Package 'BHERM'

August 29, 2016

Type Package

Title Bayesian Hierarchical Exposure Response Modeling

Version 1.0

Date 2016-08-18

Author Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer Yaohui Zeng <yaohui-zeng@uiowa.edu>

Description BHERM is a Bayesian hierarchical model framework to incorporate PK data into dose-response model for single-agent trial. It consists a lower layer of Dose-PK model and a upper layer of PK-Response model. 5 commonly PK-Response models are implemented: Logistic, Poisson, Negative Binomial, Emax models (for count data) and Cox Proportional-Hazards model (for time-to-event data). It can model two PD responses (safety and efficacy) simultaneously and allow for different PK parameters for safety and efficacy.

License GPL (>=3)

Depends R (>= 3.0.2), Rcpp (>= 0.11.0), methods

Imports ggplot2 (>= 2.0.0), rstan (>= 2.9.0), coda, ggrepel (>= 0.5), gridExtra (>= 2.2.1), grid

LinkingTo StanHeaders (>= 2.9.0), rstan (>= 2.9.0), BH (>= 1.60.0), Rcpp (>= 0.12.0), RcppEigen

RoxygenNote 5.0.1

NeedsCompilation yes

R topics documented:

BHERM-package *Bayesian Hierarchical Exposure Response Modeling*

Description

BHERM is a Bayesian hierarchical model framework to incorporate PK data into dose-response model for single-agent trial. It consists a lower layer of Dose-PK model and a upper layer of PK-Response model. 5 commonly PK-Response models are implemented: Logistic, Poisson, Negative Binomial, Emax models (for count data) and Cox Proportional-Hazards model (for time-to-event data). It can model two PD responses (safety and efficacy) simultaneously and allow for different PK parameters for safety and efficacy.

Details

See details section in BHERM function.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

BHERM *main function for BHERM*

Description

Fit the Bayesian hierarchical exposure response model (BHERM). See detail.

Usage

```
BHERM(file, varnames,
      safe.model = c("Logistic", "Poisson", "NegBin", "Emax", "Cox", "NULL"),
      eff.model = c("Logistic", "Poisson", "NegBin", "Emax", "Cox", "NULL"),
      rescale.pk.safe = NULL, rescale.pk.eff = NULL,
      file.prior = NULL, pred.doses = NULL,
      increment.dose = 10, pred.time.max = NULL,
      increment.time = 10, pred.NPK = 3,
      chains = 4, warmup = 500, iter = 1500, adapt_delta = 0.8,
      stepsize = 0.1, max_treedepth = 10, seed, ...)
```

Arguments

file	File name for a subject-level data file containing some required variables. See details.
varnames	A character vector of variable names needed. See details.
safe.model	PK-Response model option for Safety part.
eff.model	PK-Response model option for Efficacy part.

`rescale.pk.safe`
: Rescaling factor for individual PK observations of SAF. If not provided, the PK observations are rescaled by dividing the geometric mean.

`rescale.pk.eff`
: Same as above, but for EFF part.

`file.prior`
: An R file containing prior specifications for model parameters. Depending on different models selected, the parameters are different.

`pred.doses`
: Dose levels used for prediction. The range of dose levels for prediction is defined by [`min(observed dose levels)`, `max(pred.doses)`]. The increment of dose is defined by `increment.dose` below.

`increment.dose`
: Explained as above.

`pred.time.max`
: Maximum follow-up time used to trace out survival curve. Only used for Cox models.

`increment.time`
: Increment of time for tracing out survival curve. Only used for Cox models.

`pred.NPK`
: A positive integer specifying the number of PK observations used for predicting event probability.

`chains`
: A positive integer specifying the number of Markov chains. The default is 4. Note that this is also the number of cores used.

`warmup`
: A positive integer specifying the number of warmup (aka burnin) iterations per chain. The default is 500. See `stan` for details.

`iter`
: A positive integer specifying the number of iterations for each chain (including warmup). The default is 1500.

`adapt_delta`
: A parameter that controls the sampler's behavior. The default is 0.8. See `stan` for details.

`stepsize`
: A parameter that controls the sampler's behavior. The default is 0.1. See `stan` for details.

`max_treedepth`
: A parameter that controls the sampler's behavior. The default is 10. See `stan` for details.

`seed`
: The seed for random number generation. See `stan` for details.

`...`
: Any other parameters used in `stan` can be passed. For example, number of thins `thin`, initial parameter values `init`, etc. See `stan` for details.

Details

The `BHERM` package fits the Bayesian hierarchical exposure response model (BHERM) for Exposure-Response (ER) analysis. BHERM consists of two parts: the lower layer is the Dose-PK linear model; the upper layer is the PK-Response model. There are 5 model options implemented for the PK-Response model: (1) Logistic, Poisson, Negative Binomial (NegBin), and Emax model for count data; (2) Bayesian Cox proportional-hazards model for time-to-event data. Moreover, BHERM can simultaneously accommodate for two PD responses, i.e., Safety (SAF) and Efficacy (EFF), if for count data. That is, two PK-Response models will be constructed if there are PK and response data specified for both SAF and EFF. Detailed model structure, parameters, and specifications are documented in the User Guide.

To use this package,two data files are needed. First is the raw subject-level data file (.csv file) with following variables:

- ID: Subject ID.
- TRT: Treatment code (character). For example, "A", "B", "C", "D", etc, representing different treatments.

- Dose: Dose levels (numeric). For example, 50, 100, 200, 300, etc., representing different dose levels corresponding to TRT.
- PK_SAF (PK_EFF): PK parameter for SAF (EFF). For example, AUC, Cmax, SS trough concentration, etc.
- RESP_SAF (RESP_EFF): Binary event indicator for SAF (EFF). For example, Grade 3/4 AE, tumor response status, etc.
- Time_SAF (Time_EFF): Follow-up time for SAF (EFF) response. Used by Cox model.
- CENSOR_SAF (CENSOR_EFF): Censoring indicator for SAF (EFF) event, so that "1" means "censored" and "0" means "event". Used by Cox model.

NOTE (1): the variable names in the raw data file don't need to be exactly the same as the names stated above. However, the program needs to know which variable in the raw data file corresponds to which variable listed above (by correctly specifying `varnames`).

NOTE (2): `varnames` must be specified in the following order, depending on different model specifications.

- (a) both SAF and EFF, no Cox model for PK-Response: `varnames = c("ID", "TRT", "REP_SAF", "PK_SAF", "P`
- (b) only SAF, not Cox model for PK-Response: `varnames = c("ID", "TRT", "REP_SAF", "PK_SAF")`
- (c) only EFF, not Cox model for PK-Response: `varnames = c("ID", "TRT", "RESP_EFF", "PK_EFF")`
- (d) only SAF, Cox model for PK-Response: `varnames = c("ID", "TRT","CENSOR_SAF", "Time_SAF", "PK_SAF`
- (e) only EFF, Cox model for PK-Response: `varnames = c("ID", "TRT","CENSOR_SAF", "Time_SAF", "PK_SAF`

NOTE (3): if `safe.model` (`eff.model`) is "Cox", then `eff.model` (`safe.model`) must be "NULL". That is, if the Cox model is selected, only one PK-Response model is allowed. The other PD response must not be specified.

The other file needed is an R file that contains all prior specifications for the model parameters. The file is passed to the program by specifying `file.prior`. There is a template prior file provided in the source folder (`./R/default_priors.R`) that includes all the model parameters with flat prior values by default. If `file.prior` is not specified (i.e., == "NULL" by default), then the data in `default_priors.R` file will be used. If the user wants to specify different priors, she/he can simply do following: (1) copy `default_priors.R` file into current working directory; (2) only change prior values for those model parameters needed based on the model specifications; (3) pass the copied prior file name to `file.prior`. The user doesn't need to be bothered by all other parameters in the prior file since the program can automatically discard those useless parameters based on the model specifications.

Value

A `BHERM` class object containing following variables:

`file` Same as that in Arguments Section above.

`varnames` Same as that in Arguments Section above.

`safe.model` Same as that in Arguments Section above.

`eff.model` Same as that in Arguments Section above.

`rescaled.pk.safe`
= NA if `safe.model == 'NULL'`; = geometric mean of individual PK observations of SAF part otherwise.

`rescaled.pk.eff`
= NA if `eff.model == 'NULL'`; = geometric mean of individual PK observations of EFF part otherwise.

model.case	An integer indicating the case of model specification. Internal use only. The user doesn't need to know it.
summary.data	Dose-level summary data, including summary PK data, the number of PD events and observations, the number of PK observations, etc., at each dose level.
stan.data	Data variables prepared for running stan model.
stan.fit	A S4 object of stanmodel produced by stan() function. See stan for details.
pars.model	Model parameters that have sample output from stan. Posterior summary statistics of these parameters will be saved into a text file when calling post_analysis().
pars.pred	Prediction parameters that also have sample output for prediction plots.
pars.out	Parameters that have sample output.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

post_analysis

Examples

```
## Not run:
seed <- 2016
file <- "LDK378_DLT_time_CB_pfs_final.csv"

## case 1: both SAF and EFF
#-------------------------------------------------------------
safe.model <- 'Logistic' # specify Logistic model for SAF
eff.model <- 'Poisson'   # specify Poisson model for EFF
varnames <- c("SID1A", 'TRTREG1C', 'dosemg', 'AEG3OR4',
              'AVGSSTPA', 'EFFRESP', 'AVGSSALL')
main.fit <- BHERM(file, varnames, seed = seed,
                  adapt_delta = 0.99, stepsize = 0.01,
                  max_treedepth = 12,
                  safe.model = safe.model,
                  eff.model = eff.model,
                  pred.doses = 1100
)
post_analysis(main.fit)

## case 2: safety only
#-------------------------------------------------------------
safe.model <- 'Logistic'
eff.model <- 'NULL'
varnames <- c("SID1A", 'TRTREG1C', 'dosemg', 'AEG3OR4',
              'AVGSSTPA')
safe.main.fit <- BHERM(file, varnames, seed = seed,
                       safe.model = safe.model,
                       eff.model = eff.model)

## case 3: efficacy only
#-------------------------------------------------------------
safe.model <- 'NULL'
```

```
eff.model <- 'Logistic'
varnames <- c("SID1A", 'TRTREG1C', 'dosemg', 'EFFRESP', 'AVGSSALL')
eff.main.fit <- BHERM(file, varnames, seed = seed,
                      safe.model = safe.model,
                      eff.model = eff.model)

## case 4: efficacy cox only
#-------------------------------------------------------------
varnames <- c("SID1A", 'TRTREG1C', 'dosemg', "PFS1_CS",
              "PFS1_1N", "AVGSSALL")
safe.model <- 'NULL'
eff.model <- 'Cox'

eff.cox.main.fit <- BHERM(file, varnames,
                          safe.model = safe.model, eff.model = eff.model,
                          adapt_delta = 0.99, stepsize = 0.01,
                          max_treedepth = 15,
                          seed = seed)

## case 5: Safety cox only
#-------------------------------------------------------------
varnames <- c("SID1A", 'TRTREG1C', 'dosemg', "AEG3OR4",
              "DAEV_1N", "AVGSSTPA")
safe.model <- 'Cox'
eff.model <- 'NULL'

safe.cox.main.fit <- BHERM(file, varnames,
                           safe.model = safe.model, eff.model = eff.model,
                           adapt_delta = 0.99, stepsize = 0.01,
                           max_treedepth = 15,
                           seed = seed)

## End(Not run)
```

`plot_dose_pk_EFF` *produce Dose-PK prediction plot for EFF*

Description

Produce Dose-PK prediction plot for EFF, along with a confidence band and observed PK values at each dose level.

Usage

```
plot_dose_pk_EFF(fit, conf.band = c("95%", "90%"), xlab = "Dose", ylab = "PK")
```

Arguments

`fit`	A `BHERM` class object produced by calling function `BHERM`.
`conf.band`	Show 95% or 90% confidence band in the prediciton plots.
`xlab`	The x-axis label for Dose-PK prediciton plot(s). Default is "Dose".
`ylab`	The y-axis label for Dose-PK prediciton plot(s). Default is "PK".

Details

Returned is a ggplot object. Both axies are plotted on log scale, whereas the tick marks are on linear scale. Observed PK values at each dose level are shown as dots.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, post_analysis, plot_dose_pk_SAF

---

plot_dose_pk_SAF *produce Dose-PK prediction plot for SAF*

Description

Produce Dose-PK prediction plot for SAF, along with a confidence band and observed PK values at each dose level.

Usage

```
plot_dose_pk_SAF(fit, conf.band = c("95%", "90%"), xlab = "Dose", ylab = "PK")
```

Arguments

fit	A BHERM class object produced by calling function BHERM.
conf.band	Show 95% or 90% confidence band in the prediciton plots.
xlab	The x-axis label for Dose-PK prediciton plot(s). Default is "Dose".
ylab	The y-axis label for Dose-PK prediciton plot(s). Default is "PK".

Details

Returned is a ggplot object. Both axies are plotted on log scale, whereas the tick marks are on linear scale. Observed PK values at each dose level are shown as dots.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, post_analysis, plot_dose_pk_EFF

plot_mean_survival *plot posterior mean survival curves*

Description

This function produces the posterior mean survival curves corresponding to specified dose levels in a given range of follow-up time.

Usage

```
plot_mean_survival(fit)
```

Arguments

fit	A BHERM class object produced by calling function BHERM with the PK-Response model being Cox PH model.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, post_analysis

plot_predict_EFF *produce probability prediction plot for EFF*

Description

Produce probability prediction plot for EFF, along with a confidence band, and observed event proportions (and confidence intervals).

Usage

```
plot_predict_EFF(fit, conf.band = c("95%", "90%"), xlab = "PK", ylab = "Probability")
```

Arguments

fit	A BHERM class object produced by calling function BHERM.
conf.band	Show 95% or 90% confidence band in the prediciton plots.
xlab	The x-axis label for Dose-PK prediciton plot(s). Default is "PK".
ylab	The y-axis label for Dose-PK prediciton plot(s). Default is "Probability".

Details

Returned is a ggplot object. Note there are four segments in the plots showing the observed event proportions (and confidence intervals) for each quartile of the observed PK values. Basically the observed PK values are divided into four quartiles, less than 25%, 25-50%, 50-75% and above 75%. Then in each quartile the observed event proportion is computed, and an exact binomial test is conducted to obtain the 95% (or 90%) confidence interval.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, post_analysis, plot_predict_SAF

---

plot_predict_SAF *produce probability prediction plot for SAF*

Description

Produce probability prediction plot for SAF, along with a confidence band, and observed event proportions (and confidence intervals).

Usage

```
plot_predict_SAF(fit, conf.band = c("95%", "90%"), xlab = "PK", ylab = "Probability")
```

Arguments

fit	A BHERM class object produced by calling function BHERM.
conf.band	Show 95% or 90% confidence band in the prediciton plots.
xlab	The x-axis label for Dose-PK prediciton plot(s). Default is "PK".
ylab	The y-axis label for Dose-PK prediciton plot(s). Default is "Probability".

Details

Returned is a ggplot object. Note there are four segments in the plots showing the observed event proportions (and confidence intervals) for each quartile of the observed PK values. Basically the observed PK values are divided into four quartiles, less than 25%, 25-50%, 50-75% and above 75%. Then in each quartile the observed event proportion is computed, and an exact binomial test is conducted to obtain the 95% (or 90%) confidence interval.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, post_analysis, plot_predict_EFF

post_analysis *produce all post analysis results and plots*

Description

This function saves to files posterior summary statistics, Dose-PK prediction plot(s), and event probability prediction plot(s), etc.

Usage

```
post_analysis(fit, diagnosis.plot = TRUE,
              post.summary = TRUE,
              post.pred.plot = c("pdf", "png", "NULL"),
              dose.pk.plot = c("pdf", "png", "NULL"),
              conf.band = c("95%", "90%"),
              xlab.dose.pk = "Dose", ylab.dose.pk = "PK",
              xlab.pred.prob = "PK", ylab.pred.prob = "Probability")
```

Arguments

fit	A BHERM class object produced by calling function BHERM.
diagnosis.plot	Whether to produce trace and density plots of model parameters for checking convergence. Default is TRUE.
post.summary	Whether to save posterior summmary statistics of model parameters. Default is TRUE.
post.pred.plot	Whether to generate event probability prediction plot(s) and save into a pdf file or png file. If NULL, don't generate plot(s).
dose.pk.plot	Whether to generate Dose-PK prediction plot(s) and save into a pdf file or png file. If NULL, don't generate plot(s).
conf.band	Show 95% or 90% confidence band in the prediciton plots.
xlab.dose.pk	The x-axis label for Dose-PK prediciton plot(s). Default is "Dose".
ylab.dose.pk	The y-axis label for Dose-PK prediciton plot(s). Default is "PK".
xlab.pred.prob	The x-axis label for event probability prediciton plot(s). Default is "PK".
ylab.pred.prob	The y-axis label for event probability prediciton plot(s). Default is "Probability".

Details

- If both SAF and EFF models are specified, the Dose-PK prediction plots of both SAF and EFF are produced. They are then combined into one single file. So are the event probability prediction plots.
- All the files are automatically saved to current working directory.
- If separate plots for SAF and EFF are desired, the user can call functions plot_dose_pk_SAF, plot_dose_pk_EFF for Dose-PK prediction plots, and call functions plot_predict_SAF, plot_predict_SAF for probability prediction plots.

Author(s)

Yaohui Zeng, Siyan Xu, Yu-Yun Ho

Maintainer: Yaohui Zeng <yaohui-zeng@uiowa.edu>

See Also

BHERM, plot_dose_pk_SAF, plot_dose_pk_EFF, plot_predict_SAF, plot_predict_SAF

Index