Package: blrm (via r-universe)

October 16, 2026

Title Dose Escalation Design in Phase I Oncology Trial Using Bayesian Logistic Regression Modeling

Version 1.0-2

Date 2022-07-06

Author Furong Sun <furongs@vt.edu>, Zhonggai Li

<zli@bostonbiomedical.com>

Depends R (>= 2.15.1)

Imports boot, rjags, mvtnorm, openxlsx, reshape2, shiny

Description Design dose escalation using Bayesian logistic regression modeling in Phase I oncology trial.

Maintainer Furong Sun <furongs@vt.edu>

License LGPL

NeedsCompilation no

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

Date/Publication 2022-07-07 02:40:02 UTC

RemoteUrl https://github.com/cran/blrm

RemoteRef HEAD

RemoteSha 87e8c92758f0f5b65faac7998dc533ad243b6593

Contents

blrm_mono_ss 2

Index **8**

blrm_mono_ss *Dose Escalation Design in Phase I Oncology Trial using Bayesian Logistic Regression Modeling*

Description

Provides dose escalation design in Phase I oncology trial using Bayesian Logistic Regression Modeling given prior and observed cohorts

Usage

```
blrm_mono_ss(prior, data, output_excel=FALSE, output_pdf=FALSE)
blrm_mono_ms(prior, data, output_excel=FALSE, output_pdf=FALSE)
blrm_combo_ss(prior, data, output_excel=FALSE, output_pdf=FALSE)
blrm_combo_ms(prior, data, output_excel=FALSE, output_pdf=FALSE)
```

Arguments

prior	mean, standard deviation, and correlation of parameters
data	parameters, including random number generating seeds, number of simulation samples, burn-in period, drug name, dose unit, provisional doses, reference dose, tested doses, number of patients, number of dose-limiting toxicities, category bounds, category names, escalation with overdose criterion
output_excel	output_excel = FALSE by default; if output_excel = TRUE, then the simulation output will be provided in excel file
output_pdf	output_pdf = FALSE by default; if output_pdf = TRUE, then visualization of simulation output, including interval probabilities by dose and posterior distribution of dose-limiting toxicity rate, will be provided in pdf file

Details

Model-based dose-escalation design is more flexible than traditional "3 + 3" design. Bayesian logistic regression model is a two-parameter statistical model to quantify the relationship between dose-limiting toxicity (DLT) rate and drug dose.

$$\log(\frac{\pi_d}{1-\pi_d}) = \log(\alpha) + \beta \log(\frac{d}{d^*})$$

, where $\alpha > 0$, $\beta > 0$, π_d is the probability of toxicity at dose d, and d^* is the reference dose.

For more details, see Neuenschwander, et al. (2008).

Value

prob_posterior	interval probabilities by dose
para_posterior	posterior estimates of parameters
pi_posterior	posterior estimates of dose-limiting rates

next_dose	recommended next dose
current_summary	summary of simulation outputs
cohort_all	accumulated summary of each observed cohort

Author(s)

Furong Sun <furongs@vt.edu>

References

Neuenschwander, B., Branson, M. and Gsponer, T. (2008). "Critical aspects of the Bayesian approach to phase I cancer trials", *Statistics in Medicine*, 27(13), 2420-2439. doi: 10.1002/sim.3230.

Examples

```
## mono version
# prior for log(alpha) and log(beta)
mean <- c(-3.068, 0.564)
se <- c(2.706, 0.728)
corr <- -0.917
prior <- list(mean=mean, se=se, corr=corr)

# parameters
seeds <- 1:2
nsamples <- 10000
burn_in <- 0.2
drug_name <- "DRUG-X"
dose_unit <- "mg"
prov_dose <- c(360, 480, 720, 1080, 1440)
ref_dose <- 720
category_bound <- c(0.16, 0.33)
category_name <- c("under-dosing", "targeted-toxicity", "over-dosing")
ewoc <- 0.25

# observed cohorts
dose <- 480
n_pat <- 3
dlt <- 0

# combine prior, parameters, and observed cohorts to a list
data <- list(seeds=seeds, nsamples=nsamples, burn_in=burn_in, drug_name=drug_name,
             dose_unit=dose_unit, prov_dose=prov_dose, ref_dose=ref_dose,
             dose=dose, n_pat=n_pat, dlt=dlt, category_bound=category_bound,
             category_name=category_name, ewoc=ewoc)

# ready to go!
trial <- blrm_mono_ss(prior=prior, data=data, output_excel=FALSE, output_pdf=FALSE)
prob_posterior <- trial$prob_posterior
pi_posterior <- trial$pi_posterior

# visualization
```

```
## interval probabilities by dose
cols <- c("green", "red")[((prob_posterior[3,]) > ewoc)+1] # red: stop; green: pass
yrange <- function(max.y){
    yrange.final <- ifelse(rep((1.1*max.y <= 1), 2), c(0, 1.1*max.y), c(0, max.y))
}
layout(matrix(c(1,2,3), 3, 1, byrow=TRUE))
## e.g., (0.33, 1]
barplot(prob_posterior[3,],
        xlab=paste0("dose", "(", dose_unit, ")"), ylab="probability",
        ylim=yrange(max(prob_posterior[3,])), names.arg=paste(prov_dose),
        col=cols,
        main=paste0(category_name[3], ": (", category_bound[2], ", 1]"),
        cex.main=1.3, font.main=4)
if(max(prob_posterior[3,]) >= ewoc){
   abline(h=ewoc, lty=2, col="red")
   text(x=0.4, y=1.15*ewoc, labels=paste0("EWOC=", ewoc),
        col="red", cex=1.2, font.main=4)
}else{
   text(x=0.8, y=max(prob_posterior[3,]), labels=paste0("EWOC=", ewoc),
        col="red", cex=1.2, font.main=4)
}
## e.g., (0.16, 0.33]
barplot(prob_posterior[2,],
        xlab=paste0("dose", "(", dose_unit, ")"), ylab="probability",
        ylim=yrange(max(prob_posterior[2,])), names.arg=paste(prov_dose),
        col="green",
     main=paste0(category_name[2], ": (", category_bound[1], ", ", category_bound[2], "]"),
        cex.main=1.3, font.main=4)
## e.g., (0, 0.16]
barplot(prob_posterior[1,],
        xlab=paste0("dose", "(", dose_unit, ")"), ylab="probability",
        ylim=yrange(max(prob_posterior[1,])), names.arg=paste(prov_dose),
        col="green", main=paste0(category_name[1], ": (0, ", category_bound[1], "]"),
        cex.main=1.3, font.main=4)
## add a main title to the three barplots together
mtext("Interval Probabilities by Dose", side=3, outer=TRUE, line=-2,
       at=par("usr")[1]+0.035*diff(par("usr")[1:2]), cex=1.2, font=2)

## posterior distribution of DLT rate
par(mfrow=c(1,1))
plot(prov_dose, pi_posterior[4,], type='p', pch=20,
     xlab=paste0("dose", "(", dose_unit, ")"), ylab="DLT rate",
     xlim=range(prov_dose), ylim=c(0, max(pi_posterior)),
     main="Posterior Distribution of DLT Rate", bty="n")
arrows(prov_dose, pi_posterior[3,], prov_dose, pi_posterior[5,],
       code=3, angle=90, length=0.1, lwd=1.5, col=1)
if(max(pi_posterior[5,]) >= category_bound[2]){
   abline(h=category_bound, lty=2, col=c(rgb(0,1,0,alpha=0.8), rgb(1,0,0,alpha=0.8)))
  legend("topleft", c(paste(category_bound), "median", "95 percent credible interval"),
          lty=c(2,2,NA,1), lwd=c(1,1,NA,1.5), pch=c(NA,NA,20,NA),
          col=c(rgb(0,1,0,alpha=0.8), rgb(1,0,0,alpha=0.8), 1, 1), bty="n")
}else if((max(pi_posterior[5,]) >= category_bound[1]) &&
          (max(pi_posterior[5,]) < category_bound[2])){
```

```
   abline(h=category_bound[1], lty=2, col=rgb(0,1,0,alpha=0.8))
 legend("topleft", c(paste(category_bound[1]), "median", "95 percent credible interval"),
         lty=c(2,NA,1), lwd=c(1,NA,1.5), pch=c(NA,20,NA),
         col=c(rgb(0,1,0,alpha=0.8), 1, 1), bty="n")
}else{
      legend("topleft", c("median", "95 percent credible interval"),
             lty=c(NA,1), lwd=c(NA,1.5), pch=c(20,NA), col=c(1,1), bty="n")
}



## combo version
# prior
## drug 1
mean1 <- c(-1.0989, -0.1674)
se1 <- c(1.2770, 0.5713)
corr1 <- 0.5224
prior1 <- list(mean=mean1, se=se1, corr=corr1)
## drug 2
mean2 <- c(-2.9444, 0)
se2 <- c(2, 1)
corr2 <- 0
prior2 <- list(mean=mean2, se=se2, corr=corr2)
## interaction between 2 drugs
prior3 <- list(mean=0, se=1.121)
## combine three sets of priors
prior <- list(prior1, prior2, prior3)

# parameters
seeds <- 1:2
nsamples <- 10000
burn_in <- 0.5

## drug 1
drug1_name <- "DRUG-X"
dose1_unit <- "mg"
ref_dose1 <- 15
prov_dose1 <- c(1, 2.5, 5, 10)

## drug 2
drug2_name <- "DRUG-Y"
dose2_unit <- "mg"
ref_dose2 <- 350
prov_dose2 <- c(200, 250, 300, 350)

dose1 <- 1     # tested doses for drug 1
dose2 <- 200   # tested doses for drug 2
n_pat <- 3     # number of patients at each observed cohort
dlt <- 0       # number of DLTs at each observed cohort

category_bound <- c(0.16, 0.33)
category_name <- c("under-dosing", "targeted-toxicity", "over-dosing")
ewoc <- 0.25
```

```
# combine to a list
data <- list(seeds=seeds, nsamples=nsamples, burn_in=burn_in,
             drug1_name=drug1_name, dose1_unit=dose1_unit,
             ref_dose1=ref_dose1, prov_dose1=prov_dose1,
             drug2_name=drug2_name, dose2_unit=dose2_unit,
             ref_dose2=ref_dose2, prov_dose2=prov_dose2,
             dose1=dose1, dose2=dose2, n_pat=n_pat, dlt=dlt,
             category_bound=category_bound, category_name=category_name,
             ewoc=ewoc)

# ready to go!
     trial <- blrm_combo_ss(prior=prior, data=data, output_excel=FALSE, output_pdf=FALSE)
prob_posterior <- trial$prob_posterior
  pi_posterior <- trial$pi_posterior
     next_dose <- trial$next_dose

# visualization
## Interval Probabilities by Dose: `(0.33, 1]' is the target
### data manipulation
prov_dose <- expand.grid(prov_dose1, prov_dose2)
prob_posterior_3 <- cbind(prov_dose, prob_posterior[3,])
names(prob_posterior_3) <- c("drug1", "drug2", "probability")
prob_posterior_3 <- transform(prob_posterior_3,
                              level=ifelse(probability > ewoc, 1, 0))
for(i in 1:nrow(prob_posterior_3)){
    if(as.numeric(rownames(prob_posterior_3[i,])) == as.numeric(next_dose$index)){
       prob_posterior_3[i,4] <- 2
    }
}
# convert data.frame from ``long" to ``wide"
prob_posterior_3_wide <- reshape2::dcast(prob_posterior_3[,-3],
                                  drug2 ~ drug1, value.var="level")
prob_posterior_3_wide <- prob_posterior_3_wide[,-1]
rownames(prob_posterior_3_wide) <- paste(prov_dose2)
colnames(prob_posterior_3_wide) <- paste(prov_dose1)
cols <- matrix(NA, nrow=length(prov_dose2), ncol=length(prov_dose1))
for(i in 1:nrow(prob_posterior_3_wide)){
    for(j in 1:ncol(prob_posterior_3_wide)){
        if(prob_posterior_3_wide[i,j]==0){
           cols[i,j] <- "green"
        }else if(prob_posterior_3_wide[i,j]==1){
           cols[i,j] <- "red"
        }else{
           cols[i,j] <- "blue"
        }
    }
}
### generate the plot
plot(NA, NA, type='n', xaxt='n', yaxt='n', cex.lab=1.5,
     cex.main=2,
     xlim=range(1:length(prov_dose1)),
     ylim=range(1:length(prov_dose2)),
```

```
     xlab=paste0(drug1_name, "(", dose1_unit, ")"),
     ylab=paste0(drug2_name, "(", dose2_unit, ")"),
     main="Dose combo Categorization")
abline(h=1:length(prov_dose2), v=1:length(prov_dose1),
       lty=2, lwd=1, col="gray")
axis(1, at=1:length(prov_dose1), labels=paste(prov_dose1))
axis(2, at=1:length(prov_dose2), labels=paste(prov_dose2), las=2)
# add dose combos falling within different categories
for(i in 1:length(prov_dose2)){
    for(j in 1:length(prov_dose1)){
        points(j, i, pch=19, col=cols[i,j], cex=4)
    }
}
legend("topright", c(" <= EWOC", " > EWOC", "Recommended Next Dose"),
       cex=1.1, pch=rep(19, 2), col=c("green", "red", "blue"),
       pt.cex=2, xpd=TRUE, horiz=TRUE, inset=c(0, -0.045), bty='n')

## Posterior Distribution of DLT Rate
par(mfrow=c(1,1))
labels <- apply(prov_dose, 1, paste, collapse=",")
plot(1:nrow(prov_dose), pi_posterior[4,], type="p", pch=20,
     xlab="drug combo", xaxt='n', cex.lab=1.5,
     ylab="DLT rate", ylim=c(0, max(pi_posterior)),
     main="Posterior Distribution of DLT Rate", cex.main=2.0, bty='n')
axis(1, at=1:nrow(prov_dose), labels=FALSE)
text(x=1:nrow(prov_dose), par("usr")[3]-0.03, labels=labels,
     srt=90, pos=1, xpd=TRUE, cex=0.5)
arrows(1:nrow(prov_dose), pi_posterior[3,],
       1:nrow(prov_dose), pi_posterior[5,],
       code=3, angle=90, length=0.1, lwd=1.5, col=1)
if(max(pi_posterior[5,]) >= category_bound[2]){
   abline(h=category_bound, lty=2,
          col=c(rgb(0,1,0,alpha=0.8), rgb(1,0,0,alpha=0.8)))
   legend("top",
          c(paste(category_bound), "median", "95 percent credible interval"),
          lty=c(2,2,NA,1), lwd=c(1,1,NA,1.5), pch=c(NA,NA,20,NA),
          col=c(rgb(0,1,0,alpha=0.8), rgb(1,0,0,alpha=0.8), 1, 1),
          xpd=TRUE, horiz=TRUE, inset=c(0, -0.035), bty='n')
}else if((max(pi_posterior[5,]) >= category_bound[1]) &&
         (max(pi_posterior[5,]) < category_bound[2])){
    abline(h=category_bound[1], lty=2, col=rgb(0,1,0,alpha=0.8))
   legend("top", c(paste(category_bound[1]), "median", "95 percent credible interval"),
          lty=c(2,NA,1), lwd=c(1,NA,1.5), pch=c(NA,20,NA),
          col=c(rgb(0,1,0,alpha=0.8), 1, 1), bty='n',
          xpd=TRUE, horiz=TRUE, inset=c(0, -0.035))
}else{
    legend("top", c("median", "95 percent credible interval"),
           lty=c(NA, 1), lwd=c(NA, 1.5), pch=c(20, NA), col=c(1, 1),
           xpd=TRUE, horiz=TRUE, inset=c(0, -0.035), bty='n')
}
# end of visualization
```

Index

∗ **Bayesian statistics**
blrm_mono_ss, 2
∗ **dose escalation**
blrm_mono_ss, 2
∗ **logistic regression**
blrm_mono_ss, 2
∗ **oncology trial**
blrm_mono_ss, 2

blrm_combo_ms *(*blrm_mono_ss*)*, 2
blrm_combo_ss *(*blrm_mono_ss*)*, 2
blrm_mono_ms *(*blrm_mono_ss*)*, 2
blrm_mono_ss, 2