Package ‘bmabasket’

July 22, 2025

Type Package

Title Bayesian Model Averaging for Basket Trials

Version 0.1.2

Date 2022-02-24

Maintainer Matt Psioda <matt_psioda@unc.edu>

Description An implementation of the Bayesian model averaging method of Psioda and others (2019) <doi:10.1093/biostatistics/kxz014> for basket trials. Contains a user-friendly wrapper for simulating basket trials under conditions and analyzing them with a Bayesian model averaging approach.

License GPL (>= 3)

Depends R (>= 3.5.0)

Imports Rcpp (>= 1.0.3), partitions

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

URL https://github.com/ethan-alt/bmabasket

BugReports https://github.com/ethan-alt/bmabasket/issues

RoxygenNote 7.1.1

NeedsCompilation yes

Author Matt Psioda [cre],
Ethan Alt [aut]

Repository CRAN

Date/Publication 2022-02-24 17:50:13 UTC

Contents

bma *Compute posterior model probabilities*

Description

Given data and hyperparameters, computes posterior model probabilities

Usage

```
bma(pi0, y, n, P = NULL, mu0 = 0.5, phi0 = 1, priorModelProbs = NULL, pmp0 = 1)
```

Arguments

`pi0`	scalar or vector whose elements are between 0 and 1 giving threshold for the hypothesis test. If a scalar is provided, assumes same threshold for each basket
`y`	vector of responses
`n`	vector of sample sizes
`P`	integer giving maximum number of distinct parameters; default is all possible models
`mu0`	prior mean for beta prior
`phi0`	prior dispersion for beta prior
`priorModelProbs`	(optional) vector giving prior for models. Default is proportional to `exp(pmp0 * D)`, where `D` is the number of distinct parameters in the model
`pmp0`	nonnegative scalar. Value of 0 corresponds to uniform prior across model space. Ignored if priorModelProbs is specified

Value

a list with the following structure:

bmaProbs model-averaged probabilities that each basket is larger than `pi0`

bmaMeans model-averaged posterior mean for each basket

Examples

```
## Simulate data with 3 baskets
probs <- c(0.5, 0.25, 0.25)
n <- rep(100, length(probs))
y <- rbinom(length(probs), size = n, prob = probs)
bma(0.5, y, n)
```

`bma_design` *Simulate a BMA design*

Description

Simulates a BMA design given hyperparameters

Usage

```
bma_design(
  nSims,
  nBaskets,
  maxDistinct = nBaskets,
  eRates,
  rRates,
  meanTime,
  sdTime,
  ppEffCrit,
  ppFutCrit,
  futOnly = FALSE,
  rRatesNull,
  rRatesAlt,
  minSSFut,
  minSSEff,
  minSSEnr,
  maxSSEnr,
  targSSPer,
  I0,
  mu0 = 0.5,
  phi0 = 1,
  priorModelProbs = NULL,
  pmp0 = 1
)
```

Arguments

`nSims`	number of simulation studies to be performed
`nBaskets`	number of baskets
`maxDistinct`	integer between 1 and `nBaskets` giving number of distinct model probabilities to use. Defaults to `nBaskets`. It is recommended to call `numModels` to ensure that computation is tractable.
`eRates`	scalar or vector of Poisson process rates for each basket
`rRates`	scalar or vector of true response rates for each basket
`meanTime`	mean parameter for time to outcome ascertainment
`sdTime`	standard deviation parameter for time to outcome ascertainment

ppEffCrit	scalar or vector giving basket-specific posterior probability threshold for activity (i.e., efficacy).
ppFutCrit	scalar or vector giving basket-specific posterior probability threshold for futility
futOnly	logical giving whether design allows only for futility stopping (TRUE = futility only, FALSE = both futility and efficacy)
rRatesNull	scalar or vector of basket-specific null hypothesis values (for efficacy determination)
rRatesAlt	scalar or vector of basket-specific hypothesized alternative values (for futility determination)
minSSFut	minimum number of subjects in basket to assess futility
minSSEff	minimum number of subjects in basket to assess activity
minSSEnr	matrix giving minimum number of new subjects per basket before next analysis (each row is an interim analysis, each column is a basket)
maxSSEnr	matrix giving maximum number of new subjects per basket before next analysis (each row is an interim analysis, each column is a basket)
targSSPer	scalar or vector giving target sample size increment for each basket
I0	maximum number of analyses
mu0	prior mean for the response probabilities
phi0	prior dispersion response probabilities
priorModelProbs	vector giving prior probabilities for models. Default is prior of each model is proportional exp(pmp0 * D) where D is the number of distinct parameters in the model
pmp0	scalar giving power for priorModelProbs. If pmp0==0, a uniform prior is used for model probabilities. Defaults to 1. Ignored if priorModelProbs is not NULL

Value

a nested list giving results of the simulation with the following structure:

- **hypothesis.testing** - hypothesis testing information
  - **rr** - basket-specific null hypothesis rejection rate
  - **fw.fpr** - family-wise false positive rate (across all inactive baskets)
  - **nerr** - average number of false null hypothesis rejections
  - **fut** - basket-specific probability of futility stopping
- **sample.size** - trial sample size information
  - **basket.ave** - basket-specific expected sample size
  - **basket.med** - basket-specific median sample size
  - **basket.min** - basket-specific minimum sample size
  - **basket.max** - basket-specific maximum sample size
  - **overall.ave** - expected overall sample size
- **point.estimation** - point estimation information

- **PM.ave** - basket-specific average posterior mean
- **SP.ave** - basket-specific average sample proportion
- **PP.ave** - basket-specific average posterior probability
- **bias** - basket-specific bias of the posterior mean
- **mse** - basket-specific MSE of the posterior mean

• **early.stopping** - early stopping information

- **interim.stop.prob** - probability of trial stoppage by interim
- **baskets.continuing.ave** - average number of baskets continuing past interim

Examples

```
## SIMULATE DATA AND SET SIMULATION PARAMS
nSims      <- 100            ## would be much more in practice
meanTime   <- 0.01
sdTime     <- 0.0000000001
mu0        <- 0.45
phi0       <- 1.00
ppEffCrit  <- 0.985
ppFutCrit  <- 0.2750
pmp0       <- 2
n1         <- 7
n2         <- 16
targSSPer  <- c(n1, n2)
nInterim   <- 2
futOnly    <- 1
K0         <- 5
row        <- 0
mss        <- 4
minSSFut   <- mss  ## minimum number of subjects in basket to assess futility using BMA
minSSEff   <- mss  ## minimum number of subjects in basket to assess activity using BMA
rTarg      <- 0.45
rNull      <- 0.15
rRatesMod  <- matrix(rNull,(K0+1)+3,K0)
rRatesNull <- rep(rNull,K0)
rRatesMid  <- rep(rTarg,K0)
eRatesMod  <- rep(1, K0)

## min and max #' of new subjects per basket before next analysis (each row is interim)
minSSEnr <- matrix(rep(mss, K0), nrow=nInterim ,ncol=K0, byrow=TRUE)
maxSSEnr <- matrix(rep(100, K0), nrow=nInterim, ncol=K0, byrow=TRUE)

## construct matrix of rates
for (i in 1:K0)
{
  rRatesMod[(i+1):(K0+1),i]= rTarg
}
rRatesMod[(K0+2),] <- c(0.05,0.15,0.25,0.35,0.45)
rRatesMod[(K0+3),] <- c(0.15,0.30,0.30,0.30,0.45)
rRatesMod[(K0+4),] <- c(0.15,0.15,0.30,0.30,0.30)

## conduct simulation of trial data and analysis
```

```
x <- bma_design(
  nSims, K0, K0, eRatesMod, rRatesMod[i+1,], meanTime, sdTime,
  ppEffCrit, ppFutCrit, as.logical(futOnly), rRatesNull, rRatesMid,
  minSSFut, minSSEff, minSSEnr, maxSSEnr, targSSPer, nInterim, mu0,
  phi0, priorModelProbs = NULL, pmp0 = pmp0
)
```

`numModels` *Compute number of models*

Description

Given a basket size and maximal number of distinct response rates, compute the number of possible models

Usage

```
numModels(K, P)
```

Arguments

K	positive integer giving number of baskets
P	positive integer giving maximal number of distinct rates

Value

integer giving number of possible models

Examples

```
numModels(10, 10)
```

Index