Package 'BasketTrial'

June 18, 2025

Type Package

Title Bayesian Basket Trial Design and Analysis

Version 0.1.0

Date 2025-06-16

Description Provides tools for Bayesian basket trial design and analysis using a novel three-component local power prior framework with global borrowing control, pairwise similarity assessment and a borrowing threshold. Supports simulation-based evaluation of operating characteristics and comparison with other methods. Applicable to both equal and unequal sample size settings in early-phase oncology trials. For more details see Zhou et al. (2023) <doi:10.48550/arXiv.2312.15352>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Haiming Zhou [aut, cre, cph],
Philip He [aut]

Maintainer Haiming Zhou <haiming2019@gmail.com>

Repository CRAN

Date/Publication 2025-06-18 11:50:18 UTC

Contents

generate.data *Generate Data for A Basket Trial Design*

Description

Generate Data for A Basket Trial Design

Usage

```
generate.data(N, ORRs, ntrial = 10000, seed = 987897)
```

Arguments

N	a matrix with dim=c(B, stage), where B is the number of baskets and stage is the # of analyses (interim+final)
ORRs	a matrix with dim=c(nS, B), where nS is the number of trial scenarios for objective response rates.
ntrial	the total number of trials simulated.
seed	random seed for reproducibility.

Value

It returns a list including data, N, and ORRs, where data is an array with dim=c(nS, ntrial, B, stage).

Examples

```
N <- rbind(
c(10, 25),
c(10, 25),
c(10, 25)) # interim sample size and total sample size for each indication
scenarios <- rbind( c(0.15, 0.15, 0.15), c(0.3, 0.3, 0.3) )
res <- generate.data(N = N, ORRs = scenarios, ntrial = 20, seed = 2024)
```

get.Q.bwer *Get Efficacy Cutoff based on Basket-Wise Error Rate (BWER) Control*

Description

Get Efficacy Cutoff based on Basket-Wise Error Rate (BWER) Control

Usage

```
get.Q.bwer(object, alpha = 0.1, digits = 3, Qclust = NULL)
```

Arguments

object	returned by post.infer.
alpha	basket-wise type I error control.
digits	number of digits in the cutoffs.
Qclust	NULL means all cutoffs are different; If there are B=5 baskets and Qclust=(1,1,2,2,2), it means cutoffs for the first two baskets will be the same and another cutoff will be tuned separately for baskets 3-5.

Value

It returns the efficacy cutoffs.

Examples

```
N <- rbind(
c(10, 25),
c(10, 25),
c(10, 25),
c(10, 25),
c(10, 25)) # interim sample size and total sample size for each indication
scenarios <- rbind( c(0.15, 0.15, 0.15, 0.15, 0.15), c(0.3, 0.3, 0.3, 0.3, 0.3) )
res <- generate.data(N = N, ORRs = scenarios, ntrial = 1000, seed = 343809)
post <- post.infer(res, pnull = rep(0.15,5), stopbounds = cbind(c(1,1,1,1,1)),
ModelFit = "localPP", method = "PEB", a = 2, delta = 0.3)
(Q <- get.Q.bwer(post, alpha = 0.1, digits = 3, Qclust = rep(1, 5)))
Qmat <- array(NA, dim = dim(post$postprob))
for(i in 1:5) Qmat[,,i] <- Q[i]
apply(post$postprob>Qmat, c(1,3), mean)
```

---

get.weighted.power *Get Weighted Type I Error and Power.*

Description

Get weighted type I error (WE) and power(WP) cross all scenarios. including family wise (fwer) or trial wise (twer) or false discovery rate (fdr).

Usage

```
get.weighted.power(object, Q, s0 = 100, s1 = 0)
```

Arguments

object	returned by post.infer.
Q	a vector of length B for the efficacy cutoff in each basket.
s0	Setting s0=100 the weighted power reduces to type I error under global null. Please use this default.
s1	Setting s1=0 gives equal weight for calculating weighted power across scenarios. Please use this default.

Value

It returns a list with `error.tw` for average basket-wise type I error rate (BWER) under global null, `bwer` for BWERs for all null baskets, `power.cdr` for average true positive rate (TPR) across scenarios except global null, `power.ccr` for average correct classification rate (CCR) across scenarios except global null.

Examples

```
N <- rbind(
c(10, 25),
c(10, 25),
c(10, 25),
c(10, 25),
c(10, 25)) # interim sample size and total sample size for each indication
scenarios <- rbind( c(0.15, 0.15, 0.15, 0.15, 0.15), c(0.3, 0.3, 0.3, 0.3, 0.3) )
res <- generate.data(N = N, ORRs = scenarios, ntrial = 1000, seed = 343809)
post <- post.infer(res, pnull = rep(0.15,5), stopbounds = cbind(c(1,1,1,1,1)),
ModelFit = "localPP", method = "PEB", a = 2, delta = 0.3)
(Q <- get.Q.bwer(post, alpha = 0.1, digits = 3, Qclust = rep(1, 5)))
(powers <- get.weighted.power(object = post, Q = Q))
```

Independent *Independent Beta Prior for Bayesian Basket Trial*

Description

Independent Beta Prior for Bayesian Basket Trial

Usage

```
Independent(nDat, yDat, be.a0 = NULL, be.b0 = NULL)
```

Arguments

`nDat`	a vector of length B for the sample size in each basket.
`yDat`	a vector of length B for the number of responses in each basket.
`be.a0`	a vector of length B for beta prior parameter a0 in each basket.
`be.b0`	a vector of length B for beta prior parameter b0 in each basket.

Value

It returns a list including the posterior beta parameters.

Examples

```
Independent(nDat = c(25, 25, 25, 25, 25), yDat = c(2,9,11,13,20),
be.a0 = rep(0.5, 5), be.b0 = rep(0.5, 5))
```

JSD *JSD (Fujikawa et al., 2020) Method for Bayesian Basket Trial*

Description

JSD (Fujikawa et al., 2020) Method for Bayesian Basket Trial

Usage

```
JSD(nDat, yDat, be.a0 = NULL, be.b0 = NULL, epsilon = 2, tau = 0.3)
```

Arguments

nDat	a vector of length B for the sample size in each basket.
yDat	a vector of length B for the number of responses in each basket.
be.a0	a vector of length B for beta prior parameter a0 in each basket.
be.b0	a vector of length B for beta prior parameter b0 in each basket.
epsilon	the global control parameter in the JSD model.
tau	the threshold parameter in the JSD model.

Value

It returns a list including the posterior beta parameters and similarity matrix.

References

Fujikawa, K., Teramukai, S., Yokota, I., & Daimon, T. (2020). A Bayesian basket trial design that borrows information across strata based on the similarity between the posterior distributions of the response probability. Biometrical Journal, 62(2), 330-338.

Examples

```
JSD(nDat = c(25, 25, 25, 25, 25), yDat = c(2,9,11,13,20))
```

localPP *Local Power Prior for Bayesian Basket Trial*

Description

Local Power Prior for Bayesian Basket Trial

Usage

```
localPP(
  nDat,
  yDat,
  be.a0 = NULL,
  be.b0 = NULL,
  a = 1,
  delta = 0.4,
  method = "PEB",
  symmetry = FALSE
)
```

Arguments

nDat	a vector of length B for the sample size in each basket.
yDat	a vector of length B for the number of responses in each basket.
be.a0	a vector of length B for beta prior parameter a0 in each basket.
be.b0	a vector of length B for beta prior parameter b0 in each basket.
a	the global control parameter in the local PP 3-component framework.
delta	the threshold parameter in the local PP 3-component framework.
method	either PEB for the pairwise empirical Bayes or GEB for the global empirical Bayes.
symmetry	logical variable to indicate whether the similarity matrix will be set to be symmetric; default is FALSE.

Value

It returns a list including the posterior beta parameters and similarity matrix.

References

Zhou, H., Shen, R., Wu, S., & He, P. (2023). A Bayesian Basket Trial Design Using Local Power Prior. arXiv preprint arXiv:2312.15352.

Examples

```
localPP(nDat = c(25, 25, 25, 25, 25), yDat = c(2,9,11,13,20),
be.a0 = rep(0.5, 5), be.b0 = rep(0.5, 5), a = 4, delta = 1, method = "PEB")
```

post.infer *Posterior Inference for Simulated Basket Trial Data*

Description

It generates posterior probabilities P(p_j > pnull) after all interim analysis and calculates rates for early stopping, number of patients and estimated ORR.

Usage

```
post.infer(
  object,
  pnull,
  stopbounds = NULL,
  clusterk = NULL,
  nperclust = NULL,
  beta.a0 = pnull,
  beta.b0 = 1 - pnull,
  seed = 987897,
  ModelFit,
  ...
)
```

Arguments

`object`	returned from generate.data.
`pnull`	B by 1 vector of null response rates, where B is the number of baskets.
`stopbounds`	B by (stage-1) matrix: stopping boundaries for each basket at each interim.
`clusterk`	only needed for parallel computing.
`nperclust`	only needed for parallel computing.
`beta.a0`	a vector of length B for beta prior parameter a0 in each basket.
`beta.b0`	a vector of length B for beta prior parameter b0 in each basket.
`seed`	random seed for reproducibility.
`ModelFit`	the method function, e.g., `localPP`, `JSD`, and other user defined methods.
`...`	additional arguments passed to the method function defined by `ModelFit`.

Value

It returns a list including `data`, `N`, and `ORRs`, where `data` is an array with `dim=c(nS, ntrial, B, stage)`.

Examples

```
N <- rbind(
c(10, 25),
c(10, 25),
c(10, 25)) # interim sample size and total sample size for each indication
scenarios <- rbind( c(0.15, 0.15, 0.15), c(0.3, 0.3, 0.3) )
res <- generate.data(N = N, ORRs = scenarios, ntrial = 20, seed = 2024)
post <- post.infer(res, pnull = rep(0.15,3), stopbounds = cbind(c(1,1,1)),
ModelFit = "localPP", method = "PEB")
apply(post$earlystop, c(1,3), mean) # early stopping for each basket in each scenario
apply(post$npts, c(1,3), mean) # average number of pts for each basket in each scenario
apply(post$est, c(1,3), mean) # average ORR estimate for each basket in each scenario
```

Index