB(nsim, gen, prior, modType=3, mcmc=mcmc, check=FALSE)
out<-D1[2]

.Random.seed<-save.seed

## End(Not run)
```
checkMonoEmax

```
modelFun=emaxfun,
trend='positive',
binary= FALSE,logit=binary)
```

**Arguments**

- **y**
  Outcomes. Continuous y can be individual data or group means. Binary y can be individual data, group proportions, or 0/1 data with corresponding counts, as is required by fitEmaxB.

- **dose**
  Doses corresponding to outcomes.

- **parm**
  Matrix of simulated parameter values (each row is a simulated parameter vector). The parm values must be constructed for use in the model function modFun. The default is a 4-parameter Emax model with parameters (log(ED50),lambda,Emax,E0). For a 3-parameter model, set lambda=1 for each simulated parameter vector.

- **sigma2**
  Simulated draws from the residual variance (assumed additive, homogeneous). The length of sigma2 must be the same as the number of rows of parm. sigma2 is ignored when binary=TRUE.

- **nvec**
  The number of observations contributing to each y. The default is 1 for patient-level data.

- **xbase**
  Optional covariates matching y. nvec must be 1 (patient-level) data. The coefficients for xbase are the final columns of parm.

- **modelFun**
  The mean model function. The first argument is a scalar dose, and the second argument is a matrix of parameter values. The rows of the matrix are random draws of parameter vectors for the model. The default function is the 4-parameter Emax function emaxfun.

- **trend**
  The default is 'positive', so high values for lower doses yield small Bayesian predictive probabilities. Set trend to 'negative' for dose response curves with negative trends.

- **binary**
  If TRUE, the inverse logit transform is applied to the (Emax) function output for comparison to dose group sample proportions, and the predictive data are sampled from a binomial distribution.

- **logit**
  logit is deprecated, use binary

**Details**

A sample of parameters from the joint posterior distribution must be supplied (typically produced by an MCMC program). The Bayesian predictive p-value is the posterior probability that a dose group sample mean in a new study with the same sample sizes would yield a higher (or lower for negative trend) difference for one of the lower doses versus the highest dose than was actually obtained from the real sample. There must be at least two non-placebo dose groups (NA returned otherwise). Placebo response is excluded from the comparisons.

The function generates random numbers, so the random number generator/seed must be set before the function is called for exact reproducibility.

**Value**

Returns a scalar Bayesian predictive p-value.
 coefEmax

definition

Extract Emax model parameter estimates. MLE for fitEmax. Matrix of MCMC generated parameters for fitEmaxB.

Description

Extract Emax model parameter estimates. MLE for fitEmax. Matrix of MCMC generated parameters for fitEmaxB.

Usage

## S3 method for class 'fitEmax'
coef(object, ...)
## S3 method for class 'fitEmaxB'
coef(object, local=FALSE, ...)
## S3 method for class 'emaxsim'
coef(object, ...)
## S3 method for class 'emaxsimB'
coef(object, local=FALSE, ...)

Examples

## Not run:

data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
prior<-emaxPrior.control(epmu=0,epsca=10,difTargetmu=0,difTargetsca=10,dTarget=80.0,
p50=3.75,signalow=0.01,sigmaup=20)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
parms<-coef(fitout)[,1:4] #use first intercept
checkMonoEmax(y=exdat$rslt, dose=exdat$dose, parm=parms, sigma2=(sigma(fitout))^2,
nvec=exdat$sampsize, trend='negative')

## End(Not run)
Arguments

object          Output of Emax fitting function
local           When a prior distribution of type 'emaxPrior' was used to create the object, specifying local=TRUE will output the local 'difTarget' parameter estimates.
...             No additional inputs supported

Value

Vector of MLE estimates of model parameter from fitEmax. Matrix of MCMC generated parameters for fitEmaxB. Matrix with posterior median parameter estimates for each emaxsimB simulation: (led50,lambda,emax,e0) or (led50,emax,e0). For emaxsim, a list is returned with the model type fit for each simulation, and a matrix with the corresponding model coefficients. The order of the parameters is given in the emaxsim documentation.

Author(s)

Neal Thomas

See Also

sigma, fitEmax, fitEmaxB, emaxsim, emaxsimB

Examples

doselev<-c(0,5,25,50,100,350)
n<-c(78,81,81,81,77,80)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-8.0
pop<-c(log(ed50),emax,e0)
dose<-rep(doselev,n)
meanlev<-emaxfun(dose,pop)
y<-rnorm(sum(n),meanlev,sdy)

testout<-fitEmax(y,dose,modType=4)
coef(testout)
Description

Compile rstan Emax models after package clinDR is installed

Usage

compileStanModels()

Details

The compiled models are stored in the models sub-directory of the installed clinDR package. The user must have write-access to the package directory. The package can be installed in a user-specified directory if the user does not have write privileges for the default package directory. Execution requires several minutes. The compiled models are 32- or 64-bit specific. Both sets must be compiled if the compiled R type is changed (they are stored in sub-directories comp32 or comp64). It is recommended to execute the function again if the package rstan is updated.

Package rstan must be functional for CompileStanModels to be successful. See https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started. Note especially the instructions for installing Rtools, which is required for execution on a Windows machine.

Value

'basemodel.rds' and 'mrmodel.rds' should be created in the package directory in the sub-directory 'models'.

Author(s)

Neal Thomas

Description

Plot Bayes or confidence interval density contours over a grid of points (usually dose or time)

Density plot for distributions conditional on a variable. A grid of values are specified for the conditioning variable, which is plotted on the horizontal axis. The conditioning variable is typically dose or time
Usage

DRDensityPlot(x,qL,qH,qlevL=c(0.025,0.05,0.10,0.25),
xlim,ylim,xlab='x',ylab='y')

Arguments

x A grid of conditioning values to be plotted on the horizontal axis. This grid typically represents dose or time.

qL Lower percentiles, confidence or probability levels. qL is a matrix with rows corresponding to x, and columns corresponding to qlevL. The percentiles must be increasing in order and less that 0.50.

qH Upper percentiles, confidence or probability levels. qH levels correspond to the qL levels but are ordered from highest to lowest (1-qlevL), with the smallest greater than 0.50.

qlevL Density intervals are formed with percentile boundaries at (qlevL,1-qlevL). qlevL must be increasing between (0,0.5).

xlim Plot limits for the x-axis

ylim Plot limits for the y-axis

xlab x-axis label

ylab y-axis label

Details

The function takes as input percentiles defining confidence intervals or Bayesian probability intervals at different levels (e.g. 5,95, 25,75) for distributions conditional on a variable that is typically dose or time. Regions defined by different confidence/probability levels are represented by different levels of shading. The input parameter, qlevL, is used only to define the input in the matrices qL and qH. The qlevL is not used for any numerical calculations, which must be done before executing the function.

Value

Plotted output only.

Author(s)

Neal Thomas

See Also

plotBdensity
Examples

```r
# Not run:
data('metaData')
exdat<-metaData[metaData$taid==32,]

msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmax(exdat$rslt,exdat$dose,modType=3,count=exdat$sampsize,
    msSat=msSat)
dgrid<-seq(0,100,length=100)
seout95<-predict(fitout,dgrid,clev=0.95)
seout90<-predict(fitout,dgrid,clev=0.9)
seout80<-predict(fitout,dgrid,clev=0.8)
seout50<-predict(fitout,dgrid,clev=0.5)
qlev<-c(0.025,0.05,0.10,0.25)
qL<-cbind(seout95$ubdif,seout90$ubdif,seout80$ubdif,seout50$ubdif)
qH<-cbind(seout95$lbdif,seout90$lbdif,seout80$lbdif,seout50$lbdif)
DRDensityPlot(dgrid,qL,qH,qlevL=qlev,xlab='Dose',ylab='Diff with PBO')
```

## End(Not run)

### `emaxalt`

**Fit 4- or 3-parameter Emax model substituting simpler curves if convergence not achieved.**

**Description**

ML estimation for 4- and 3-parameter Emax model. If the 4-parameter model is requested, it is estimated and the 3-parameter model is fit only if the 4-parameter estimation fails. If 3-parameter estimation fails, the linear, log-linear, or exponential model producing the smallest residual SS is substituted. For binary data, the model is fit on the logit scale and then back-transformed.

**Usage**

```r
emaxalt(y, dose, modType=3,binary=FALSE,
    iparm=NA,ed50cutoff=2.5*max(doselev),
    ed50lowcutoff=doselev[2]/1000,switchMod= TRUE,
    truncLambda=6)
```

**Arguments**

- `y` Response vector
Doses corresponding to \( y \)

When modType=4, the fitting begins with the 4 parameter model. If estimation fails or modType=3, the 3-parameter estimation is applied. If it fails, a best-fitting model linear in its parameters is selected.

When specified, the Emax model is fit on the logit scale, and then the results are back-transformed to proportions.

Vector of optional initial values for the Emax fit. Starting values are computed if not specified.

Upper allowed limit for ED50 estimates.

Lower allowed limit for the ED50 estimates.

If switchMod is TRUE, the algorithm substitutes a simpler model if (1) convergence is not achieved, (2) the information matrix is not positive definite at the converged values, (3) the ED50 estimates are outside the cutoff bounds. If switchMod is F, only conditions (1) or (2) cause a simpler model to be used.

When modType=4 and the converged estimate of the Hill parameter lambda exceeds truncLambda, the model fit is judged unstable and discarded. Set truncLambda=Inf for no truncation.

The partial linear method is used in nls. If it fails, gauss-newton is attempted. If both methods fail, the next simpler model is attempted. For the 4-parameter model, the next step is the 3-parameter model. For the 3-parameter model, a linear, log-linear \( \log(\text{dose} + 1.0) \), and \( \exp(\text{dose} / \max(\text{dose})) \) are fit using lm, and the 2-parm fit with the smallest residual SS is selected.

A list assigned class "emxalt" with the following elements:

- \( dm \): Vector containing dose group means
- \( dsd \): Vector containing dose group SDs
- \( Sparm \): Vector of starting values for 3-parameter Emax fit.
- \( fitType \): Character vector with "4", "3", "L", "LL", or "E" for 4-Emax, 3-Emax, linear, log-linear, or exponential when an alternative model is selected.
- \( vc \): The variance-covariance matrix of the model parameters stored as a vector. The length is 16, 9, 4 depending on fitType.
- \( fitpred \): Dose groups means estimated from the model
- \( residSD \): The residual SD based on the selected model.
- \( sepred \): SEs for estimates in fitpred
- \( sedif \): SEs for model-based estimates of difference with placebo
- \( bigC \): \( \text{bigC}= \text{TRUE} \) if the primary fit (from modType) yielded an ED50 \( > \) ED50 upper limit.
- \( negC \): \( \text{negC}= \text{TRUE} \) if the primary fit (from modType) yielded an ED50 estimate \( < \) ED50 lower limit.
est4  4-parameter Emax fit (ed50,lambda,emax,e0). NA if failed to converge or 3-parameter model requested.
est3  3-parameter Emax fit (ed50,emax,e0). NA if failed to converge or 4-parameter model successfully fit.
estA  Alternative parameter estimates. NA if Emax model fit successfully

Author(s)
Neal Thomas

See Also
eemaxsim, nls

Examples

```r
save.seed<-.Random.seed
set.seed(12357)

doselev<-c(0.5,25,50,100)
n<-c(78,81,81,77)
dose<-.rep(doselev,n)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanresp<-emaxfun(dose,pop)
y<-rnorm(sum(n),meanresp,sdy)
simout<-emaxalt(y,dose)
simout2<-emaxalt(y,dose,modType=4)

#.Random.seed<save.seed
```

Emaxfun

Vectorized versions of the hyperbolic and sigmoidal Emax models

Description
Evaluate Emax models for a vector of dose levels for multiple sets of parameters.
**Usage**

`emaxfun(dose, parm)`

**Arguments**

- `dose` A vector (or scalar) of dose levels
- `parm` A vector or matrix with columns containing \( \log(\text{ed50}) \), Hill parameter if sigmoid model, \( \text{emax}, e0 \)

**Details**

The Hill parameter is omitted from `parm` for the hyperbolic model

**Value**

Returns a matrix of Emax function evaluations. The rows correspond to the parameter replications, and the columns correspond to the dose levels.

**Note**

The ordering of the parameters was selected to facilitate use of the 'plinear' algorithm in function `nls`.

**Author(s)**

Neal Thomas

**See Also**

`dlogis`

**Examples**

```r
doselev <- c(0.5, 25, 50, 100)
e0 <- 2.465375
ed50 <- 67.481113
dtarget <- 100
diftarget <- 9.032497
lambda <- 2
emax <- solveEmax(diftarget, dtarget, log(ed50), lambda, e0)

parm <- c(log(ed50), lambda, emax, e0)
plot(doselev, emaxfun(doselev, parm))
```
Set the parameters of the prior distribution for the Emax model implemented in fitEmaxB.

### Description
Set the parameters of the prior distribution for the Emax model implemented in fitEmaxB.

### Usage
```r
eemaxPrior.control(epmu=NULL, epsca=NULL, 
difTargetmu=NULL, difTargetsca=NULL, 
dTarget=NULL, p50=NULL, 
sigmalow=NULL, sigmaup=NULL, 
effDF=parmDF, parmDF=5, 
loged50mu=0.0, loged50sca=1.73, 
loglammu=0.0, loglamsca=0.425, parmCor=-0.45, 
basemu=NULL, basevar=NULL, binary=FALSE)
```

### Arguments
- **epmu**: Mean for E0 in a t-prior distribution. Logistic scale for binary data.
- **epsca**: The scale parameter for E0 in a t-prior distribution. Logistic scale for binary data.
- **difTargetmu**: Mean for the prior distribution of the effect at dose dTarget versus placebo. Logistic scale for binary data.
- **difTargetsca**: The scale parameter for the prior distribution of the effect at dose dTarget versus placebo. Logistic scale for binary data.
- **dTarget**: Target dose for prior effect. Typically the highest dose planned and/or the proof-of-concept dose.
- **p50**: Projected ED50. See references for its use in creating the prior distribution for the ED50.
- **sigmalow**: Lower bound for a uniform prior distribution for the residual SD (continuous data).
- **sigmaup**: Upper bound for a uniform prior distribution for the residual SD (continuous data).
- **effDF**: The degrees of freedom for the log-t prior distributions for the placebo and difTarget parameters. If a vector of length 2 is specified, the first value is the degrees of freedom for placebo and the second for difTarget.
- **parmDF**: The degrees of freedom of the bivariate log-t prior distribution for the ED50 and lambda parameters.
- **loged50mu**: Mean of prior t-distribution for the log(ED50). See references for its default value and interpretation.
- **loged50sca**: Scale (analogous to SD) of the prior t-distribution for the log(ED50).
loglammu  Mean of prior t-distribution for the Hill parameter lambda. See references for its default value and interpretation.

loglamsc  Scale (analogous to SD) of the prior t-distribution for the Hill parameter lambda.

dtype  Correlation for the bivariate log-t prior distribution for the 
       ED50 and lambda parameters.

basemu  A vector of prior means for the covariate regression parameters.

basevar  The prior variance-covariance matrix for the covariate regression parameters.
         The covariate regression parameters are apriori independent of the other dose
         response model parameters.

binary  Set to TRUE for binary data applications. Used to check for consistency in usage.

Details

The prior distribution is based on meta-analyses of dose response described in the references. The 
E0 and difTarget parameters have independent t-distribution prior distributions. For binary data, 
these parameters are computed on the logistic scale. The prior means and scales of these parameters 
must be assigned compound-specific values. The predicted ED50 at the study design stage must 
also be specified as 'P50'. For continuous data, the prior distribution for the residual SD is 
uniform on a user-specified scale.

The prior distribution of the log(ED50) has a t-distribution centered at log(P50), with scale, degrees 
of freedom (parmDF), and offset to the P50, defaulting to values given in the references (these 
can be changed, but they are difficult to interpret outside the context of the meta-analyses). If 
modType=4, the prior distribution for the Hill parameter is also t-distribution with parmDF degrees 
of freedom and corParm correlation with the log(ED50).

Value

List of class emaxPrior of prior parameter values for use in fitEmaxB.

Author(s)

Neal Thomas

References

<doi:10.1080/19466315.2014.924876>


<doi:10.1177/0962280216684528>

See Also

fitEmaxB
Simulate Emax maximum likelihood estimation

Description

Simulate dose response data and apply 4- or 3- parameter Emax MLE estimation. For binary data, the model is fit on the logit scale and then back-transformed. When MLE estimation fails, models with fewer parameters (including models linear in their parameters) are substituted. Summaries of estimation performance are returned for further analyses. An MCP-MOD test is also performed for each simulated data set.

Usage

```r
emaxsim(
  nsim,
  genObj,
  modType=3,
  binary=FALSE,
  seed=12357,
  nproc = parallel::detectCores(),
  negEmax=FALSE,
  ed50contr=NULL,
  lambdacontr=NULL,
  testMods=NULL,
  idmax=length(doselev),
  iparm=NA,
  ed50cutoff=2.5*max(doselev),
  ed50lowcutoff=doselev[2]/1000,
  switchMod= TRUE,
  truncLambda=6,
  description=""
)
```

Arguments

- `nsim`: Number of simulation replications
- `genObj`: Object containing inputs and function to create simulated data sets. These objects are created by special constructor functions; the current choices are `FixedMean` and `RandEmax`.
- `modType`: When modType=4, the fitting begins with the 4 parameter model. If estimation fails or modType=3, the 3-parameter estimation is applied. If it fails, a best-fitting model linear in its parameters is selected.
- `binary`: When specified, the Emax model is fit on the logit scale, and then the results are back-transformed to proportions.
- `seed`: Seed for random number generator used to create data.
nproc The number of processors to use in parallel computation of the simulations, which are divided into equal-sized computational blocks. When nproc=1 a single local processor.

negEmax When TRUE, the intended effect is assumed to be negative.

ed50contr A vector of ED50 values for creating a global null test using the MCP-MOD package DoseFinding based on Emax model-based contrasts. The default is 3 contrasts: the mid-point between pbo and the lowest dose, the mid-point between the 2 highest doses, and the median of the dose levels. When there are <=4 doses including pbo, the median-based contrast is excluded.

lambdacontr Hill parameters matched to the ed50contr. The default value is 1 for each contrast model.

testMods The model object for a MCP-MOD test created by Mods from package DoseFinding. If specified, the other contrast inputs are ignored. The Mods call should use the unique sorted dose levels. The direction of the trend should be specified in the call to Mods. The negEmax is stored for use by support functions, but it does not determine the direction of the effect when testMods is specified. The validity of testMods is not checked.

idmax Index of the default dose group for comparison to placebo. Most analysis functions allow other dose groups to be specified. The default is the index of the highest dose.

iparm Starting values for the Emax fit. If unspecified, starting values are computed. The order of the variables is (log(ED50),Emax,E0) or (log(ED50),lambda,Emax,E0). Note the transformation of ED50.

ed50cutoff The upper limit for the ED50 parameter estimates. The default is large enough to ensure a near linear fit to the data from an Emax model.

ed50lowcutoff Lower allowed limit for the ED50 estimates.

switchMod If switchMod is TRUE, the algorithm substitutes a simpler model if (1) convergence is not achieved, (2) the information matrix is not positive definite at the converged values, (3) the ED50 estimates are outside the cutoff bounds. If switchMod is F, only conditions (1) or (2) cause a simpler model to be used.

truncLambda When modType=4 and the converged estimate of the Hill parameter lambda exceeds truncLambda, the model fit is judged unstable and discarded. Set truncLambda=Inf for no truncation. Four parameter model fits are also discarded when lambda is less than 0.1.

description Optional text describing the simulation setting that is stored with the simulation output.

Details Continuous data can be simulated from any dose response curve with homogeneous normally distributed residuals. The estimation procedure starts with ML estimation of a 4- or 3- parameter Emax model depending on modType. If modType=3 or 4-parameter estimation fails, a 3 parameter Emax model is fit by maximum likelihood non-linear least squares. If 1) nls fails to converge for a 3 parameter Emax model, 2) the ED50 estimate is <=0, or 3) the ED50 estimate exceeds ed50cutoff, a linear, log-linear (offset of 1.0), or scaled exponential (exp(dose/max(dose))), is fit using simple linear least squares estimation. The model selected has the smallest residual SS.
Binary data are handled similarly using maximum likelihood implemented with the nlm function. The models are fit on the logit scale and then back-transformed for estimation of dose response. Reduced linear models are selected based on the corresponding likelihood deviance.

MCP-MOD tests are created from contrasts based on the Emax function using the DoseFinding package. Different ED50 and lambda (Hill) parameters can be specified to form the contrasts. A contrast matrix output from the DoseFinding package can be specified instead, allowing for other contrast choices.

Value

A list is returned with class(emaxsim) containing:

- **description**: User description of simulation
- **binary**: Binary response data.
- **modType**: User supplied starting Emax model
- **genObj**: List object with data and function used to generate study data
- **pop**: Matrix with rows containing population parameters for each simulation. Type of parameter depends on constructor function generating study data.
- **popSD**: Vector containing the population SD used to generate continuous data. NULL for binary data.
- **init**: Matrix with rows containing the starting Emax parameters for each simulation
- **est4**: Matrix with 4 parameter Emax fit. NA if failed to converge or modType=3
- **est3**: Matrix with 3 parameter Emax fit. NA if failed to converge or 4-parameter estimation was successful.
- **estA**: Matrix with alternative parameter estimates. NA if Emax model fit successfully
- **vc**: Variance-covariance matrix for the estimated parameters stored as a vector for each simulation. The vc vector stored has 16, 9, or 4 elements depending on fitType (with NA values on the end if elements are unused).
- **residSD**: The residual SD based on the selected model.
- **fitType**: Character vector with "4", "3", "L", "LL", or "E" for 4-Emax, 3-Emax, linear, log-linear, or exponential when an alternative model is selected.
- **pVal**: The nsim p-values from the global null test. The p-values are 1-sided computed using MCP-Mod.
- **selContrast**: The index of the test contrast producing the smallest p-value.
- **testMods**: Object of class Mods from R package DoseFinding that defines the contrasts used in MCP-MOD testing. The functions can be plotted with DoseFinding loaded.
- **negEmax**: User input stored for subsequent reference.
- **ed50cutoff**: Upper allowed limit for ED50 estimates
- **ed50lowcutoff**: Lower allowed limit for the ED50 estimates.
- **switchMod**: If switchMod is TRUE, the algorithm substitutes a simpler model if (1) convergence is not achieved, (2) the information matrix is not positive definite at the converged values, (3) the ED50 estimates are outside the cutoff bounds. If switchMod is F, only conditions (1) or (2) cause a simpler model to be used.
negC = TRUE if the primary fit (from modType) yielded an ED50 estimate < ED50 lower limit.

bigC = TRUE if the primary fit (from modType) yielded an ED50 > ED50 upper limit.

depop = Matrix with population means for each dose group

mv = Matrix with rows containing dose group sample means

sdv = Matrix with rows containing dose group sample SD

fitpredv = Matrix with rows containing dose groups means estimated from the model

sepredv = Matrix with rows containing SE for fitpredv

sedifv = Matrix with rows containing SE for model-based differences with placebo

rseed = Starting random number seed for each simulated data set that can be assigned to .Random.seed. To reproduce the data, the random number generator must also be changed to RNGkind("L'Ecuyer-CMRG").

idmax = Index of default dose group for comparison to placebo (e.g., for plotting Z-statistics).

Author(s)
Neal Thomas

See Also
print.emaxsim, summary.emaxsim, plot.emaxsim, coef.emaxsim, sigma.emaxsim, vcov.emaxsim, predict.emaxsim, emaxfun

Examples

```r
## Not run:
## emaxsim changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)
```
### FixedMean is specialized constructor function for emaxsim

```r
gen<-FixedMean(n,doselev,meanlev,sdy)
```

```r
D1 <- emaxsim(nsim,gen,modType=3)
summary(D1,testalph=0.05)
```

```r
D4 <- emaxsim(nsim,gen,modType=4)
summary(D4,testalph=0.05)
```

## End(Not run)

---

**emaxsimB**

*Simulate Emax Bayesian estimation*

**Description**

Simulate dose response data and apply 4- or 3- parameter sigmoidal or hyperbolic Bayesian estimation. The prior distribution is input by the user with default values for some parameters based on the empirical distribution estimated from dose response meta-analyses. For binary response data, the Emax model is fit on the logit scale, and then back-transformed.

**Usage**

```r
eemaxsimB(nsim, genObj, prior, modType = 4, binary = FALSE, seed=12357,
check = FALSE, nproc=parallel::detectCores(),
negEmax = FALSE, ed50contr = NULL,
lambdacontr = NULL, testMods = NULL,
idmax = length(doselev),
mcmc = mcmc.control(),
customCode=NULL, customParms=NULL,
description = "")
```

**Arguments**

- `nsim` Number of simulation replications
- `genObj` Object containing inputs and function to create simulated data sets. These objects are created by special constructor functions; the current choices are `FixedMean` and `RandEmax`.
- `prior` Prior specification through an object of type `emaxPrior` or `prior`. See `emaxPrior.control` and `prior.control` for details. The `emaxPrior` specifies the magnitude of the potential effect for a specified dose (typically the highest anticipated dose and/or the dose in a POC study), while the `prior` specifies the theoretical maximum effect (the emax parameter). The `prior` specification is deprecated and will be removed.
modType

When modType=3, a hyperbolic Emax model is fit. When modType=4, a sigmoid Emax model is fit.

binary

When specified, the Emax model is fit on the logit scale, and then the results are back-transformed to proportions.

seed

Seed for random number generator used to create data. A separate seed can be passed to rstan through the MCMC object.

check

When TRUE, a single simulated data set is created and the data and rstan object are returned for convergence checking. The data are in the form needed for developing customCode. Note that customCode is not called when check=TRUE.

nproc

The number of processors to use in parallel computation of the simulations, which are divided into equal-sized computational blocks. When nproc=1 a single local processor.

negEmax

When TRUE, the intended effect is assumed to be negative.

ed50contr

A vector of ED50 values for creating a global null test using the MCP-MOD package DoseFinding based on Emax model-based contrasts. The default is 3 contrasts: the mid-point between pbo and the lowest dose, the mid-point between the 2 highest doses, and the median of the dose levels. When there are <=4 doses including pbo, the median-based contrast is excluded.

lambdacontr

Hill parameters matched to the ed50contr. The default value is 1 for each contrast model.

testMods

The model object for a MCP-MOD test created by Mods from package DoseFinding. If specified, the other contrast inputs are ignored. The Mods call should use the unique sorted dose levels. The direction of the trend should be specified in the call to Mods. The negEmax is stored for use by support functions, but it does not determine the direction of the effect when testMods is specified. The validity of testMods is not checked.

idmax

Index of the default dose group for comparison to placebo. Most analysis functions allow other dose groups to be specified. The default is the index of the highest dose.

mcmc

MCMC settings created using mcmc.control

customCode

An optional user supplied function that computes custom estimates/decision criteria from each simulated data set and its Bayesian model fit. The output are stored in a list, customOut, of length nsim. See the Details section below for a description of the mandatory inputs to the customCode function.

customParms

Optional parameters that can be passed to customCode.

description

Optional text describing the simulation setting that is stored with the simulation output.

Details

The Bayesian model fits are implemented in rstan using function fitEmaxB. The function compileStanModels must be executed once to create compiled STAN code before emaxsimB can be used.

Continuous data can be simulated from any dose response curve with homogeneous normally distributed residuals.
Binary data are handled similarly. The models are fit on the logit scale and then back-transformed for estimation of dose response. Reduced linear models are selected based on the corresponding likelihood deviance.

MCP-MOD tests are created from contrasts based on the Emax function using the DoseFinding package. Different ED50 and lambda (Hill) parameters can be specified to form the contrasts. A contrast matrix output from the DoseFinding package can be specified instead, allowing for other contrast choices.

Customized code:
For binary data, the inputs to the function customCode for each simulated data set will be (parms,pVal,dose,y), where parms is the matrix of parameters generated from the posterior distribution with columns in the order given in function emaxfun, pVal is the MCP-MOD p-value, dose and y are the patient-level simulated data. For continuous data, the inputs are (parms,residSD,pVal,dose,y, where residSD are the variance parameters generated from their posterior distribution. The customParms supply other user-inputs such as a target efficacy level. When it is not null, the customCode inputs must be (parms,pVal,dose,y,customParms) or (parms,residSD,pVal,dose,y,customParms).

Value
A list is returned with class(emaxsim) containing:

- **description** User description of simulation
- **localParm** TRUE when the prior prior distribution is input using emaxPrior.
- **binary** Binary response data.
- **modType** Type of Emax model fit (3 or 4 parameters)
- **genObj** List object with data and function used to generate study data
- **pop** Matrix with rows containing population parameters for each simulation. Type of parameter depends on constructor function generating study data.
- **popSD** Vector containing the population SD used to generate continuous data. NULL for binary data.
- **mcmc** mcmc input settings
- **prior** Input prior distribution.
- **est** Matrix with posterior median parameter estimates for each simulation: (led50,lambda,emax,e0,difTarget) or (led50,emax,e0,difTarget). The difTarget are omitted for the deprecated distribution.
- **estlb,estub** Array with lower posterior (0.025,0.05,0.1) and upper posterior (0.975,0.95,0.9) percentiles of the model parameters. The array ordering is model parameters, simulation, and percentile.
- **residSD** The posterior median of the residual SD for each simulation.
- **pVal** The nsim p-values from the global null test. The p-values are 1-sided computed using MCP-Mod.
- **selContrast** The index of the test contrast producing the smallest p-value.
- **testMods** Object of class Mods from R package DoseFinding that defines the contrasts used in MCP-MOD testing. The functions can be plotted with DoseFinding loaded.
Goodness of fit test computed by checkMonoEmax.

User input stored for subsequent reference.

Matrix with population means for each dose group

Matrix with rows containing dose group sample means

Matrix with rows containing dose group sample SD

Pooled within-dose group sample variance

Matrix with rows containing dose groups means estimated by the posterior medians of the MCMC generated values.

Matrix with rows containing SE (posterior SD) associated with fitpredv

Matrix with rows containing dose groups mean differences with placebo estimated by the posterior medians of the differences of the MCMC generated values.

Matrix with rows containing SE (posterior SD) for the differences with placebo

Array with lower posterior (0.025, 0.05, 0.1) and upper posterior (0.975, 0.95, 0.9) percentiles of differences between dose group means and placebo. The array ordering is dose group minus placebo, simulation, and percentile.

The proportion of divergent MCMC iterations from each simulated analysis.

Starting random number seed for each simulated data set set that can be assigned to .Random.seed. To reproduce the data, the random number generator must also be changed to RNGkind("L’Ecuyer-CMRG").

Index of default dose group for comparison to placebo (e.g., for plotting Z-statistics).

List with customized output. It will be NULL if customCOde is not specified.

Note

The default modType was changed from 3 to 4 for clinDR version >2.0

Author(s)

Neal Thomas

References


See Also

print.emaxsimB, summary.emaxsimB, plot.emaxsimB, coef.emaxsimB, sigma.emaxsimB, emaxfun
Examples

```r
## Not run:

### emaxsimB changes the random number seed

nsim<-50
idmax<-5
doselev<-c(0.5,25,50,100)
n<-c(78,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsc=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,
propInit=0.15,adapt_delta = 0.95)

### custom code to compute the distribution of the dose yielding
### a target diff with pbo

customCode<-function(parms,residSD,pVal,dose,y,customParms){
  target<-customParms
ed50<-exp(parms[,1])
  emax<-parms[,2]
  td<-ifelse(emax-target>0,ed50*(target/(emax-target)),Inf)
  tdest<-median(td)
  lb<-quantile(td,0.1)
  ub<-quantile(td,0.9)
  return(c(td=tdest,lb=lb,ub=ub))
}

D1 <- emaxsimB(nsim,gen, prior, modType=4,seed=12357,mcmc=mcmc,check=FALSE,
customCode=customCode,customParms=1.0)
D1

## End(Not run)
```
Solve Emax function for target value

Description

Solve the Emax function for dose or Emax to yield a specified response.

Usage

solveEmax(target, dose, led50, lambda, e0, pboadj=TRUE)
solveDose(target, led50, lambda, emax, e0, pboadj=TRUE)

Arguments

target The targetted response. If the Emax model is specified on the logit scale for binary data, target and e0 must be logit transformed also.
dose The dose yielding target. It is specified for solveEmax, and returned for solveDose
led50, lambda, e0 Emax model parameters (ed50 log transformed)
emax The Emax model parameter for solveDose. The value returned for solveEmax
pboadj When TRUE, target is placebo-adjusted.

Author(s)

Neal Thomas

See Also

fitEmax, fitEmaxB, emaxsim, emaxsimB

Examples

e0<-10
dose<-1
led50<-log(0.5)
lambda<-2
target<- -1.5
emax<-solveEmax(target, dose, led50, lambda, e0)
emax

dose1<-solveDose(target, led50, lambda, emax, e0)
dose1
emaxfun(dose=dose1, parm=c(led50, lambda, emax, e0)) - e0
**fitEmax**

*ML fit of hyperbolic or sigmoidal Emax models to continuous/binary dose response data.*

**Description**

Calls Newton-Raphson optimizers, nls and nlm, for a hyperbolic or sigmoidal Emax model. Different intercepts for multiple protocol-data are supported. For binary data, the Emax model is on the logit scale.

**Usage**

```r
fitEmax(y, dose, iparm, xparm, modType=4,
      prot=rep(1,length(y)), count=rep(1,length(y)), xbase=NULL,
      binary=FALSE, diagnostics=TRUE, msSat=NULL,
      pboAdj=rep(FALSE, max(prot)), optObj=TRUE)
```

**Arguments**

- `y`  
  Outcome for each patient. Missing `Y` values are not permitted. Dose/protocol group means for grouped continuous data. For binary data, `y` must be 0/1 and counts must be supplied for each 0/1 value.

- `dose`  
  Dose for each patient.

- `iparm`  
  Optional starting values for the Newton-Raphson algorithm. The order of the variables is (log(ED50),Emax,E0) or (log(ED50),lambda,Emax,E0). Note the transformation of ED50. If there is more than one protocol, the E0 is automatically duplicated.

- `xparm`  
  Optional starting values for the baseline covariate slopes (if any). `xparm` must be specified when `iparm` and `xbase` are specified. `startEmax` is used to obtain starting values if no starting values are specified.

- `modType`  
  modType=3 (default) for the 3-parameter hyperbolic Emax model. modType=4 for the 4-parameter sigmoidal Emax model.

- `prot`  
  Protocol (group) membership used to create multiple intercepts. The default is a single protocol.

- `count`  
  Counts for the number of patients when the `Y` are dose continuous group means or binary 0/1 values. Default is 1 (ungrouped data).

- `xbase`  
  A matrix of baseline covariates with rows corresponding to `y` that enter as linear additive predictors. The baseline covariates must be centered about their (protocol-specific) means. `xbase` does not include an intercept or protocol indicators. Covariates cannot be specified with PBO adjusted or aggregated input.

- `diagnostics`  
  Print trace information per iteration and any error messages from the optimizing methods. Printing can be suppressed for use in simulation studies.

- `binary`  
  When `TRUE`, the `y` are assumed to be coded 0/1, and the means reported are proportions. The Emax model is specified on the logit scale, and proportions are estimated from the model by back-transformation.
msSat  If continuous $Y$ are dose/protocol group means rather than individual measurements, the within group variance, $msSat$, should be supplied. This variance is the mean square from the model saturated in dose and protocol. It is used for goodness-of-fit (GOF) testing, and to improve the residual variance estimate for the Emax model. If it is not supplied, statistics needed for GOF will not be available, and the residual SD (and associated SE) will have low degrees of freedom.

pboAdj  For published data with only pbo-adjusted dose group means and SEs, the model is fit without an intercept(s). If initial parameters are supplied, the intercept ($E_0$) should be assigned $0$. A zero for the placebo mean should not be included in $Y$. This option is not available for binary data. Potential correlation between between placebo-adjusted means is ignored.

optObj  Include the output object from the R optimization code in the fitEmax output.

Details

Fits the 3- or 4- Emax model using \texttt{nls}. A newton-raphson algorithm is tried first followed by a partial linear optimization if needed. Binary data are fit using \texttt{nlm}.

Value

A list assigned class "fitEmax" with:

\begin{itemize}
  \item \texttt{fit}  The parameter estimates and their variance-covariance matrix.
  \item \texttt{y, dose, modType, prot, count, binary, pboAdj}  Input values.
  \item \texttt{gofTest}  Goodness of fit p-value based on likelihood ratio comparison of the model to a saturated fit.
  \item \texttt{nll}  $-2\log$likelihood for the Emax model and the saturated model. Residual sums of squares are returned for continuous data models. These statistics can be used to construct other tests using multiple calls to fitEmax (e.g., 3 vs 4 parameter Emax models, or a common intercept model across protocols).
  \item \texttt{df}  Residual degrees of freedom for the Emax model and the saturated model.
  \item \texttt{optobj}  When requested, the fit object returned by the R optimization functions.
\end{itemize}

Author(s)

Neal Thomas

See Also

\texttt{nls, nlm, nllogis, predict.fitEmax, plot.fitEmax, coef.fitEmax}

Examples

```r
## the example changes the random number seed

doselev<-c(0,5,25,50,100,350)
```
n<-c(78,81,81,81,77,80)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)

sdy<-8.0
pop<-c(log(ed50),emax,e0)
dose<-rep(doselev,n)
meanlev<-emaxfun(dose,pop)
y<-rnorm(sum(n),meanlev,sdy)
testout<-fitEmax(y,dose,modType=4)

---

**fitEmaxB**

Bayesian fit of hyperbolic or sigmoidal Emax models to continuous/binary dose response data.

---

**Description**

Uses Rpackage *rstan* to fit a Bayesian hyperbolic or sigmoidal Emax model. Different intercepts for multiple protocol-data are supported. For binary data, the Emax model is on the logit scale.

**Usage**

```r
fitEmaxB(y, dose, prior, modType = 4, prot = rep(1, length(y)),
          count = rep(1, length(y)), xbase=NULL,
          binary = FALSE, msSat = NULL,
          pboAdj = FALSE, mcmc = mcmc.control(), estan = NULL,
          diagnostics = FALSE, nproc = getOption("mc.cores", 1L))
```

**Arguments**

- **y** Outcome for each patient. Missing Y values are are not permitted. Dose/protocol group means for grouped continuous data. For binary data, y must be 0/1 and counts must be supplied for each 0/1 value.
- **dose** Dose for each patient.
- **prior** Prior specification through an object of type 'emaxPrior' or 'prior'. See `emaxPrior.control` and `prior.control` for details. The 'emaxPrior' specifies the magnitude of the potential effect for a specified dose (typically the highest anticipated dose and/or the dose in a POC study), while the 'prior' specifies the theoretical maximum effect (the emax parameter). The 'prior' specification is deprecated and will be removed.
modType  modType=3 (default) for the 3-parameter hyperbolic Emax model. modType=4 for the 4-parameter sigmoidal Emax model.

prot  Protocol (group) membership used to create multiple intercepts. The default is a single protocol. The prior distribution for the placebo response is re-used independently for each intercept.

count  Counts for the number of patients when the Y are dose continuous group means or binary 0/1 values. Default is 1 (ungrouped data).

xbase  A matrix of baseline covariates with rows corresponding to y that enter as linear additive predictors. The baseline covariates must be centered about their (protocol-specific) means. xbase does not include an intercept or protocol indicators. Covariates cannot be specified with PBO adjusted or aggregated input.

binary  When TRUE, the y are assumed to be coded 0/1, and the means reported are proportions. The Emax model is specified on the logit scale, and proportions are estimated from the model by back-transformation.

msSat  If continuous Y are dose/protocol group means rather than individual measurements, the within group variance, msSat, should be supplied. This variance is the mean square from the model saturated in dose and protocol. It is used to improve the residual variance estimate for the Emax model. If it is not supplied, the residual SD (and associated SE) will have low degrees of freedom.

pboAdj  For published data with only pbo-adjusted dose group means and SEs, the model is fit without an intercept(s). If initial parameters are supplied, the intercept (E0) should be assigned 0. A zero for the placebo mean should not be included in y. This option is not available for binary data. Potential correlation between between placebo-adjusted means is ignored.

mcmc  Inputs controlling rstan execution. See mcmc.control for details.

estan  The compiled rstan Emax model is usually loaded automatically. It can be load to an object using the function selEstan and passed to fitEmaxB for repeated executions to improve efficiency and stability.

diagnostics  Printed output from rstan. See Details for more information.

nproc  The number of processor requested for STAN MCMC computations. Defaults to the value set by the rstan installation. When set explicitly, nproc is usually 1 or the number of MCMC chains. If greater than the number of chains, it is set to the number of chains.

Details

The function compileStanModels must be executed once to create compiled STAN code before fitEmaxB can be used.

MCMC fit of a Bayesian hyperbolic or sigmoidal Emax model. The prior distributions available are based on the publication Thomas, Sweeney, and Somayaji (2014) and Thomas and Roy (2016).

The posterior distributions are complex because the distributions of the Emax and ED50 parameters change substantially as a function of the lambda, often creating 'funnel' type conditions. Small numbers of divergences are common and do not appear easily avoided. Extensive simulation using evaluations with emaxsimB support the utility of the resulting approximate posterior distributions. The number of divergences can be viewed using diagnostics=TRUE. The usual convergence diagnostics should always be checked.
A list assigned class "fitEmaxB" with:

- `estanfit`: The rstan object with the model fit.
- `y, dose, prot, count, modType, binary, pboAdj, nbase, msSat, prior, mcmc`: Input values.

**Note**

The default `modType` was changed from 3 to 4 for clinDR version >2.0

**Author(s)**

Neal Thomas

**References**


**See Also**

`fitEmax`, `predict.fitEmaxB`, `plot.fitEmaxB`, `coef.fitEmaxB`

**Examples**

```r
## Not run:
data("metaData")
exdat<-(metaData[metaData$taid==1,]
prior<-emaxPrior.control(epmu=0,epsca=4,difTargetmu=0,difTargetsca=4,dTarget=20,
p50=(2+5)/2,
sigmalow=0.01,sigmaup=3)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,prot=exdat$protid,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
plot(fitout)
## End(Not run)
```
FixedMean

**Fixed means (proportions) random data constructor for emaxsim for continuous or binary data**

**Description**

Creates a list object that contains inputs and a function to create simulated data sets with a common mean (proportion) for use in emaxsim with normal or continuous data.

**Usage**

```r
FixedMean(n, doselev, meanlev, resSD, parm = NULL, binary=FALSE)
```

**Arguments**

- `n`: Sample size for each dose group
- `doselev`: Dose levels (including 0 for placebo) in the study corresponding to `n`. Must be in increasing order.
- `meanlev`: Mean response at each doselev. For binary data, these are the proportion of responders (no logit transformation).
- `resSD`: Standard deviation for residuals within each dose group (assumed common to all dose groups)
- `parm`: Population parameters that are saved for later reference, but are not used when creating simulated data. `parm` can contain parameters for a 3- or 4-parameter Emax model that generated `meanlev`. They should be stored in the order given in `emaxfun`. Default is `NULL`.
- `binary`: Normal data with homogeneous variance are generated unless `binary` is `TRUE`, and then means are interpreted as proportions and 0/1 data are generated.

**Value**

A list of length 2. The first element is itself a list named `genP` that contains named elements `n`, `resSD`, `doselev`, `dose`, `parm`, `binary`, and the element `meanlev`, which is specific to `FixedMean`. The second element is a function named `genFun` that takes `genP` as input and returns a list with named elements `meanlev`, `parm`, `resSD`, `y`.

**Author(s)**

Neal Thomas

**See Also**

`emaxsim`, `RandEmax`
Examples

```r
## Not run:
## example changes the random number seed

doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113

dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)

meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
genp<-FixedMean(n,doselev,meanlev,sdy,pop)

### binary example
n<-rep(500,5)
doselev<-c(0,5,25,50,1000)
dose<-rep(doselev,n)
e0<- qlogis(0.2)
ed50<-20
diftarget<-qlogis(0.6)-qlogis(0.2)
lambda<-2
dtarget<-100
emax<-solveEmax(diftarget,dtarget,log(ed50),lambda,e0)
pop<-c(log(ed50),lambda,emax,e0)
meanlev<-plogis(emaxfun(doselev,pop))
genp<-FixedMean(n,doselev,meanlev,sdy,pop,binary=TRUE)
tapply(genp$genFun(genp$genP)$y,dose,mean)

## End(Not run)
```

mcmc.control

Settings for restan execution in function fitEmaxB
metaData

Description

Set MCMC controls. Also control spread of initial parameter values.

Usage

mcmc.control(chains = 1, thin = 1,
warmup = 1000, iter = 3333* thin+warmup,
propInit = 0.25, seed = 12357, adapt_delta = 0.95)

Arguments

chains Number of chains
thin Number of discarded sampled parameter values. warmup and iter include thin, so for example, to output 1000 samples, iter must be 1000 times thin.
warmup See rstan documentation for function sampling.
iter See rstan documentation for function sampling.
propInit Initial values for $E_0$ and $E_{\text{max}}$ are derived from the prior mean plus/minus propInit times the prior SD. propInit can be set to a small proportion if very diffuse prior distributions are specified.
seed Seed passed to rstan.
adapt_delta See rstan documentation for function sampling.

Note

Some defaults were changed with version>=2.0. For earlier versions, warmup = 500, iter = 5000* thin, and adapt_delta=0.8

---

metaData  Dose response data from several published meta-analyses

Description

Dose response data from over 200 compounds included in published meta-analyses. The data are aggregated in a single data frame in a common format.

Usage

data('metaData')
Format

The data frame has one row for each compound, protocol within compound, and dose group within protocol. Compound and protocol level descriptors are repeated on each row of the data frame.

drugid A numerical ID identifying each drug
taid A drug can be studied in more than one therapeutic area. The taid ID identifies each TA/drug combination.
protid Numerical (1,2,3,...) ID for protocols specific to each TAIID.
gname Generic drug name
bname Branded(USA) drug name
drugtype Drug classified as SMALL MOLECULE, BIOLOGIC, OTHER
route Route of administration, e.g., oral, subcutaneous,...
routeshort Abbreviated format for route
oralform Formulation (e.g., TABLET, POWDER,...) for drugs with oral administration.
fdaapproved NA if status was not yet determined
metasource Meta-analysis contributing compounds. BIO14: biological compounds through 2014;
protno Sponsor assigned protocol name/number
nctno Clinical trial.gov protocol ID
protyear When available, year of first patient/first visit. In some cases, date of journal publication design PARELLEL, CROSSOVER,...
actcomp Indicator if an active comparator was included in the protocol
etype etype=1 for the designated primary endpoint. For completeness, where there was ambiguity in the selection of the endpoint, additional endpoint data was included on separate rows and indicated by etype=2,3,... Most analyses subset on etype=1
poptype For a compound and TA, there can be distinctly different populations with anticipated response differences, e.g., treatment-naive and pre-treated patients. The population with the most studied doses has poptype=1. For completeness, additional populations are included and identified by poptype=2,3,... Most analyses subset on poptype=1
primsource IRO/PRO investigator/patient reported outcome; L lab, V vitals
primtype Primary endpoint is BINARY, CONTINOUS, TIMETOEVEN
primtime Time units to primary endpoint from randomization
timeunit DAY, HR, MIN, MONTH, WK for primary endpoint
indication Disease description
broadta Broad TA classification of the indication
endpointLong, endpointshort Endpoint name and an abbreviated form using for example, cfb and pcfb for change from baseline and percent change from baseline
dose Total daily dose for small molecules, total weekly dose for biologics in mg or mg/kg for weight-based dosing.
Amount of any loading dose

Number of visits with a loading dose

Dosing frequency

primregimen = 1 for most doses/regimens, but primregimen = 2 for a few regimens that clearly differed from the most common regimen for the same total dose. Most analyses subset on primregimen = 1

rslt The sample dose group mean (continuous) or proportion (binary) of the primary endpoint. Analyses of the time-to-event endpoints was compound specific (either a mean or a proportion was estimated).

se Standard error of rslt

sd Dose group sample standard deviation for continuous data

lcl, ucl, alpha alpha-level interval (lcl, ucl) when confidence intervals were extracted from the original data source because se were not reported

sampsize Sample size reported for rslt. The handling of missing data by the protocol sponsors varied, but ‘completers’ was most common.

ittsize The number randomized. The counts are usually available, except for internal data before 2009, where it was not collected.

pmiss Percent of missing data.

Details

Compound sampling plans and other details are given in the publications:


Examples

data('metaData')
names(metaData)

---

**nllogis**

The negative log likelihood function for a 3- or 4- parameter Emax model on the logit scale for binary dose response.

Description

The negative log likelihood function evaluated with a single input set of parameters for the binary Emax model on the logistic scale. For use with function fitEmax
nllogis

Usage

nllogis(parms,y,dose,
prot=rep(1,length(y)),
count=rep(1,length(y)),
xbase=NULL)

Arguments

parms Emax model parameter values. The order of the variables is (log(ED50),Emax,E0) or (log(ED50),lambda,Emax,E0). There must be an E0 for each protocol. Note the transformation of ED50.
y Binary outcome variable for each patient. Missing values are deleted. Must be coded 0/1.
dose Dose for each patient
prot Protocol (group) membership used to create multiple intercepts. The default is a single protocol. The value of prot must be 1,2,3...
count Counts for the number of patients with each dose/y value. Default is 1 (un-grouped data).
xbase Optional matrix of baseline covariates that enter the model linearly. If there is a single covariate, it should be converted to a matrix with one column.

Details

The negative log likelihood for the 3- or 4- Emax model on the logit scale for binary data. Note the ordering of the parameters and their transformations. A 3 vs 4 parameter model is determined by the length of parms.

Value

Negative log likelihood value is returned.

Author(s)

Neal Thomas

See Also

nlm, fitEmax

Examples

data('metaData')
exdat<-metaData[metaData$taid==8,]

cy<-round(exdat$sampsize*exdat$rs1t)
y<-c(rep(1,length(cy)),rep(0,length(cy)))
cy<-c(cy,exdat$sampsize-cy)
drep<-c(exdat$dose,exdat$dose)
plot.emaxsim

plotD(exdat$rslt, exdat$dose, se=FALSE)
nllogis(parms=c(log(2.5),-3.26,-0.15), y, drep,count=cy)

plot.emaxsim  Plot the output of emaxsim

Description

A Q-Q plot of the dose response estimate of the mean at a specified dose minus the population value divided by the standard error of the estimator (computed using the delta method). Estimates based on alternative models when the Emax estimation fails are highlighted in red.

Usage

## S3 method for class 'emaxsim'
plot(x, id = x$idmax, plotDif = TRUE, ...)

Arguments

x  Output of emaxsim
id  Index of the dose to be assessed (placebo index=1).
plotDif  If true (default), the estimates and population values are differences with placebo. IF false, absolute dose response values are used.
...  Optional parameters passed to the plotting function

Value

No output is returned.

Author(s)
Neal Thomas

See Also

emaxsim, print.emaxsim, summary.emaxsim

Examples

## Not run:
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
\[ \begin{align*}
e0&=-2.465375 \\
ed50&=-67.481113 \\
dtarget&=-100 \\
diftarget&=-9.032497 \\
ema&=\text{solveEmax}(\text{diftarget, dtarget, log(}ed50\text{), 1, e0}) \\
sdy&=-7.967897 \\
pop.parm&=c(\text{log(}ed50\text{), e0}, \text{e0}) \\
meanlev&=\text{emaxfun}(\text{doselev, pop.parm}) \\
###\text{FixedMean is specialized constructor function for emaxsim} \\
gen.parm&=\text{FixedMean}(n, \text{doselev, meanlev, sdy}) \\
D1 &\leftarrow \text{emaxsim}(\text{nsim, gen.parm}) \\
\text{plot}(D1, id=3) \\
## \text{End(Not run)}
\end{align*} \]

---

**plot.emaxsimB**

**Plot the output of emaxsimB**

**Description**

A Q-Q plot of the posterior mean of the mean dose response at a specified dose minus the population value divided by the posterior SD of the mean difference.

**Usage**

```r
## S3 method for class 'emaxsimB'
plot(x, id = x$idmax, plotDif = TRUE, ...)
```

**Arguments**

- **x** Output of `emaxsimB`
- **id** Index of the dose to be assessed (placebo index=1).
- **plotDif** If true (default), the estimates and population values are differences with placebo. IF false, absolute dose response values are used.
- **...** Optional parameters passed to the plotting function

**Value**

ggplot object is returned

**Author(s)**

Neal Thomas
See Also

`emaxsimB`, `print.emaxsimB`, `summary.emaxsimB`

Examples

```r
## Not run:
## emaxsimB changes the random number seeds
nsim<-50
iddmax<5
doselev<-c(0.5,25,50,100)
n<-c(78,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

### FixedMean is specialized constructor function for emaxsim
gen<FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)
D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)

plot(D1,id=3)
## End(Not run)
```

---

**plot.emaxsimBobj**

*Plot dose response from a data set generated by emaxsimB*

**Description**

Plot of population dose response curve, sample dose group means, posterior and posterior predictive intervals, and the model-based estimated (posterior means) dose response curve.
Usage

```r
Usage
## S3 method for class 'emaxsimBobj'
plot(
  x, clev=0.9, plotDif=FALSE,
  plotPop=c('m','3','4'),
  logScale=FALSE, plotResid=FALSE,
  plot=TRUE, ... )
```

Arguments

- `x`: Extracted data object from `emaxsimB`
- `clev`: Level for posterior intervals
- `plotDif`: When TRUE, the difference with placebo is plotted.
- `plotPop`: When plotPop='m', the mean values at each dose in the designs are joined using linear interpolation. Otherwise, the the population Emax parameters must be supplied with the data generator (see `FixedMean` or `RandEmax`). If the Emax parameters are not available, linear interpolation is used.
- `logScale`: Not implemented
- `plotResid`: Not implemented
- `plot`: Return plotting output without plotting.
- `...`: Other plot parameters. See `plot.fitEmaxB` for details

Note

The estimated curve is the posterior mean evaluated along a grid of dose values.

Examples

```r
Examples
## Not run:
## emaxsimB changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
```
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)

nsim<-50
idmax<-5
doselev<-c(0.5,25,50,100)
n<-c(78,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsc=30,dTarget=100,p50=50,sigmaLow=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)

D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)
plot(D1,id=3)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)
D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)
plot(D1[2])

## End(Not run)

---

plot.emaxsimobj

Plot dose response from a data set generated by emaxsim

---

Description

Plot of population dose response curve, dose group means with CIs, predictive intervals, and the model-based estimated dose response curve.
Usage

## S3 method for class 'emaxsimobj'
plot(
  x, xlim, xat=NULL, ylim, xlab, ylab,
  plotDif=FALSE,
  plotResid=FALSE,
  clev = 0.9,
  plotPop=c('m','3','4'),
  negC = FALSE,
  logScale=FALSE,
  predict=TRUE,
  plot=TRUE, ...)

Arguments

x Extracted data object from emaxsim
xlim x-axis limits
xat The points at which tick-marks are to be drawn. Errors occur if the points are outside the range of xlim. By default (when NULL) tickmark locations are computed.
ylim y-axis limits
xlab x-axis label
ylab y-axis label
plotDif When TRUE, the difference with placebo is plotted.
plotResid When TRUE, residuals (dose group means) are plotted.
clev Level for confidence intervals
plotPop Plot population dose response curve when plotPop='m' using linear interpolation between population means, when PlotPop='3' or '4', using the population Emax parameters that must be supplied with the data generator (see FixedMean or RandEmax). If the Emax parameters are not available, linear interpolation is used.
negC If the ED50<lower ED50 limit, TRUE causes the Emax model to be plotted in addition to the alternative model selected.
logScale If TRUE, log scale is used for dose.
predict When TRUE, predictive intervals are plotted with grey errorbars in addition to the confidence intervals.
plot Return plotting output without plotting.
...
Other plot parameters (not used).

Value
ggplot object is returned
plot.fitEmax

Author(s)
Neal Thomas

See Also
emaxsim, print.emaxsimobj, summary.emaxsimobj, update.emaxsimobj

Examples

## Not run:
## emaxsim changes the random number seed

nsim<-50
idmax<-5
doselev<-c(0.5,25,50,100)
n<-c(78,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen.parm)
e49<-D1[49]
plot(e49,clev=0.8)
## End(Not run)

plot.fitEmax

Plot a Emax model and dose group means.

Description
Plot an Emax model stored in an object created by function fitEmax.
Usage

## S3 method for class 'fitEmax'
plot(
  x, int = 0, plotResid = FALSE, clev = 0.9,
  predict = TRUE, plotci = TRUE, plotDif = FALSE,
  xlab = 'Dose',
  ylab = ifelse(plotResid, 'Residuals', ifelse(plotDif,
      'Difference With Placebo', 'Response')),
  symbol = NULL, symbolLabel = 'Group', symbolShape = 8,
  symbolColor = 'red', symbolSize = 4,
  bwidth = NULL,
  xlim = NULL,
  xat = NULL,
  ylim = NULL,
  logScale = FALSE,
  ngrid = 200,
  plot = TRUE, 
  ...
)

Arguments

x Output of fitEmax with class "fitEmax".

int The index for the protocol (intercept) to use for the predictions and computation of dose group means and standard errors. The default value is 0, which displays all protocols in a grid layout.

plotResid If TRUE, a residual plot of the observed dose group means is produced instead of a dose response curve plot.

clev Confidence level for intervals about the estimated mean for each dose.

predict When predict=TRUE, predictive intervals for sample dose group means are plotted. They are gray-shaded bars. If there is >1 symbol group mean for a protocol/dose combination, then the smaller sample size is used when computing the prediction interval.

plotci When plotCI=TRUE, confidence intervals for the population dose group means are plotted. They are black bars.

plotDif Plot difference between doses and placebo. It is assumed the lowest dose in each protocol is placebo.

xlab Label for the x-axis

ylab Label for the y-axis

symbol An optional grouping variable. The values of symbol must correspond to the original data used in fitEmax.

symbolLabel Label given to symbol in plot legend.

symbolShape A character vector with named elements giving the shapes assigned to different levels of variable symbol. If a single shape is specified, it is replicated for all dose group means. See package ggplot2 for symbol mappings.
symbolColor  
A character vector with named elements giving the colors assigned to different levels of variable symbol. If a single color is specified, it is replicated for all dose group means. See package ggplot2 for color mappings.

symbolSize  
The size of the symbol for the dose group sample means. Set symbolSize=0 to suppress plotting the means.

bwidth  
Width of the cap on the predictive interval bars.

xlim  
Plot limits for the x-axis

xat  
The points at which tick-marks are to be drawn. Errors occur if the points are outside the range of xlim. By default (when NULL) tickmark locations are computed.

ylim  
Plot limits for the y-axis

logScale  
If TRUE, log scale is used for dose.

ngrid  
The number doses evaluated when plotting the curve.

plot  
Return plotting output without plotting.

...  
No additional plotting options are currently used.

Details

Model estimates, standard errors, and confidence bounds are computed using function SeEmax.

The function generates random numbers when predict=TRUE, so the random number generator/seed must be set before the function is called for exact reproducibility.

Value

A list with ggplot object, and a matrix with the confidence and prediction interval limits.

Author(s)

Neal Thomas

See Also

nls

Examples

### example changes the random number seed

doselev<-c(0,5,25,50,100,350)
n<-c(78,81,81,81,77,80)

e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
```r
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-8.0
pop.parm<-c(log(ed50),emax,e0)
dose<-rep(doselev,n)
meanlev<-emaxfun(dose,pop.parm)
y<-rnorm(sum(n),meanlev,sdy)
testout<-fitEmax(y,dose,modType=4)
plot(testout)
```

---

**plot.fitEmaxB**

Plot a Emax model and dose group means.

**Description**

Plot an Emax model stored in an object created by function `fitEmaxB`.

**Usage**

```r
## S3 method for class 'fitEmaxB'
plot(
  x, int=0, plotResid=FALSE, clev=0.9, predict=TRUE, plotci=TRUE, plotDif=FALSE,
  xlab="Dose", ylab=ifelse(plotResid,"Residuals",ifelse(plotDif,
    'Difference With Placebo','Response')),
  symbol=NULL, symbolLabel="Group", symbolShape=8,
  symbolColor="red", symbolSize=4,
  bwidth=NULL, xlim=NULL, xat=NULL, ylim=NULL, ylim=NULL,
  logScale=FALSE, ngrid=200,
  plot=TRUE, ...
)
```

**Arguments**

- `x` Output of `fitEmaxB` with class "fitEmaxB".
- `int` The index for the protocol (intercept) to use for the predictions and computation of dose group means/proportions. The default value is 0, which displays all protocols in a grid layout.
- `plotResid` If TRUE, a residual plot of the observed dose group means/proportions less the model-based MCMC median estimates of the means/proportions.
plot.fitEmaxB

`clev` Level for posterior probability intervals about the mean/proportion for each dose.

`predict` When `predict=TRUE`, predictive intervals for sample dose group means/proportions are plotted. They are gray-shaded bars. If there is >1 symbol group mean/proportion for a protocol/dose combination, then the smaller sample size is used when computing the prediction interval.

`plotci` When `plotCI=TRUE`, posterior intervals for the population dose group means/proportions are plotted. They are black bars.

`plotDif` Plot difference between doses and placebo. It is assumed the lowest dose in each protocol is placebo.

`xlab` Label for the x-axis

`ylab` Label for the y-axis

`symbol` An optional grouping variable. The values of symbol must correspond to the original data used in `fitEmax`.

`symbolLabel` Label given to symbol in plot legend.

`symbolShape` A character vector with named elements giving the shapes assigned to different levels of variable `symbol`. If a single shape is specified, it is replicated for all dose group means/proportions. See package `ggplot2` for symbol mappings.

`symbolColor` A character vector with named elements giving the colors assigned to different levels of variable `symbol`. If a single color is specified, it is replicated for all dose group means/proportions. See package `ggplot2` for color mappings.

`symbolSize` The size of the symbol for the dose group sample means. Set `symbolSize=0` to suppress plotting the means.

`bwidth` Width of the cap on the predictive interval bars.

`xlim` Plot limits for the x-axis.

`xat` The points at which tick-marks are to be drawn. Errors occur if the points are outside the range of `xlim`. By default (when NULL) tickmark locations are computed.

`ylim` Plot limits for the y-axis.

`logScale` If `TRUE`, log scale is used for dose.

`ngrid` The number doses evaluated when plotting the curve.

`plot` Return plotting output without plotting.

`...` No additional plotting options are currently used.

**Details**

Model-based medians, standard deviations, and interval bounds for the dose groups means/proportions based on the MCMC parameters evaluated in the Emax function.

The function generates random numbers when `predict=TRUE`, so the random number generator/seed must be set before the function is called for exact reproducibility.

If baseline covaritates were included in the fit, then the mean of the predictions for the protocol given by `int` is plotted. This can be computationally intensive when the dosing grid is dense, the MCMC sample size is large, and the input sample size is large. Consider reducing `ngrid` in this situation. Note that the protocol must be specified, or the prediction defaults to patients from the first protocol.
Value

A list with ggplot object, and posterior and prediction interval limits.

Author(s)

Neal Thomas

See Also

fitEmaxB

Examples

## Not run:

data("metaData")
exdat<-metaData[metaData$taid==1,]
prior<-emaxPrior.control(epmu=0,epsca=4,difTargetmu=0,difTargetsca=4,dTarget=20,
p50=(2+5)/2,
sigmaLow=0.01,sigmaUp=3)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
f不出<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,prot=exdat$protid,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
plot(f不出)
## End(Not run)

plot.plotB

Plot Bayes dose response curve and dose group means

Description

Plot a dose response curve fit by Bayes MCMC methods (with optional posterior interval bars). Also plot dose group means (with optional CI bars)

Usage

## S3 method for class 'plotB'
plot( x,
plotDif= FALSE, plotMed= FALSE,
plotResid=FALSE, predict= TRUE,
logScale=FALSE, xlim,
xat=NULL,
...)

plot.plotB
plot.plotB

ylim, xlab,
ylab,labac='Act Comp',shapeac=8,colac='red',
symbolLabel='Group',symbolShape=8,
symbolColor='red',symbolSize=4, ...)

Arguments

x plotB object output from function plotB.
plotDif Plot difference between doses and placebo. It is assumed the lowest dose is placebo. If activeControl, the difference is with the active control mean, and the active controls are not plotted.
plotMed If TRUE, model-based curves are medians rather than means.
plotResid If TRUE, a plot of the residuals formed from the dose group means minus the posterior dose group means.
predict When predict=TRUE, predictive intervals for sample dose group proportions are plotted. They are gray-shaded bars.
logScale If TRUE, log scale is used for dose.
xlim x-axis limits
xat The points at which tick-marks are to be drawn. Errors occur if the points are outside the range of xlim. By default (when NULL) tickmark locations are computed.
ylim y-axis limits
xlab x-axis label
ylab y-axis label
labac x-axis label for the active control group.
shapeac Shape of the symbol for the active control group.
colac Color of the symbol for the active control group.
symbolLabel Label given to symbol in plot legend.
symbolShape A character vector with names giving the shapes assigned to different levels of variable symbol. If a single shape is specified, it is replicated for all dose groups. See package ggplot2 for symbol mappings.
symbolColor A character vector with names giving the colors assigned to different levels of variable symbol. If a single color is specified, it is replicated for all dose groups. See package ggplot2 for color mappings.
symbolSize The size of the symbol for the dose group sample means. Set symbolSize=0 to supress plotting.
... Additional parameters (not used)

Details

Produce additional plots from output of plotB without any re-computing. A plot is produced by default on return from the function. When active control is specified, the plot is 'printed' within the function. If there is a symbol group variable, it must be specified when plotB is executed. The symbol label, shape, color, and size must be re-specified in subsequent plot requests.
Value

ggplot object of the dose response curve, which will be plotted by default unless the output of the plot is assigned. When an active control group is present, the value returned is an invisible list with the ggplot for the dosing data, and a second ggplot for the ac data.

Note

PlotB can also be used with draws from a prior distribution to evaluate the prior dose response curve.

Author(s)

Neal Thomas

See Also

plotB, plotD, plot.fitEmax

Examples

```r
## Not run:
data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
prior<-emaxPrior.control(epmu=0,epsca=100,difTargetmu=0,difTargetsca=100,dTarget=80.0,
p50=3.75,sigmalow=0.01,sigmaup=20)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
parms<-coef(fitout)[,1:4] #use first intercept

outB<-plotB(exdat$rslt,exdat$dose,parms, sigma2=(sigma(fitout))^2,
ylab="Change in EDD")

plot(outB,plotDif=TRUE)

## End(Not run)
```

plotB

Plot Bayes dose response curve and dose group means

Description

Plot a dose response curve fit by Bayes MCMC methods (with optional posterior interval bars). Also plot dose group means (with optional CI bars)
**plotB**

**Usage**

```r
plotB(y, dose, parm, sigma2, count=rep(1,length(y)), dgrid=sort(unique(c(seq(0,max(dose),length=50), dose))), predict= TRUE,plotDif=FALSE,plotMed=FALSE, plotResid=FALSE,clev=0.9, binary=c('no','logit','probit','BinRes'),BinResLev, BinResDir=c('>','<'), activeControl=FALSE,ac,yac, countac=rep(1,length(yac)), labac='Act Comp',shapeac=8,colac='red', symbol,symbolLabel='Group',symbolShape=8, symbolColor='red',symbolSize=4, xlim,ylim,xat=NULL,xlab="Dose", ylab=ifelse(plotDif,"Diff with Comparator","Mean"), modelFun=emaxfun,makePlot=TRUE, ...)
```

**Arguments**

- **y**  Outcomes, which may be sample means (see counts). LSmeans from a saturated anacova model can be supplied, in which case it is assumed that the Bayesian dose response model also included the additive baseline covariates.
- **dose**  Doses corresponding to outcomes
- **parm**  Matrix of simulated parameter values (each row is a simulated parameter vector). The parm values must be constructed for use in the model function modFun. The default is a 4-parameter Emax model with parameters (log(ED50),lambda,Emax,E0). For a 3-parameter model, set lambda=1 for each simulated parameter vector.
- **sigma2**  Simulated draws from the residual variance (assumed additive, homogeneous). The length of sigma2 must be the same as the number of rows of parm. Set sigma2 to all ones for binary data.
- **count**  Sample sizes for means-only summarized data.
- **dgrid**  The Bayes posterior summaries are evaluated and plotted on the dgrid dosing values
- **predict**  If TRUE(default), the plotted intervals are predictive intervals for the dose group sample means.
- **plotDif**  Plot difference between doses and placebo. It is assumed the lowest dose is placebo. If activeControl, the difference is with the active control mean, and the active controls are not plotted.
- **plotMed**  If TRUE, model-based curves are medians rather than means.
- **plotResid**  If TRUE, a plot of the residuals formed from the dose group means minus the posterior dose group means.
clev  Level for confidence and Bayes intervals

binary  If binary is 'logit' or 'probit', y is assumed to be binary and the appropriate
backtransformation is applied to the Emax model output. If binary is 'Bin-
Res', the continuous variable y is converted to a binary responder variable using
BinResLev and BinResDir. The continuous Emax model output is converted to
binary estimation and prediction assuming normally distributed residuals.

BinResLev  A cut level for a responder variable formed from a continuous endpoint. Rates
are computed from the (continuous outcome) model parameters assuming nor-
mally distributed residuals. The input y variable is converted to a responder
variable.

BinResDir  If BinResDir='>', the responder variable is 1 when y is greater than the cut level,
otherwise, it is 1 when y is less than the cut level.

activeControl  When TRUE, active comparator data must be supplied. Each dose group (in-
cluding PBO) are compared to the active comparator rather than PBO.

ac  Simulations from the posterior distribution of the mean response on active com-
parator. The number of simulations must match those for the dose response
model. For binary data, the simulated values must be transformed to the propor-
tion scale. This differs from the simulated model parameters.

yac  Outcomes for the active comparator group. The coding conventions for y are
used.

countac  Sample sizes for summarized data corresponding to count.

labac  x-axis label for the active control group.

shapeac  Shape of the symbol for the active control group.

colac  Color of the symbol for the active control group.

symbol  An optional grouping variable for the dose group sample means.

symbolLabel  Label given to symbol in plot legend.

symbolShape  A character vector with names giving the shapes assigned to different levels of
variable symbol. If a single shape is specified, it is replicated for all dose groups.
See package ggplot2 for symbol mappings.

symbolColor  A character vector with names giving the colors assigned to different levels of
variable symbol. If a single color is specified, it is replicated for all dose groups.
See package ggplot2 for color mappings.

symbolSize  The size of the symbol for the dose group sample means. Set symbolSize=0 to
suppress plotting.

xlim  Plot limits for the x-axis

ylim  Plot limits for the y-axis

xat  The points at which tick-marks are to be drawn. Errors occur if the points are
outside the range of xlim. By default (when NULL) tickmark locations are
computed.

xlab  x-axis label

ylab  y-axis label
modelFun

The mean model function. The first argument is a scalar dose, and the second argument is a matrix of parameter values. The rows of the matrix are random draws of parameter vectors for the model. The default function is the 4-parameter Emax function emaxfun.

makePlot

If FALSE, create numerical output but no plot.

... Parameters passed to generic plot function (not used)

Details

A sample of parameters from the joint posterior distribution must be supplied (typically produced by BUGS). The Bayesian dose response curve is the Bayes posterior mean (or median) at each value on dgrid. The bar (interval) is the (clev/2,1-clev/2) Bayes posterior interval (which can differ from the Bayes HPD interval). The intervals are plotted only at the dose levels included in the study. Predictive intervals are formed by adding independent random draws from the sampling distributions of the dose group sample means to the population means.

The function generates random numbers when predict=TRUE, so the random number generator/seed must be set before the function is called for exact reproducibility.

Value

Returns an object of class plotB. Three inputs are saved for later plotting: doses in the original design, dgrid, and clev. The following matrices are saved:

pairwise The dose group means and their differences with placebo. If a baseline is supplied, the means are lsmeans adjusted to the mean baseline value.

modelABS Model-based posterior mean, median, posterior (clev/2,1-clev/2) intervals for the population means and sample means. One row per dose group

modelABSG Same as modelABS but computed on the input grid of doses.

modelDIF Same as modelABS but with differences from placebo.

modelDIFG Same as modelDIF but computed on the input grid of doses.

Note

PlotB can also be used with draws from a prior distribution to evaluate the prior dose response curve.

Author(s)

Neal Thomas

References


See Also

plot.plotB, plotD, plot.fitEmax
Examples

```r
## Not run:
data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
prior<-emaxPrior.control(epmu=0,epsca=100,diffTargetmu=0,diffTargetsca=100,dTarget=80.0,
p50=3.75,sigmaLow=0.01,sigmaUp=20)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
parms<-coef(fitout)[,1:4]  # use first intercept
outB<-plotB(exdat$rslt,exdat$dose,parms, sigma2=(sigma(fitout))^2,
ylab="Change in EDD")
plot(outB,plotDif=TRUE)
## End(Not run)
```

plotBdensity

Density plot displaying Bayes prior or posterior dose response

Description

Density plot over a grid of doses displaying the prior or posterior distribution for the mean dose response computed from simulated input model parameters.

Usage

```r
plotBdensity(dgrid,
  parm,
  modelFun=emaxfun,
  qlevL=c(0.025,0.05,0.10,0.25),
  plotDif= FALSE,
  logit= FALSE, ...)
```

Arguments

- **dgrid**: The Bayes prior or posterior summaries are evaluated and plotted on the `dgrid` dosing values.
- **parm**: Matrix of simulated parameter values (each row is a simulated parameter vector). The parm values must be constructed for use in the model function `modFun`. The default is a 4-parameter Emax model with parameters (log(ED50),lambda,Emax,E0). For a 3-parameter model, set lambda=1 for each simulated parameter vector.
modelFun  The mean model function. The first argument is a scalar dose, and the second argument is a matrix of parameter values. The rows of the matrix are random draws of parameter vectors for the model. The default function is the 4-parameter Emax function `emaxfun`.

qlevL  Intervals are formed with percentile boundaries at \((qlevL, 1-qlevL)\). \(qlevL\) must be increasing between \((0, 0.5)\).

plotDif  If TRUE, plot difference between doses and placebo.

logit  Default is F. If T, inverse logit transform applied to Emax function output for comparison to dose group sample proportions.

...  Parameters passed to generic plot function

Details

A sample of parameters from the joint prior or posterior distribution must be supplied (typically produced by BUGS). A density plot with contours corresponding to the percentiles in \(qlevL\) created by function `DRDensityPlot`.

Value

A list containing two matrices with the number of rows equal to the number dose grid points, and columns corresponding to percentiles in \(qlevL\):

\[qL\]  Lower percentiles from \(qlevL\)

\[qH\]  Upper percentiles \(1-qlevL\).

Author(s)

Neal Thomas

References


See Also

`plot.plotB, plotD, plot.fitEmax, DRDensityPlot`

Examples

```r
## Not run:
data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
prior<-emaxPrior.control(epmu=0, epsca=10, difTargetmu=0, difTargetsca=10, dTarget=80.0, p50=3.75, sigmalow=0.01, sigmapup=20)
mcmc<-mcmc.control(chains=3)
```
msSat <- sum((exdat$sampsize - 1) * (exdat$sd)^2) / (sum(exdat$sampsize) - length(exdat$sampsize))
fitout <- fitEmaxB(exdat$rslt, exdat$dose, prior, modType = 4, count = exdat$sampsize, msSat = msSat, mcmc = mcmc)
parms <- coef(fitout)[, 1:4]  # use first intercept
dgrid <- seq(0, 1, length = 100)
pout <- plotBdensity(dgrid, parm = parms)
pout2 <- plotBdensity(dgrid, parm = parms, plotDif = TRUE, xlab = "Var Dose", ylab = "Dif with PBO")

## End(Not run)

---

**plotD**

*Basic plot of dose group means*

**Description**

Plot dose group means vs dose with options to connect points by lines, and include CI about each dose group mean based on within-group SDs.

**Usage**

```r
plotD(y, dose, baseline, se = TRUE, line = TRUE, meansOnly = FALSE, sem = NULL, clev = 0.9, xlab = "Dose", ylab = "Response", logScale = FALSE)
```

**Arguments**

- `y`  Outcomes
- `dose`  Doses corresponding to outcomes
- `baseline`  If present, ANACOVA means are plotted, adjusted for baseline. Baseline is optional.
- `se`  If T, plot CI for each dose group.
- `line`  If T, dose group means are connected by a line
- `meansOnly`  If T, y contains dose group means rather than individual observations. Baseline cannot be specified.
- `sem`  If meansOnly and se=T, sem must contain the corresponding standard errors
- `clev`  Level of CI for dose group means
- `xlab`  Label for x-axis
- `ylab`  Label for y-axis
- `logScale`  If TRUE, log scale is used for dose.
Value

Returns a list with the ggplot object and two vectors with the dose group means and their standard errors.

Author(s)

Neal Thomas

See Also

plot.fitEmax, plotB

Examples

data("metaData")
exdat<-metaData[metaData$taid==2 & metaData$etype==1,]
with(exdat,plotD(rslt,dose,meansOnly=TRUE,se=TRUE,sem=se,ylab="Y",xlab="Dose(mg)"))

predict.emaxalt Mean response and SE for specified doses for a simulated object output by function emaxalt

Description

Estimated mean and standard error for specified doses computed from the output of a model fit by function emaxalt. Also returns mean difference with placebo and their standard errors.

Usage

## S3 method for class 'emaxalt'
predict(object, dose, dref=0, ...)

Arguments

object Output of emaxalt
dose Vector (can be a single value) of doses where dose response curve is to be evaluated.
dref A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.
... Optional arguments are not used.
Value

A list containing:

- `fitpred`: Vector with mean dose response estimate for each specified dose.
- `fitdif`: Corresponding differences with placebo.
- `sepred`: SEs for `fitpred`.
- `sedif`: SEs for `fitdif`.

Author(s)

Neal Thomas

See Also

`emaxalt`, `predict.emaxsimobj`, `predict.emaxsim`  

Examples

```r
## Not run:
## random number seed changed by this example

doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
dose<-rep(doselev,n)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop.parm<-c(log(ed50),e0,emax)
meanresp<-emaxfun(dose,pop.parm)
y<-rnorm(sum(n),meanresp,sdy)

simout<-emaxalt(y,dose)
predict(simout,c(75,150))

simout2<-emaxalt(y,dose,modType=4)
predict(simout2,c(75,150))

## End(Not run)
```
**predict.emaxsim**

**Mean response and SE for specified doses for each replicate data set in an emaxsim object**

### Description

Estimated mean/proportion and standard error for each simulated data set in an emaxsim object. Also returns mean difference with placebo and their standard errors.

### Usage

```r
## S3 method for class 'emaxsim'
predict(object, dose, dref=0, ...)
```

### Arguments

- **object**: Output of `emaxsim`
- **dose**: Vector (can be a single value) of doses where dose response curve is to be evaluated.
- **dref**: A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.
- **...**: Optional arguments are not used.

### Value

A list containing:

- **fitpredv**: Matrix with mean dose response estimate for each simulated data set. Number of columns is the number of doses specified.
- **fitdifv**: Matrix with mean dose response estimate minus mean placebo response for each simulated data set. Number of columns is the number of doses specified.
- **sepredv**: Matrix of SEs for `fitpredv`.
- **sedifv**: Matrix of SEs for `fitdifv`.

### Author(s)

Neal Thomas

### See Also

`emaxsim`, `summary.emaxsim`, `plot.emaxsim`
Examples

```r
## Not run:
## random number seed changed by this example
nsim<-50
idmax<-5
doselev<-c(0, 5, 25, 50, 100)
n<-c(78, 81, 81, 81, 77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget, dtarget, log(ed50), 1, e0)
sdy<-7.967897
pop.parm<-c(log(ed50), emax, e0)
meanlev<-emaxfun(doselev, pop.parm)

### FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n, doselev, meanlev, sdy)
D1 <- emaxsim(nsim, gen.parm)
predout<-predict(D1, c(75, 150))

## End(Not run)
```

---

**predict.emaxsimB**

*Mean response and SE for each replicate data set in an emaxsimB object*

**Description**

Return warning and explanation that only predicted values at doses included in the study are available. The code needed to obtain predicted values at other doses is indicated.

**Usage**

```r
## S3 method for class 'emaxsimB'
predict(object, dose, dref=0, ...)
```

**Arguments**

- **object** Output of `emaxsim`
predict.emaxsimB

dose Vector (can be a single value) of doses where dose response curve is to be evaluated.
dref A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.

Value
No output.

Author(s)
Neal Thomas

See Also
emaxsimB, summary.emaxsimB, plot.emaxsimB

Examples

## Not run:
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,
propInit=0.15,adapt_delta = 0.95)

D1 <- emaxsimB(nsim,gen, prior, modType=3,seed=12357,mcmc=mcmc,check=FALSE)
predict(D1, dose=20)

## End(Not run)

predict.emaxsimBobj  

Mean response estimates (posterior means) and SE (posterior SD) for specified doses for a simulated emaxsimBobj object

Description
Estimated mean and standard error for specified doses (posterior means and SD) computed from the output of a simulated data set created by function emaxsimB. Also returns mean difference with placebo and their standard errors.

Usage
## S3 method for class 'emaxsimBobj'
predict(object,  
    dose, dref=0, clev=0.9,  
    ...)  

Arguments
object  
Output of the extract function [] applied to an object created by emaxsimB.
dose  
Vector (can be a single value) of doses where dose response curve is to be evaluated.
dref  
A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.
clev  
Specified probability of the posterior interval
...
Optional arguments are not used.

Value
A list containing:
pred  
Vector with mean dose response estimates for each specified dose.
fitdif  
Corresponding differences with placebo.
se  
SEs (posterior SD) for pred.
sedif  
SEs (posterior SD) for fitdif.
lb, ub, lbdif, ubdif  
Bounds of clev posterior intervals.

Author(s)  
Neal Thomas
See Also

emaxsim, summary.emaxsim, predict.emaxsim

Examples

```r
## Not run:
### emaxsimB changes the random number seed
nsim<-50
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)

D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)
predict(D1[1],dose=c(75,125))
## End(Not run)
```

```r
predict.emaxsimobj  Mean response and SE for specified doses for a simulated emaxsimobj object
```

Description

Estimated mean/proportion and standard error for specified doses computed from the output of a simulated data set created by function emaxsim. Also returns mean difference with placebo and their standard errors.
Usage

## S3 method for class 'emaxsimobj'
predict(object,
    dose, dref=0,
    ...)  

Arguments

object Output of the extract function applied to an object created by emaxsim.
dose Vector (can be a single value) of doses where dose response curve is to be evaluated.
dref A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.
...

Optional arguments are not used.

Value

A list containing:

- `fitpred` Vector with mean dose response estimate for each specified dose.
- `fitdif` Corresponding differences with placebo.
- `sepred` SEs for `fitpred`.
- `sedif` SEs for `fitdif`.

Author(s)

Neal Thomas

See Also

emaxsim, summary.emaxsim, predict.emaxsim

Examples

## Not run:
## emaxsim changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdv<-7.967897
pop.parm<-c(log(ed50), emax, e0)
meanlev<-emaxfun(doselev, pop.parm)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n, doselev, meanlev, sdy)
D1 <- emaxsim(nsim, gen.parm)
d10<-D1[10]
predict(d10, c(75, 150))

# End(Not run)

predict.fitEmax

Estimated mean/proportion and confidence intervals derived from the
maximum likelihood fit of a 3- or 4- parameter Emax model.

Description

The estimated means from an Emax model is computed along with confidence bounds. The results
are computed for a vector of input dose levels. For binary outcomes, the results are computed on
the logit scale and then back-transformed.

Usage

## S3 method for class 'fitEmax'
predict(object, dosevec, clev=0.9,
        int=1, dref=0, xvec=NULL, ...)

Arguments

object Output of fitEmax with class "fitEmax".
dosevec Vector of doses to be evaluated.
clev Confidence level for intervals about the estimated mean/proportion at each do-
        sevec.
int The index for the protocol (intercept) to use for the predictions
dref Differences in response between doselev and dref are computed.
xvec The vector of centered baseline values for the prediction model when xbase was
        specified in the model fit. Centering must be done using the protocol-specific
        means consistent with int. See details for the default calculations when xvec is
        not specified.
...

No additional parameters will be utilized.
Details

Model estimates, standard errors, and confidence bounds are computed with the function `SeEmax`. If baseline covariates were included in the fit and `xvec` is not specified, then the predicted value is the mean of the predictions for all patients in the specified protocol. Note that the protocol must be specified, or the prediction defaults to patients from the first protocol. Note that for binary data, the distinction between the mean of the predicted values and the predicted value as the mean of the covariates can be important.

Value

A list with estimated dose group means/proportions, lower bound, upper bound, SE, and corresponding values for differences with the reference dose. One value for each dose in `dosevec`.

Author(s)

Neal Thomas

See Also

`nls`

Examples

```r
## Not run:
## this example changes the random number seed
doselev<-c(0,5,25,50,100,350)
n<-c(78,81,81,81,77,80)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-8.0
pop.parm<-c(log(ed50),emax,e0)
dose<-rep(doselev,n)
meanlev<-emaxfun(dose,pop.parm)
y<-rnorm(sum(n),meanlev,sdy)
testout<-fitEmax(y,dose,modType=4)
predout<-predict(testout,dosevec=c(20,80),int=1)

## End(Not run)
```
predict.fitEmaxB

Estimated mean and posterior intervals derived from a Bayesian hyperbolic or sigmoidal Emax model.

Description

The mean/proportion response for different doses estimated from a Bayesian Emax model is computed along with corresponding posterior intervals. The results are computed for a vector of input dose levels. The estimates are posterior means or medians of the MCMC generated means/proportions. For binary outcomes, the estimated response rates are computed on the logit scale and then back-transformed before forming the estimates and posterior intervals.

Usage

```r
## S3 method for class 'fitEmaxB'
predict(object, dosevec, clev = 0.9, int = 1, dref = 0, xvec=NULL, ...)
```

Arguments

- `object`: Output of `fitEmax` with class "fitEmaxB".
- `dosevec`: Vector of doses to be evaluated.
- `clev`: Level for the posterior intervals about the mean/proportion at each `dosevec`.
- `int`: The index for the protocol (intercept) to use for the predictions.
- `dref`: Differences in response between `dosevec` and `dref` are computed.
- `xvec`: The vector of centered baseline values for the prediction model when `xbase` was specified in the model fit. Centering must be done using the protocol-specific means consistent with `int`. See details for the default calculations when `xvec` is not specified.
- `...`: No additional parameters will be utilized.

Details

Results computed from simple tabulations of the MCMC parameters evaluated in the Emax function.

If baseline covariates were included in the fit and `xvec` is not specified, then the predicted value is the mean of the predictions for all patients in the specified protocol. Note that the protocol must be specified, or the prediction defaults to patients from the first protocol. Note that for binary data, the distinction between the mean of the predicted values and the predicted value as the mean of the covariates can be important.

Value

A list with estimated mean/proportion (pred, predMed), lower bound, upper bound, posterior SD, and corresponding values for differences with the reference dose. One value for each dose in `dosevec`. The MCMC response means (proportions for binary data) are in `simResp`, and the residual SD for continuous data are in `sigsim`. 
Author(s)
Neal Thomas

See Also
fitEmaxB

Examples

```r
## Not run:
data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
prior<-emaxPrior.control(epmu=0,epsca=10,difTargetmu=0,difTargetsc=10,dTarget=80.0,
p50=3.75,smalow=0.01,sigmaup=20)
mcmc<-mcmc.control(chains=3)
msSat<-sum((exdat$sampsize-1)*(exdat$sd)^2)/(sum(exdat$sampsize)-length(exdat$sampsize))
fitout<-fitEmaxB(exdat$rslt,exdat$dose,prior,modType=4,
count=exdat$sampsize,msSat=msSat,mcmc=mcmc)
predout<-predict(fitout,dosevec=sort(unique(exdat$dose)))
## End(Not run)
```

print.emaxsim

Print simulation output from emaxsim

Description
Prints key summary variables of Emax estimation performance for each simulation. Can be used to identify simulated data sets yielding problems with common estimation methods.

Usage

```r
## S3 method for class 'emaxsim'
print(x,
nprint = min(length(x$fitType), 20),
id = x$idmax,
digits = 3, ...)
```

Arguments

- **x**: Output of `emaxsim`
- **nprint**: Number of simulations to print. If a vector of length 2, `nprint` is the range of simulations to print.
- **id**: Output includes the stdBias for the dose with index `id` vs placebo
print.emaxsim

digits Number of decimal digits to print for Z and p-values
...
Other print parameters (none currently implemented)

Value
Printed output returned as invisible matrix.

Note
The stdBias printed is the difference between the estimated dose response at the dose with index id and its population value. The difference is divided by the SE of the estimator computed using the delta method.

Author(s)
Neal Thomas

See Also
emaxsim, summary.emaxsim, plot.emaxsim

Examples

## Not run:
## emaxsim changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113

dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)

sdy<-7.967897
pop.parm<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop.parm)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen.parm)

print(D1,c(31,50),digits=2,id=4)

print(D1,c(1,20))
## implicitly calls print with default parameter settings
## End(Not run)

---

### print.emaxsimB

*Print simulation output from emaxsimB*

#### Description

Prints key summary variables of Emax estimation performance for each simulation. Can be used to identify simulated data sets yielding unusual estimates.

#### Usage

```r
## S3 method for class 'emaxsimB'
print(x,
    nprint = min(nsim, 20),
    id = x$idmax,
    digits = 3, ...)
```

#### Arguments

- `x`: Output of `emaxsimB`
- `nprint`: Number of simulations to print. If a vector of length 2, `nprint` is the range of simulations to print.
- `id`: Output includes the stdBias for the dose with index `id` vs placebo
- `digits`: Number of decimal digits to print for Z and p-values
- `...`: Other print parameters (none currently implemented)

#### Value

Printed output returned as invisible matrix.

#### Note

The stdBias printed is the difference between the posterior mean of the dose response at the dose with index `id` and its population value. The difference is divided by the SE (posterior SD).

#### Author(s)

Neal Thomas

#### See Also

`emaxsimB`, `summary.emaxsimB`, `plot.emaxsimB`
Examples

## Not run:
## emaxsimB changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)

D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)
print(D1)

## End(Not run)

print.emaxsimBobj

Print a summary of the fitted Emax model

Description

Print a summary of the fitted Emax model. Printed output returned as invisible matrix.

Usage

## S3 method for class 'emaxsimBobj'
print(x, nprint=min(length(x$y),20), ...)

S3 method for class 'emaxsimBobj'
print(x, nprint=min(length(x$y),20), ...)
Arguments

- **x**: Object output by the extractor function [] for `emaxsimB`
- **nprint**: Number of observations to print. If a vector of length 2, `nprint` is the range of data to print.
- **...**: No options implemented.

**print.emaxsimobj**  
*Print a data set generated by `emaxsim`*

Description

Print a data set that has been extracted from emaxsim output

Usage

```r
## S3 method for class 'emaxsimobj'
print(x, nprint = min(length(x$y), 20), ...)
```

Arguments

- **x**: Extracted simulation object
- **nprint**: Number of observations to print. If a vector of length 2, `nprint` is the range of data to print.
- **...**: No other parameters currently implemented

Value

Printed output returned as invisible matrix.

Author(s)

Neal Thomas

See Also

`emaxsim`, `plot.emaxsimobj`, `summary.emaxsimobj`

Examples

```r
## Not run:
save.seed<-.Random.seed
set.seed(12357)
nsim<-50
idmax<-5
```
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen.parm)
e49<-D1[49]

e49

print(e49,c(101,200))

.Random.seed<-save.seed

## End(Not run)

print.fitEmax

Print a summary of the fitted Emax model

Description

Print a summary of the fitted Emax model

Usage

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

Arguments

x Object output by fitEmax

... No options implemented.
**print.fitEmaxB**  
*Print a summary of the fitted Bayesian Emax model*

**Description**
Print a summary of the fitted Bayesian Emax model

**Usage**
```r
## S3 method for class 'fitEmaxB'
print(x, ...)  
```

**Arguments**
- `x`: Object output by `fitEmaxB`
- `...`: No options implemented.

---

**prior.control**  
*Set the parameters of the prior distribution for the Emax model implemented in fitEmaxB.*

**Description**
Set the parameters of the prior distribution for the Emax model implemented in `fitEmaxB`. `prior.control` is deprecated. See `emaxPrior.control`.

**Usage**
```r
prior.control(epmu = NULL, epsd = NULL, emaxmu = NULL, emaxsd = NULL,  
p50 = NULL, sigmalow = NULL, sigmaup = NULL,  
led50mu = 0.79, led50sca = 0.6, edDF = 3,  
lama = 3.03, lamb = 18.15, lamsca = 6,  
basemu=NULL,basevar=NULL,  
binary = FALSE)
```

**Arguments**
- `epmu`: Mean for E0 in a normal prior distribution. Logistic scale for binary data.
- `epsd`: SD for E0 in a normal prior distribution. Logistic scale for binary data.
- `emaxmu`: Mean for Emax in a normal prior distribution. Logistic scale for binary data.
- `emaxsd`: SD for Emax in a normal prior distribution. Logistic scale for binary data.
- `p50`: Projected ED50. See reference for its use in creating the prior distribution for the ED50.
sigmalow  Lower bound for a uniform prior distribution for the residual SD (continuous data).
sigmaup  Upper bound for a uniform prior distribution for the residual SD (continuous data).
led50mu  Mean of log-t prior distribution for the ED50 before final scaling. See reference for its interpretation in the prior distribution for the ED50.
led50sca  Scale (analogous to SD) of the log-t prior distribution for the ED50.
edDF  The degrees of freedom of the log-t prior distribution for the ED50.
lama  Parameter in the re-scaled beta distribution for Hill slope parameter in the sigmoidal Emax model. See reference for its use and empirical basis.
lamb  Parameter in the re-scaled beta distribution for Hill slope parameter in the sigmoidal Emax model.
lamsca  The beta prior distribution for the Hill parameter is re-scaled to have support on (0,lamsca).
basemu  A vector of prior means for the covariate regression parameters.
basevar  The prior variance-covariance matrix for the covariate regression parameters. The covariate regression parameters are apriori independent of the other dose response model parameters.
binary  Set to TRUE for binary data applications. Used to check for consistency in usage.

Details

The prior distributions are based two meta-analyses of dose response described in the references. Each parameter is independent in the prior distribution. The E0 and Emax parameters have normal prior distributions. For binary data, these parameters are computed on the logistic scale. The predicted ED50 must be specified as 'P50'. The prior distribution of the log(ED50) has a t-distribution centered at log(P50), with scale, degrees of freedom, and offset to the P50, defaulting to values given in the references (these can be changed, but they are difficult to interpret outside the context of the meta-analyses). If modType=4, the prior distribution for the Hill parameter is a beta distribution scaled to (0,lamsca). The default degrees of freedom were obtained from the meta-analyses. For continuous data, the prior distribution for the residual SD is uniform on a user-specified scale.

Value

List of prior parameter values for use in fitEmaxB.

Author(s)

Neal Thomas

References


RandEmax

Random data constructor function for emaxsim creating random parameters for an Emax model for continuous or binary data.

Description

Creates a list object that contains inputs and a function to create simulated data sets for emaxsim. Data sets are created by generating random parameters from beta or log-normal distributions for a 3/4 parameter Emax model. For binary data, the Emax model is on the logit scale and then back-transformed.

Usage

RandEmax(n, doselev, parmEmax, parmE0, p50, parmED50=c(3,0.79,0.6), parmLambda=c(3.03,18.15,0,6), resSD, dfSD=Inf, binary=FALSE)

Arguments

- **n**: Sample size for each dose group.
- **doselev**: Dose levels (including 0 for placebo) included in the study corresponding to n. Must be in increasing order.
- **parmEmax**: Vector with mean and standard deviation for a random normal Emax
- **parmE0**: Vector with mean and standard deviation for a random normal intercept.
- **p50**: The predicted ED50
- **parmED50**: The log(ED50) is generated from a t-distribution with df=parmED50[1], mean=log(p50)+parmED50[2], and scale=parmED50[3]. The default values are taken from the reference below.
- **parmLambda**: For a beta distributed sigmoid lambda, a vector with (df1,df2,lower bound, upper bound). For a hyperbolic model, lambda=1.
- **resSD**: Standard deviation for residuals within each dose (normal data only)
- **dfSD**: If a finite value is specified, the within-dose group SD is randomly generated from resSD times sqrt(dfSD/chisquare(dfSD))), which is the form of a posterior distribution for a SD based on a existing sample.
- **binary**: When TRUE, 0/1 data are generated from the Emax model, which is computed on the logit scale and then backtransformed to yield proportions.

See Also

fitEmaxB
Details

All parameters are independent. Normal data are generated from the dose response curves with homogeneous-variance normal residuals. Binary data are 0/1 generated from Bernoulli distributions with proportions computed by transforming the Emax model output from the logit to proportion scale. Default values are based on recommendations in Thomas, N., Sweeney, K., and Somayaji, V. (2014). Meta-analysis of clinical dose response in a large drug development portfolio. <doi:10.1080/19466315.2014.924876>

Value

A list of length 2. The first element is itself a list named genP that contains named elements n, resSD, dfSD, doselev, dose, binary and the elements parmE0, p50, parmED50, parmEmax, and parmLambda. which are specific to RandEmax. The second element is a function named genFun that takes genP as input and returns a list with named elements meanlev, parm, resSD, y.

Author(s)

Neal Thomas

See Also

emaxsim, FixedMean

Examples

```r
simParm<-RandEmax(n=c(99,95,98,94,98,98),doselev=c(0,5,10,25,50,150),
parmE0=c(-2.6,2.5),p50=25,parmEmax=c(-1.25,2),resSD=3.88)
```

runShiny

Shiny app for function emaxsim(B)

Description

Shiny app for function emaxsim(B)

Usage

runShiny()

Note

The code section of the shiny app provides the code required for batch execution of the current shiny results.

The ‘Analysis’ section of the shiny app must be visited before an example can be run.

For Bayesian output, the clinDR package function compileStanModels() must be executed once before using the shiny app or any of the package functions utilizing Bayes methods.
Author(s)
Neal Thomas, Mike K. Smith

See Also
emaxsimB

Examples

```r
if (interactive()) {
  runShiny()
}
```

---

SeEmax

**Asymptotic SE for dose response estimates from a 3- or 4-parameter Emax model**

**Description**

Compute the asymptotic SE for dose response estimates based on the asymptotic variance-covariance matrix from the fit of a 3- or 4-parameter Emax model

**Usage**

```r
SeEmax(fit, doselev, modType, dref=0, nbase=0, x=NULL,
       binary=FALSE, clev=0.9)
```

**Arguments**

- `fit` Output of `nls` fit to a 3- or 4-parameter Emax model. The order of the parameters in the fit must be (log(ed50), emax, e0) or (log(ed50), lambda, emax, e0). Alternatively, fit can be a list with the first element the coefficient vector, and the second element the variance-covariance matrix. List input can be used with multiple protocols and baseline covariates (see details).
- `doselev` SEs are evaluated at vector of doses
- `modType` modType=3,4 for a 3 or 4 parameter model.
- `dref` A reference dose (0 by default) for contrasts, but other values can be specified. If specified, a single reference value must be given.
- `nbase` The number of baseline predictors included in the model.
- `x` The model is evaluated at baseline covariate values, `x`. If `x` is a matrix, then each row is a vector of baseline predictors, and the results are for the dose response averaged over all of the predictors in `x`.
- `binary` Emax model on logistic scale, then backtransformed.
- `clev` Confidence level for intervals.
Details

The Emax models supported by SeEmax should now be fit using fitEmax and predict.fitEmax. SeEmax remains available primarily for backward compatibility.

SeEmax can be used with models that allow different placebo response for multiple protocols by selecting the intercept for a specific protocol. Coefficients for baseline covariates can also be included following the intercept. The variance-covariance matrix from the full model must be subsetted to match the included coefficients (i.e., the rows and columns corresponding to the omitted intercepts must be removed). List input must be used for the more general models.

Value

Returns a list:

- `doselev` Doses to evaluate
- `dref` Differences in response between doselev and dref are computed.
- `fitpred` Estimated dose response at doselev
- `sepred` SE for estimated dose responses
- `fitdif` Estimated response at doselev minus estimated response at placebo
- `sedif` SE for fitdif estimated differences
- `fitref` Estimated dose response at the reference dose.
- `seref` SE for the estimated dose response at the reference dose
- `covref` The covariance between each estimated response and the estimated response at the reference dose. These covariances can be used to compute asymptotic variances of differences after back-transformation (e.g., for logistic regression with binary data).

Author(s)

Neal Thomas

References


See Also

fitEmax

Examples

```r
## Not run:

## this example changes the random number seed
doselev<-c(0,5,25,50,100,250)
n<-c(78,81,81,81,77,80)
dose<-rep(doselev,n)
```
### population parameters for simulation

e0<-2.465375  
ed50<-67.481113  
led50<-log(ed50)  
lambda=1.8

dtarget<100  
diftarget<-9.032497  
emax<-solveEmax(diftarget,dtarget,log(ed50),lambda,e0)

sdy<-7.967897  
pop<-c(led50,lambda,emax,e0)

meanresp<-emaxfun(dose,pop)  
y<-rnorm(sum(n),meanresp,sdy)

nls.fit<-nls(y ~ e0 + (emax * dose^lambda)/(dose^lambda + exp(led50*lambda)),  
start = pop, control = nls.control(maxiter = 100),trace=TRUE,na.action=na.omit)

SeEmax(nls.fit,doselev=c(60,120),modType=4)  
SeEmax(list(coef(nls.fit),vcov(nls.fit)),c(60,120),modType=4)

## End(Not run)

---

**selEstan**

*Select a pre-compiled rstan Emax model*

**Description**

Emax models for use in `fitEmaxB` and `emaxsimB` which have been pre-compiled are loaded for use outside of the fitting functions. This is most useful for repeated simulations in which the loading of the compiled models from a disk file can be performed once. `fitEmaxB` will load the model automatically for single execution, so the model does not need to be pre-loaded.

**Usage**

```
selEstan(emod=c('basemodel.rds','mrmodel.rds'))
```

**Arguments**

- **emod**
  
  Two parameterizations of the emax function are currently supported. 'basemodel' uses the maximal effect 'emax' parameter. 'mrmodel' uses the effect of the drug at a high dose specified by the user versus placebo. The 'emax' effect model is deprecated and will be eliminated.
showStanModels

Value
An Emax 'stanmodel'.

Author(s)
Neal Thomas

See Also
fitEmaxB, emaxsimB

Examples
```r
## Not run:
estan<-selEstan()

## End(Not run)
```

showStanModels Display STAN model code.

Description
Display the STAN Bayesian model code for fitting Emax models

Usage
```
showStanModels(emod=c('basemodel.stan','mrmodel.stan'))
```

Arguments
- `emod` Two parameterizations of the emax function are currently supported. 'basemodel' uses the maximal effect 'emax' parameter. 'mrmodel' uses the effect of the drug at a high dose specified by the user versus placebo. The 'emax' effect model is deprecated and will be eliminated.

Author(s)
Neal Thomas

See Also
fitEmaxB, emaxsimB

Examples
```r
## Not run:
showStanModels()

## End(Not run)
```
Extract Emax model residual SD estimates.

Usage

## S3 method for class 'fitEmax'
sigma(object, ...)
## S3 method for class 'fitEmaxB'
sigma(object, ...)
## S3 method for class 'emaxsim'
sigma(object, ...)
## S3 method for class 'emaxsimB'
sigma(object, ...)

Arguments

object Output of Emax fitting and simulation functions
...
None additional inputs supported

Value

MLE estimate of the residual SD from fitEmax. Vector of MLE estimates of the residual SD for each emaxsim simulation. Vector of MCMC generated residual SD for fitEmaxB. Vector of posterior median estimates of the residual SD for each emaxsimB simulation.

Author(s)

Neal Thomas

See Also

coe, fitEmax, fitEmaxB, emaxsim, emaxsimB

Examples

doselev<-c(0, 5, 25, 50, 100, 350)
n<-c(78, 81, 81, 81, 77, 80)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
startEmax <- solveEmax(diftarget, dtarget, log(ed50), 1, e0)

sdy <- 8.0
pop <- c(log(ed50), emax, e0)
dose <- rep(doselev, n)
meanlev <- emaxfun(dose, pop)

y <- rnorm(sum(n), meanlev, sdy)

testout <- fitEmax(y, dose, modType = 4)
sigma(testout)

---

**startEmax**

*Compute starting parameter values for the 3- or 4- Emax model.*

**Description**

Compute starting parameter values for iterative procedures for estimating parameters of the 3- or 4-parameter Emax model.

**Usage**

```r
startEmax(y, 
  dose, 
  baseline, 
  count = rep(1, length(y)), 
  modType = 3, 
  binary = FALSE, 
  lbED50 = doselev[2]/10, 
  ubED50 = max(doselev), 
  lbLambda = 0.5, 
  ubLambda = 5)
```

**Arguments**

- **y**: Outcome (response) variable for the Emax modeling.
- **binary**: The default is continuous (binary = FALSE). When (binary = TRUE), y must be 0/1 and starting values are returned for an Emax model on the logit scale.
- **dose**: Dose variable corresponding to each outcome value.
- **baseline**: Optional baseline covariate(s) of same length as y. When baseline is specified, starting values are created from anacova adjusted dose group means.
- **count**: Counts for the number of patients with each dose/y value. Default is 1 (un-grouped data).
- **modType**: modType = 3 (default) for the 3-parameter Emax model. modType = 4 for the 4-parameter Emax model.
- **lbED50**: If the starting ED50 is below lbED50, it is set to lbED50.
ubED50    If the starting ED50 is above ubED50, it is set to ubED50.
1bLambda  If the starting lambda is below lbLambda, it is set to lbLambda.
ubLambda  If the starting lambda is above ubLambda, it is set to ubLambda.

Value

Returns a vector with named elements for the starting values for a 3 or 4 parameter Emax model. The order is log(ED50), (lambda, 4 parm), emax, and e0. If baseline is specified, a 'beta' starting parameter is also returned at the end of the vector.

Note

The method is modified from functions created by J. Rogers and start functions supplied with R (SSfp1). The ED50 (and lambda) are computed using the logit-linear relationship between the proportion of the mean response out of the max response and the log(dose). The method assumes placebo data are present, but it will return a starting value even if it is not present. A minimum of four dose levels is required for 4-parameter starting values.

Author(s)

Neal Thomas

See Also

nls, emaxalt

Examples

data("metaData")
exdat<-metaData[metaData$taid==6 & metaData$poptype==1,]
startEmax(exdat$rslt,exdat$dose)

summary.emaxsim    Summary of output of emaxsim

Description

Detailed summary of repeated sampling properties of Emax estimation and comparison with simple pairwise comparisons.

Usage

```r
## S3 method for class 'emaxsim'
summary(object, testalpha = 0.05, clev = 0.9,
        seSim = FALSE, ...)
```
**summary.emaxsim**

**Arguments**

- `object`  
  - Output of `emaxsim`

- `testalpha`  
  - Alpha level for a one-sided MCP-MOD trend test

- `clev`  
  - Nominal confidence level for reported CIs

- `seSim`  
  - If TRUE, then simulation standard errors are reported in parentheses. These should be distinguished from standard errors for estimators in the simulation.

- `...`  
  - Other unspecified parameters (none currently utilized)

**Details**

For pairwise comparisons, the 'most favorable pairwise comparison' means the dose with the best difference versus placebo is compared to the population mean response for the selected dose, thus the target value for coverage, bias, and RMSE changes depending on the selected dose.

**Value**

The function produces annotated output summarizing the properties of the estimation procedures. The summaries are also returned as an invisible list for extracting results.

**Author(s)**

Neal Thomas

**See Also**

- `emaxsim`, `print.emaxsim`, `plot.emaxsim`

**Examples**

```r
## Not run:
## emaxsim changes the random number seed
nsim<-50
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop.parm<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop.parm)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n,doselev,meanlev,sdy)
```
D1 <- emaxsim(nsim,gen.parm)
summary(D1,testalph=0.05,clev=0.95)

## End(Not run)

summary.emaxsimB  
*Summary of output of emaxsimB*

**Description**

Detailed summary of repeated sampling properties of Bayesian Emax estimation and comparison with simple pairwise comparisons.

**Usage**

```r
## S3 method for class 'emaxsimB'
summary(object, testalpha = 0.05,
clev = c('0.9','0.95','0.8'),
seSim = FALSE, ...)
```

**Arguments**

- `object`: Output of `emaxsimB`
- `testalpha`: Alpha level for a one-sided MCP-MOD trend test.
- `clev`: Posterior probabilities for reported intervals
- `seSim`: If TRUE, then simulation standard errors are reported in parentheses. These should be distinguished from posterior SD in the simulations.
- `...`: Other unspecified parameters (none currently utilized)

**Details**

For pairwise comparisons, the 'most favorable pairwise comparison' means the dose with the best difference versus placebo is compared to the population mean response for the selected dose, thus the target value for coverage, bias, and RMSE changes depending on the selected dose.

**Value**

The function produces annotated output summarizing the properties of the estimation procedures. The summaries are also returned as an invisible list for extracting results.

**Author(s)**

Neal Thomas

**See Also**

`emaxsim`, `print.emaxsim`, `plot.emaxsim`
Examples

```r
## Not run:
## `emaxsimB` changes the random number seed
nsim<-50
idmax<-5
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

### FixedMean is specialized constructor function for emaxsim
gen<- FixedMean(n,doselev,meanlev,sdy)
prior<- emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmalow=0.1,
sigmaup=30,parmDF=5)
mcmc<- mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)
D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)

summary(D1,testalph=0.05,clev='0.95')

## End(Not run)
```

Description

Summary of the Bayesian Emax fit to a simulated data set

Usage

```r
## S3 method for class 'emaxsimBobj'
summary(object, ...)
```
Arguments

object: Extracted simulation object

...: No other parameters are currently implemented

Value

Printed output only. No values are returned.

Author(s)

Neal Thomas

See Also

emaxsimB, plot.emaxsimBobj, print.emaxsimBobj

Examples

## Not run:

## emaxsimB changes the random number seed
nsim<-50
doselev<-c(0.5,25,50,100)
n<-c(78,81,81,81,77)
Ndose<-length(doselev)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-2.464592
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<-FixedMean(n,doselev,meanlev,sdy)
prior<-emaxPrior.control(epmu=0,epsca=30,difTargetmu=0,
difTargetsca=30,dTarget=100,p50=50,sigmaLw=0.1,
sigmaup=30,parmDF=5)
mcmc<-mcmc.control(chains=1,warmup=500,iter=5000,seed=53453,propInit=0.15,adapt_delta = 0.95)
D1 <- emaxsimB(nsim,gen, prior, modType=3,mcmc=mcmc,check=FALSE)
summary(D1[1])

## End(Not run)
Description

Summary of the Emax or alternative fit to a simulated data set

Usage

```r
## S3 method for class 'emaxsimobj'
summary(object, ...)
```

Arguments

- `object`: Extracted simulation object
- `...`: No other parameters are currently implemented

Value

Printed output only. No values are returned.

Author(s)

Neal Thomas

See Also

`emaxsim`, `plot.emaxsimobj`, `print.emaxsimobj`

Examples

```r
## emaxsim changes the random number seed
nsim<-3
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)

meanlev<-emaxfun(doselev,pop)
```
### FixedMean is specialized constructor function for emaxsim

gen.parm<-FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen.parm,nproc=1)
e3<-D1[3]

summary(e3)

---

### summary.fitEmax

*Print a summary of the fitted Emax model*

**Description**

Print a summary of the fitted Emax model

**Usage**

```r
## S3 method for class 'fitEmax'
summary(object, ...)
```

**Arguments**

- `object` Object output by `fitEmax`
- `...` No options implemented.

---

### summary.fitEmaxB

*Print a summary of the fitted Bayesian Emax model*

**Description**

Print a summary of the fitted Bayesian Emax model

**Usage**

```r
## S3 method for class 'fitEmaxB'
summary(object, ...)
```

**Arguments**

- `object` Object output by `fitEmaxB`
- `...` No options implemented.
**targetBeta**  

*Find a scaled Beta distribution matching specified probabilities*

**Description**

Find the (a,b) parameters of a scaled Beta distribution with specified cumulative probabilities for two specified points from the distribution.

**Usage**

```
targetBeta(minval,pminV,pmaxV,maxval=1,aInit=1,bInit=1,upB=1)
```

**Arguments**

- `minval` The minimum value with a targeted cumulative probability
- `pminV` The targeted cumulative probability less than `minval`
- `pmaxV` The targeted cumulative probability less than `maxval`
- `maxval` The maximum value with a targeted cumulative probability
- `aInit` An initial guess for the first parameter of the scaled Beta distribution with the specified probabilities.
- `bInit` An initial guess for the second parameter of the scaled Beta distribution with the specified probabilities.
- `upB` The upper limit of the scaled Beta distribution. It is specified by the user.

**Details**

The Beta distribution with the targeted probabilities is found from starting values using the `optim` function.

**Value**

Returns the (a,b) parameters of the scaled beta distribution if one with the specified probabilities can be found. An error message is returned otherwise.

**Author(s)**

Neal Thomas

**Examples**

```R
### set quartiles at .15 and 1.0 for a beta distribution on (0,3)
targetBeta(minval=.15,pminV=0.25,pmaxV=0.75,maxval=1.0,upB=3)
```
targetCI

*Compute the dose with confidence interval exceeding a target change from placebo for each simulated example in an emaxsim object.*

**Description**

Selects the lowest dose from a user-specified grid of doses with confidence interval exceeding a targetted change from placebo for each simulated data set in an emaxsim object.

**Usage**

```r
targetCI (object, target, dgrid, clev=0.90, high= TRUE)
```

**Arguments**

- `object`: An emaxsim object
- `target`: Target improvement from placebo
- `dgrid`: The lowest dose is found by a search over a user-specified grid of doses. If `dgrid` is a single value, it is interpreted as the number of equally-spaced doses to select from zero to the highest dose in the simulated design.
- `clev`: One-sided confidence interval level.
- `high`: When TRUE, lower bounds are computed and must be higher than the target. When FALSE, upper bounds must be less than the target.

**Value**

Returns a vector with the lowest dose meeting the criteria. If a simulated example does not have a qualifying dose, Inf is returned.

**Note**

If the grid is very large (>200), execution will slow as a large number of estimates and SEs are computed.

**Author(s)**

Neal Thomas

**See Also**

emaxsim, predict.emaxsim, targetD
targetD

Examples

## Not run:

# emaxsim changes the random number seed
nsim<-100
doselev<-c(0,5,25,50,100)
n<-c(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen.parm<-FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen.parm,modType=3)
target<-6
tD<- ( (target*ed50)/(emax-target) )
selectedDose<-targetCI(D1,target,dgrid=c(1:100)+0.5,clev=0.80,high=TRUE)

## End(Not run)

targetD

Description

The MLE (se) of the dose required to achieve a targetted improvement from placebo. The fit can be from a 3- or 4- parameter Emax model or output from function emaxalt, or an object of class emaxsimobj. The Emax model is on the logit scale for binary data.

Usage

targetD (fit,
target,
modType=4,
binary=FALSE)
Arguments

fit  Output of nls fit to a 3- or 4-parameter Emax model. The order of the parameters in the fit must be \((\log(ed50), emax, e0)\) or \((\log(ed50), \lambda, emax, e0)\). fit can also be a list with the first element the coefficient vector, and the second element the variance-covariance matrix. Alternatively, fit may be of class emaxalt or emaxsimobj, and the target dose is based on the fitted model.

target  Targeted change from placebo (positive or negative).

modType  Value is 3 or 4 for the 3 or 4-parameter Emax model output from nls with parameters in the order \((ed50, emax, e0)\) or \((ed50, \lambda, emax, e0)\). modType is ignored if fit is from emaxalt or emaxsimobj.

binary  When TRUE, the fit is assumed to be for binary data on the logistic scale. target is input as a risk difference, and transformed internally. When the fit is of class emaxalt or emaxsimobj, the binary status is taken from the object and binary is ignored.

Value

Returns a vector with two elements:

targetDose  The MLE of the dose achieving the target.

seTD  SE for target.dose

Note

Asymptotic SE computed using the delta method

Author(s)

Neal Thomas

See Also

SeEmax, emaxalt

Examples

```r
## Not run:
## emaxsim changes the random number seed
doselev<-c(0,5,25,50,100,250)
n<-c(78,81,81,81,77,80)
dose<-rep(doselev,n)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
```
update.emaxsimobj

Emaxsimobj

Description

Allows re-estimation for a data set generated by emaxsim using a different starting value. Typically used to test different starting values when nls has failed to converge.

Usage

## S3 method for class 'emaxsimobj'
update(object, new.parm, modType=object$modType,...)

Arguments

- **object**: Extracted simulation object
- **new.parm**: New starting value for Emax estimation. Must have order (ed50,Emax,e0)
- **modType**: When modType=4, the fitting begins with the 4 parameter model. If estimation fails or modType=3, the 3-parameter estimation is applied. If it fails, a best-fitting model linear in its parameters is selected.
- **...**: No other parameters currently used.
Value

A list is returned with class(emaxsimobj). It has the same format as those extracted by object[ ]

Author(s)

Neal Thomas

See Also

emaxsim

Examples

## Not run:

## emaxsim changes the random number seed
nsim<50
idmax<5
doselev<-(0,5,25,50,100)
n<-(78,81,81,81,77)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-7.967897
pop<-c(log(ed50),emax,e0)
meanlev<-emaxfun(doselev,pop)

###FixedMean is specialized constructor function for emaxsim
gen<FixedMean(n,doselev,meanlev,sdy)
D1 <- emaxsim(nsim,gen)
e49<-D1[49]

#### re-try estimation starting at the population value
e49u<- update(e49,pop)

## End(Not run)
Description

Extract Emax model variance-covariance matrix for ML estimates

Usage

## S3 method for class 'fitEmax'
vcov(object, ...)

## S3 method for class 'emaxsim'
vcov(object, ...)

Arguments

object  Output of Emax fitting and simulation functions
...
None additional inputs supported

Value

Variance-Covariance matrix for the MLE estimates of the parameters from fitEmax. The lower half of the variance-covariance matrix for the estimated parameters stored as a vector in column-major order for each emaxsim simulation. The vc matrix has 16, 9, or 4 elements depending on fitType.

Author(s)

Neal Thomas

See Also

fitEmax, emaxsim

Examples

doselev<-c(0,5,25,50,100,350)
n<-c(78,81,81,81,77,80)

### population parameters for simulation
e0<-2.465375
ed50<-67.481113
dtarget<-100
diftarget<-9.032497
emax<-solveEmax(diftarget,dtarget,log(ed50),1,e0)
sdy<-8.0
pop<-c(log(ed50),emax,e0)
dose<-rep(doselev,n)
meanlev<-emaxfun(dose,pop)
y<-rnorm(sum(n),meanlev,sdy)
testout<-fitEmax(y,dose,modType=4)
vcov(testout)
Index

* Bayes
  emaxsimB, 24
  plot.emaxsimB, 42
* Emax
  emaxsimB, 24
  plot.emaxsimB, 42
* datasets
  metaData, 37
* dose response
  fitEmaxB, 32
* hplot
  DRDensityPlot, 12
  plot.plotB, 52
  plotB, 54
  plotBdensity, 58
  plotD, 60
* mcmc
  fitEmaxB, 32
* nonlinear
  checkMonoEmax, 8
  coefEmax, 10
  DRDensityPlot, 12
  emaxalt, 14
  emaxfun, 16
  emaxsim, 20
  emaxsolve, 29
  fitEmax, 30
  FixedMean, 35
  nllogis, 39
  plot.emaxsim, 41
  plot.emaxsimobj, 45
  plot.fitEmax, 47
  plot.fitEmaxB, 50
  plot.plotB, 52
  plotB, 54
  plotBdensity, 58
  predict.emaxalt, 61
  predict.emaxsim, 63
  predict.emaxsimB, 64
  predict.emaxsimBobj, 66
  predict.emaxsimobj, 67
  predict.fitEmax, 69
  print.emaxsim, 72
  print.emaxsimB, 74
  print.emaxsimobj, 76
  RandEmax, 80
  SeEmax, 82
  sigmaEmax, 86
  startEmax, 87
  summary.emaxsim, 88
  summary.emaxsimB, 90
  summary.emaxsimobj, 91
  summary.emaxsimobj, 93
  targetBeta, 95
  targetCI, 96
  targetD, 97
  update.emaxsimobj, 99
  vcovEmax, 100
* package
  clinDR-package, 3
  checkMonoEmax, 8
  clinDR-package, 3
  coef, 86
  coef.emaxsim, 23
  coef.emaxsim(coefEmax), 10
  coef.emaxsimB, 27
  coef.emaxsimB(coefEmax), 10
  coef.fitEmax, 31
  coef.fitEmax(coefEmax), 10
  coef.fitEmaxB, 34
  coef.fitEmaxB(coefEmax), 10
  coefEmax, 10
  compileStanModels, 12
  dlogis, 17
  DoseFinding, 3
  DRDensityPlot, 12, 59
INDEX

emaxalt, 14, 61, 62, 88, 98
emaxfun, 16, 23, 26, 27, 35
emaxPrior.control, 18, 24, 32
emaxsim, 3–5, 11, 16, 20, 29, 35, 41, 46, 47, 63, 64, 67, 68, 72, 73, 76, 81, 86, 89, 90, 93, 96, 100, 101
emaxsimB, 3, 6, 7, 11, 24, 29, 42–44, 65, 66, 74, 76, 82, 85, 86, 90, 92
emaxsolve, 29
fitEmax, 3, 11, 29, 30, 34, 39, 40, 48, 69, 71, 77, 86, 94, 101
fitEmaxB, 3, 11, 25, 29, 32, 50, 52, 78, 85, 86, 94
FixedMean, 20, 24, 35, 44, 46, 81
mcmc.control, 25, 33, 36
metaData, 3, 37
Mods, 21, 25
nllogis, 31, 39
nls, 16, 31, 49, 70, 82, 88, 98
plot.emaxsim, 23, 41, 63, 73, 89, 90
plot.emaxsimB, 27, 42, 65, 74
plot.emaxsimobj, 7, 43, 92
plot.emaxsimobj, 5, 45, 76, 93
plot.fitEmax, 3, 10, 31, 47, 54, 57, 59, 61
plot.fitEmaxB, 3, 34, 50
plot.plotB, 10, 52, 57, 59
plotB, 3, 53, 54, 54, 61
plotDdensity, 13, 58
plotD, 3, 10, 54, 57, 59, 60
predict.emaxalt, 61
predict.emaxsim, 23, 62, 63, 67, 68, 96
predict.emaxsimB, 64
predict.emaxsimobj, 66
predict.emaxsimobj, 62, 67
predict.fitEmax, 31, 69
predict.fitEmaxB, 34, 71
print.emaxsim, 23, 41, 72, 89, 90
print.emaxsimB, 27, 43, 74
print.emaxsimobj, 7, 75, 92
print.emaxsimobj, 5, 47, 76, 93
printFitEmax, 77
printFitEmaxB, 78
prior.control, 24, 32, 78
RandEmax, 20, 24, 35, 44, 46, 80
rstan, 32
runShiny, 81
SeEmax, 49, 70, 82, 98
selEstan, 84
showStanModels, 85
sigma, 11
sigma.emaxsim, 23
sigma.emaxsim(sigmamax), 86
sigma.emaxsimB, 27
sigma.emaxsimB(sigmamax), 86
sigma.fitEmax(sigmamax), 86
sigma.fitEmaxB(sigmamax), 86
sigmaEmax, 86
solveDose(emaxsolve), 29
solveEmax(emaxsolve), 29
startEmax, 87
summary.emaxsim, 23, 41, 63, 67, 68, 73, 88
summary.emaxsimB, 27, 43, 65, 74, 90
summary.emaxsimobj, 91
summary.emaxsimobj, 47, 76, 93
summary.fitEmax, 94
summary.fitEmaxB, 94
targetBeta, 95
targetCI, 96
targetD, 96, 97
update.emaxsimobj, 5, 47, 99
vcov.emaxsim, 23
vcov.emaxsim(vcovEmax), 100
vcov.fitEmax(vcovEmax), 100
vcovEmax, 100