Package 'basksim'

July 22, 2025

Type Package

Title Simulation-Based Calculation of Basket Trial Operating Characteristics

Version 1.0.0

Description Provides a unified syntax for the simulation-based comparison of different single-stage basket trial designs with a binary endpoint and equal sample sizes in all baskets. Methods include the designs by Baumann et al. (2024) <doi:10.48550/arXiv.2309.06988>, Fujikawa et al. (2020) <doi:10.1002/bimj.201800404>, Berry et al. (2020) <doi:10.1177/1740774513497539>, Neuenschwander et al. (2016) <doi:10.1002/pst.1730> and Psioda et al. (2021) <doi:10.1093/biostatistics/kxz014>. For the latter three designs, the functions are mostly wrappers for functions provided by the packages 'bhmbasket' and 'bmabasket'.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports arrangements, bhmbasket, bmabasket, doFuture, extraDistr, foreach, HDInterval, progressr

Suggests covr, testthat (>= 3.0.0)

Config/testthat/edition 3

URL https://github.com/lbau7/basksim

BugReports https://github.com/lbau7/basksim/issues

NeedsCompilation no

Author Lukas Baumann [aut, cre] (ORCID: <https://orcid.org/0000-0001-7931-7470>), Lukas Sauer [ctb] (ORIC: 0000-0002-1340-9994)

Maintainer Lukas Baumann <baumann@imbi.uni-heidelberg.de>

Repository CRAN

Date/Publication 2024-04-12 15:20:14 UTC

Contents

adjust_lambda *Adjust Lambda*

Description

Adjust Lambda

Usage

```
adjust_lambda(design, ...)
```

Arguments

design	An object created with one of the setup functions.
...	Further arguments.

Details

The default method for adjust_lambda uses a combination of uniroot and grid search and calls toer in every iteration. For methods implemented in the bhmbasket package there are separate methods that are computationally more efficient.

Value

A list containing the greatest estimated value for lambda with prec_digits decimal places which controls the family wise error rate at level alpha (one-sided) and the estimated family wise error rate for the estimated lambda.

Examples

```
design <- setup_cpp(k = 3, p0 = 0.2)
adjust_lambda(design = design, n = 20, alpha = 0.05,
  design_params = list(tune_a = 1, tune_b = 1), iter = 1000)
```

adjust_lambda.bhm *Adjust Lambda for the BHM Design*

Description

Adjust Lambda for the BHM Design

Usage

```
## S3 method for class 'bhm'
adjust_lambda(
  design,
  n,
  p1 = NULL,
  alpha = 0.05,
  design_params = list(),
  iter = 1000,
  n_mcmc = 10000,
  prec_digits = 3,
  data = NULL,
  ...
)
```

Arguments

design	An object created with one of the setup functions.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
alpha	The one-sided significance level.
design_params	A list of params that is specific to the class of design.
iter	The number of iterations in the simulation. Is ignored if data is specified.
n_mcmc	Number of MCMC samples.
prec_digits	Number of decimal places that are considered when adjusting lambda.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the greatest estimated value for lambda with prec_digits decimal places which controls the family wise error rate at level alpha (one-sided) and the estimated family wise error rate for the estimated lambda.

Examples

```
design <- setup_bhm(k = 3, p0 = 0.2, p_target = 0.5)
adjust_lambda(design = design, n = 15,
  design_params = list(tau_scale = 1), iter = 100, n_mcmc = 5000)
```

`adjust_lambda.default` *Adjust Lambda*

Description

Adjust Lambda

Usage

```
## Default S3 method:
adjust_lambda(
  design,
  n,
  p1 = NULL,
  alpha = 0.05,
  design_params = list(),
  iter = 1000,
  prec_digits = 3,
  data = NULL,
  ...
)
```

Arguments

`design`	An object created with one of the `setup` functions.
`n`	The sample size per basket.
`p1`	Probabilities under the alternative hypothesis. If `NULL` then the type 1 error rate under the global null hypothesis is calculated.
`alpha`	The one-sided significance level.
`design_params`	A list of params that is specific to the class of `design`.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`prec_digits`	Number of decimal places that are considered when adjusting lambda.
`data`	A data matrix with k column with the number of responses for each basket. Has to be generated with `get_data`. If `data` is used, then `iter` is ignored.
`...`	Further arguments.

Details

It is recommended to use `data` and then use the same simulated data set for all further calculations. If `data = NULL` then new data is generated in each step of the algorithm, so `lambda` doesn't necessarily protect the family wise error rate for different simulated data due to Monte Carlo simulation error.

Value

A list containing the greatest estimated value for `lambda` with `prec_digits` decimal places which controls the family wise error rate at level `alpha` (one-sided) and the estimated family wise error rate for the estimated `lambda`.

Examples

```
# Example for a basket trial with Fujikawa's Design
design <- setup_fujikawa(k = 3, p0 = 0.2)
adjust_lambda(design = design, n = 20, alpha = 0.05,
  design_params = list(epsilon = 2, tau = 0), iter = 1000)
```

`adjust_lambda.exnex` *Adjust Lambda for the EXNEX Design*

Description

Adjust Lambda for the EXNEX Design

Usage

```
## S3 method for class 'exnex'
adjust_lambda(
  design,
  n,
  p1 = NULL,
  alpha = 0.05,
  design_params = list(),
  iter = 1000,
  n_mcmc = 10000,
  prec_digits = 3,
  data = NULL,
  ...
)
```

Arguments

`design`	An object created with one of the `setup` functions.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`alpha`	The one-sided significance level.
`design_params`	A list of params that is specific to the class of `design`.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`n_mcmc`	Number of MCMC samples.
`prec_digits`	Number of decimal places that are considered when adjusting lambda.

data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the greatest estimated value for lambda with prec_digits decimal places which controls the family wise error rate at level alpha (one-sided) and the estimated family wise error rate for the estimated lambda.

Examples

```
design <- setup_exnex(k = 3, p0 = 0.2)
adjust_lambda(design = design, n = 15,
  design_params = list(tau_scale = 1, w = 0.5), iter = 100, n_mcmc = 5000)
```

---

ecd *Calculate the Expected Number of Correct Decisions for a Basket Trial Design*

Description

Calculate the Expected Number of Correct Decisions for a Basket Trial Design

Usage

```
ecd(
  design,
  n,
  p1,
  lambda,
  design_params = list(),
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object created with one of the setup functions.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
design_params	A list of params that is specific to the class of design.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A numeric value.

Examples

```
# Example for a basket trial with Fujikawa's Design
design <- setup_fujikawa(k = 3, p0 = 0.2)
ecd(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  design_params = list(epsilon = 2, tau = 0), iter = 1000)
```

---

get_data *Simulate Data Based on a Binomial Distribution*

Description

Simulate Data Based on a Binomial Distribution

Usage

```
get_data(k, n, p, iter, type = c("matrix", "bhmbasket"))
```

Arguments

k	The number of baskets.
n	The sample size per basket.
p	Probabilities used to simulate the data
iter	The number of iterations in the simulation. Is ignored if data is specified.
type	Type of output. Use bhmbasket for the BHM and EXNED design and matrix for everything else.

Details

For type = "bhmbasket" this is simply a wraper for bhmbasket::simulateScenarios.

Value

If type = "matrix" then a matrix is returned, if type = "bhmbasket" then an element with class scenario_list.

Examples

```
get_data(k = 3, n = 20, p = c(0.2, 0.2, 0.5), iter = 1000)
```

get_details *Get Details of a Basket Trial Simulation*

Description

Get Details of a Basket Trial Simulation

Usage

```
get_details(design, ...)
```

Arguments

design	An object created with one of the setup functions.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors of all baskets and the family-wise error rate. For some methods the mean limits of HDI intervals are also returned.

Examples

```
# Example for a basket trial with Fujikawa's Design
design <- setup_fujikawa(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  epsilon = 2, tau = 0, iter = 100)
```

get_details.bhm *Get Details of a BHM Basket Trial Simulation*

Description

Get Details of a BHM Basket Trial Simulation

Usage

```
## S3 method for class 'bhm'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  tau_scale,
  iter = 1000,
```

```
  n_mcmc = 10000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class bhm.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
tau_scale	Standard deviation of the half normal prior distribution for the variance of the thetas.
iter	The number of iterations in the simulation. Is ignored if data is specified.
n_mcmc	Number of MCMC samples.
data	An object of class scenario_list as returned by the function bhmbasket::simulateScenarios.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_bhm(k = 3, p0 = 0.2, p_target = 0.5)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5),
  lambda = 0.95, tau_scale = 1, iter = 100)
```

---

get_details.bma *Get Details of a BMA Basket Trial Simulation*

Description

Get Details of a BMA Basket Trial Simulation

Usage

```
## S3 method for class 'bma'
get_details(design, n, p1 = NULL, lambda, pmp0, iter = 1000, data = NULL, ...)
```

Arguments

design	An object of class bma.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
pmp0	Power parameter that is used to compute prior probabilities. See bma for details.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, and mean squared errors for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_bma(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = 0.5, lambda = 0.95, pmp0 = 1,
  iter = 100)
```

---

get_details.cpp *Get Details of a Basket Trial Simulation with the Calibrated Power Prior Design*

Description

Get Details of a Basket Trial Simulation with the Calibrated Power Prior Design

Usage

```
## S3 method for class 'cpp'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  tune_a,
  tune_b,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class cpp.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
tune_a	First tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
tune_b	Second tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_cpp(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tune_a = 1, tune_b = 1, iter = 100)
```

---

get_details.cppglobal *Get Details of a Basket Trial Simulation with the Global Calibrated Power Prior Design*

Description

Get Details of a Basket Trial Simulation with the Global Calibrated Power Prior Design

Usage

```
## S3 method for class 'cppglobal'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  tune_a,
```

```
  tune_b,
  epsilon,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class cppgen.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
tune_a	First tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
tune_b	Second tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
epsilon	Tuning parameter that determines the amount of borrowing based on overall heterogeneity.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_cppglobal(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tune_a = 1, tune_b = 1, epsilon = 2, iter = 100)
```

---

get_details.exnex *Get Details of a Basket Trial Simulation with the EXNEX Design*

Description

Get Details of a Basket Trial Simulation with the EXNEX Design

Usage

```
## S3 method for class 'exnex'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  tau_scale,
  w,
  iter = 1000,
  n_mcmc = 10000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class exnex.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
tau_scale	Standard deviation of the half normal prior exchangeability distribution for the variance of the thetas.
w	Fixed prior weight for the exchangeability part of the model.
iter	The number of iterations in the simulation. Is ignored if data is specified.
n_mcmc	Number of MCMC samples.
data	An object of class scenario_list as returned by the function bhmbasket::simulateScenarios.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_exnex(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tau_scale = 1, w = 0.5, iter = 100)
```

`get_details.fujikawa` *Get Details of a Basket Trial Simulation with Fujikawa's Design*

Description

Get Details of a Basket Trial Simulation with Fujikawa's Design

Usage

```
## S3 method for class 'fujikawa'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  epsilon,
  tau,
  logbase = 2,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

`design`	An object of class `fujikawa`.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`lambda`	The posterior probability threshold.
`level`	Level of the credibility intervals.
`epsilon`	Tuning parameter that determines the amount of borrowing. See `setup_fujikawa`).
`tau`	Tuning parameter that determines how similar the baskets have to be that information is shared. See `setup_fujikawa`).
`logbase`	Tuning parameter. The base of the logarithm that is used to calculate the Jensen-Shannon divergence.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`data`	A data matrix with k column with the number of responses for each basket. Has to be generated with `get_data`. If `data` is used, then `iter` is ignored.
`...`	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate and the experiment-wise power.

Examples

```
design <- setup_fujikawa(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  epsilon = 2, tau = 0, iter = 100)
```

---

get_details.jsdglobal *Get Details of a Basket Trial Simulation with the Power Prior Design Based on Global JSD Weights*

Description

Get Details of a Basket Trial Simulation with the Power Prior Design Based on Global JSD Weights

Usage

```
## S3 method for class 'jsdglobal'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  eps_pair,
  tau = 0,
  eps_all,
  logbase = 2,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class jsdgen.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
eps_pair	Tuning parameter that determines the amount of borrowing based on pairwise similarity.
tau	Tuning parameter that determines how similar the baskets have to be that information is shared.
eps_all	Tuning parameter that determines the amount of borrowing based on overall heterogeneity.

logbase	Tuning parameter. The base of the logarithm that is used to calculate the Jensen-Shannon divergence.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_jsdglobal(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  eps_pair = 2, eps_all = 2, iter = 100)
```

---

get_details.mml *Get Details of a Basket Trial Simulation with the MML Design*

Description

Get Details of a Basket Trial Simulation with the MML Design

Usage

```
## S3 method for class 'mml'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class cpp.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.

iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_mml(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tune_a = 1, tune_b = 1, iter = 100)
```

---

get_details.mmlglobal *Get Details of a Basket Trial Simulation with the Global MML Design*

Description

Get Details of a Basket Trial Simulation with the Global MML Design

Usage

```
## S3 method for class 'mmlglobal'
get_details(
  design,
  n,
  p1 = NULL,
  lambda,
  level = 0.95,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class mmlglobal.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
level	Level of the credibility intervals.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A list containing the rejection probabilites, posterior means, mean squared errors and mean limits of HDI intervals for all baskets as well as the family-wise error rate.

Examples

```
design <- setup_mmlglobal(k = 3, p0 = 0.2)
get_details(design = design, n = 20, p1 = 0.5, lambda = 0.95, iter = 100)
```

---

get_results *Get Results for Simulation of Basket Trial Designs*

Description

Get Results for Simulation of Basket Trial Designs

Usage

```
get_results(design, ...)
```

Arguments

design	An object created with one of the setup functions.
...	Further arguments.

Value

A matrix of results with iter rows. A 0 means, that the null hypothesis that the response probability exceeds p0 was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
# Example for a basket trial with Fujikawa's Design
design <- setup_fujikawa(k = 3, p0 = 0.2)
get_results(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  epsilon = 2, tau = 0, iter = 100)
```

get_results.bhm *Get Results for Simulation of a Basket Trial with the BHM Design*

Description

Get Results for Simulation of a Basket Trial with the BHM Design

Usage

```
## S3 method for class 'bhm'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  tau_scale,
  iter = 1000,
  n_mcmc = 10000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class bhm.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
tau_scale	Standard deviation of the half normal prior distribution for the variance of the thetas.
iter	The number of iterations in the simulation. Is ignored if data is specified.
n_mcmc	Number of MCMC samples.
data	An object of class scenario_list as returned by the function bhmbasket::simulateScenarios.
...	Further arguments.

Value

A matrix of results with iter rows. A 0 means, that the null hypothesis that the response probability exceeds p0 was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_bhm(k = 3, p0 = 0.2, p_target = 0.5)
get_results(design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tau_scale = 1, iter = 100)
```

get_results.bma *Get Results for Simulation of a Basket Trial with the BMA Design*

Description

Get Results for Simulation of a Basket Trial with the BMA Design

Usage

```
## S3 method for class 'bma'
get_results(design, n, p1 = NULL, lambda, pmp0, iter = 1000, data = NULL, ...)
```

Arguments

design	An object of class bma.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
pmp0	Power parameter that is used to compute prior probabilities. See bma for details.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A matrix of results with iter rows. A 0 means, that the null hypothesis that the response probability exceeds p0 was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_bma(k = 3, p0 = 0.2)
get_results(design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  pmp0 = 1, iter = 100)
```

`get_results.cpp` *Get Results for Simulation of a Basket Trial with a Calibrated Power Prior Design*

Description

Get Results for Simulation of a Basket Trial with a Calibrated Power Prior Design

Usage

```
## S3 method for class 'cpp'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  tune_a,
  tune_b,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

`design`	An object of class `cpp`.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`lambda`	The posterior probability threshold.
`tune_a`	First tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
`tune_b`	Second tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`data`	A data matrix with k column with the number of responses for each basket. Has to be generated with `get_data`. If `data` is used, then `iter` is ignored.
`...`	Further arguments.

Value

A matrix of results with `iter` rows. A 0 means, that the null hypothesis that the response probability exceeds `p0` was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_cpp(k = 3, p0 = 0.2)
get_results(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tune_a = 1, tune_b = 1, iter = 100)
```

---

get_results.cppglobal *Get Results for Simulation of a Basket Trial with a Global Calibrated Power Prior Design*

Description

Get Results for Simulation of a Basket Trial with a Global Calibrated Power Prior Design

Usage

```
## S3 method for class 'cppglobal'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  tune_a,
  tune_b,
  epsilon,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class cppgen.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
tune_a	First tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
tune_b	Second tuning parameter that determines the amount of borrowing based on pairwise similarity between baskets.
epsilon	Tuning parameter that determines the amount of borrowing based on overall heterogeneity.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A matrix of results with `iter` rows. A 0 means, that the null hypothesis that the response probability exceeds `p0` was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_cppglobal(k = 3, p0 = 0.2)
get_results(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tune_a = 1, tune_b = 1, epsilon = 2, iter = 100)
```

`get_results.exnex` *Get Results for Simulation of a Basket Trial with the EXNEX Design*

Description

Get Results for Simulation of a Basket Trial with the EXNEX Design

Usage

```
## S3 method for class 'exnex'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  tau_scale,
  w,
  iter = 1000,
  n_mcmc = 10000,
  data = NULL,
  ...
)
```

Arguments

`design`	An object of class `exnex`.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`lambda`	The posterior probability threshold.
`tau_scale`	Standard deviation of the half normal prior exchangeability distribution for the variance of the thetas.
`w`	Fixed prior weight for the exchangeability part of the model.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`n_mcmc`	Number of MCMC samples.
`data`	An object of class `scenario_list` as returned by the function `bhmbasket::simulateScenarios`.
`...`	Further arguments.

Value

A matrix of results with `iter` rows. A 0 means, that the null hypothesis that the response probability exceeds `p0` was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_exnex(k = 3, p0 = 0.2)
get_results(design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  tau_scale = 1, w = 0.5, iter = 100)
```

`get_results.fujikawa` *Get Results for Simulation of a Basket Trial with Fujikawa's Design*

Description

Get Results for Simulation of a Basket Trial with Fujikawa's Design

Usage

```
## S3 method for class 'fujikawa'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  epsilon,
  tau,
  logbase = 2,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

`design`	An object of class `fujikawa`.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`lambda`	The posterior probability threshold.
`epsilon`	Tuning parameter that determines the amount of borrowing. See `setup_fujikawa`).
`tau`	Tuning parameter that determines how similar the baskets have to be that information is shared. See `setup_fujikawa`).
`logbase`	Tuning parameter. The base of the logarithm that is used to calculate the Jensen-Shannon divergence.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.

data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A matrix of results with iter rows. A 0 means, that the null hypothesis that the response probability exceeds p0 was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_fujikawa(k = 3, p0 = 0.2)
get_results(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  epsilon = 2, tau = 0, iter = 100)
```

---

get_results.jsdglobal *Get Results for Simulation of a Basket Trial with the Power Prior Design Based on Global JSD Weights*

Description

Get Results for Simulation of a Basket Trial with the Power Prior Design Based on Global JSD Weights

Usage

```
## S3 method for class 'jsdglobal'
get_results(
  design,
  n,
  p1 = NULL,
  lambda,
  eps_pair,
  tau = 0,
  eps_all,
  logbase = 2,
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object of class jsdgen.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.

eps_pair	Tuning parameter that determines the amount of borrowing based on pairwise similarity.
tau	Tuning parameter that determines how similar the baskets have to be that information is shared.
eps_all	Tuning parameter that determines the amount of borrowing based on overall heterogeneity.
logbase	Tuning parameter. The base of the logarithm that is used to calculate the Jensen-Shannon divergence.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A matrix of results with iter rows. A 0 means, that the null hypothesis that the response probability exceeds p0 was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_jsdglobal(k = 3, p0 = 0.2)
get_results(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  eps_pair = 2, eps_all = 2, iter = 100)
```

---

get_results.mml *Get Results for Simulation of a Basket Trial with the MML Design*

Description

Get Results for Simulation of a Basket Trial with the MML Design

Usage

```
## S3 method for class 'mml'
get_results(design, n, p1 = NULL, lambda, iter = 1000, data = NULL, ...)
```

Arguments

design	An object of class mml.
n	The sample size per basket.
p1	Probabilities used for the simulation. If NULL then all probabilities are set to p0.
lambda	The posterior probability threshold.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A matrix of results with `iter` rows. A 0 means, that the null hypothesis that the response probability exceeds `p0` was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_mml(k = 3, p0 = 0.2)
get_results(design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  iter = 100)
```

---

get_results.mmlglobal *Get Results for Simulation of a Basket Trial with the Global MML Design*

Description

Get Results for Simulation of a Basket Trial with the Global MML Design

Usage

```
## S3 method for class 'mmlglobal'
get_results(design, n, p1 = NULL, lambda, iter = 1000, data = NULL, ...)
```

Arguments

`design`	An object of class `mmlglobal`.
`n`	The sample size per basket.
`p1`	Probabilities used for the simulation. If `NULL` then all probabilities are set to p0.
`lambda`	The posterior probability threshold.
`iter`	The number of iterations in the simulation. Is ignored if `data` is specified.
`data`	A data matrix with k column with the number of responses for each basket. Has to be generated with `get_data`. If `data` is used, then `iter` is ignored.
`...`	Further arguments.

Value

A matrix of results with `iter` rows. A 0 means, that the null hypothesis that the response probability exceeds `p0` was not rejected, a 1 means, that the null hypothesis was rejected.

Examples

```
design <- setup_mmlglobal(k = 3, p0 = 0.2)
get_results(design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  iter = 100)
```

get_scenarios *Create a Scenario Matrix*

Description

Creates a default scenario matrix.

Usage

```
get_scenarios(design, p1)
```

Arguments

design	An object created with one of the setup functions.
p1	Probability under the alternative hypothesis.

Details

get_scenarios creates a default scenario matrix that can be used for opt_design. The function creates k + 1 scenarios, from a global null to a global alternative scenario.

Value

A matrix with k rows and k + 1 columns.

Examples

```
design <- setup_fujikawa(k = 3, p0 = 0.2)
get_scenarios(design = design, p1 = 0.5)
```

opt_design *Optimize a Basket Trial Design*

Description

Optimize a Basket Trial Design

Usage

```
opt_design(
  design,
  n,
  alpha,
  design_params = list(),
  scenarios,
  prec_digits,
```

```
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object created with one of the setup functions.
n	The sample size per basket.
alpha	The one-sided significance level.
design_params	A list of params that is specific to the class of design.
scenarios	A matrix of scenarios.
prec_digits	Number of decimal places that are considered when adjusting lambda.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A list of data matrices generated with get_data. The list elements have to correspond to the columsn of scenarios.
...	Further arguments.

Value

A matrix with the expected number of correct decisions.

Examples

```
design <- setup_fujikawa(k = 3, p0 = 0.2)
scenarios <- get_scenarios(design, p1 = 0.5)

# Without simulated data
opt_design(design, n = 20, alpha = 0.05, design_params =
  list(epsilon = c(1, 2), tau = c(0, 0.5)), scenarios = scenarios,
  prec_digits = 3)

# With simulated data
scenario_list <- as.list(data.frame(scenarios))
data_list <- lapply(scenario_list,
  function(x) get_data(k = 3, n = 20, p = x, iter = 1000))
opt_design(design, n = 20, alpha = 0.05, design_params =
  list(epsilon = c(1, 2), tau = c(0, 0.5)), scenarios = scenarios,
  prec_digits = 3, data = data_list)
```

setup_bhm *Setup BHM Design Object*

Description

Setup BHM Design Object

Usage

```
setup_bhm(k, p0, p_target, mu_mean = NULL, mu_sd = 100)
```

Arguments

k	The number of baskets.
p0	A common probability under the null hypothesis.
p_target	The response rate of interest. See details.
mu_mean	Mean of the normal prior distribution for the mean of the thetas. See details.
mu_sd	Standard deviation of the normal prior distribution for the mean of the thetas.

Details

The class bhm implements the Bayesian Hierarchical Model proposed by Berry et al. (2013). Methods for this class are mostly wrappers for functions from the package bhmbasket.

In the BHM the thetas of all baskets are modeled, where theta_i = logit(p_i) - logit(p_target). These thetas are assumed to come from a normal distribution with mean mu_mean and standard deviation mu_sd. If mu_mean = NULL then mu_mean is determined as logit(p0) - logit(p_target), hence the mean of the normal distribution corresponds to the null hypothesis.

Value

An S3 object of class bhm

References

Berry, S. M., Broglio, K. R., Groshen, S., & Berry, D. A. (2013). Bayesian hierarchical modeling of patient subpopulations: efficient designs of phase II oncology clinical trials. Clinical Trials, 10(5), 720-734.

Examples

```
design_bhm <- setup_bhm(k = 3, p0 = 0.2, p_target = 0.5)
```

setup_bma *Setup bma Design Object*

Description

Creates an object of class bma.

Usage

```
setup_bma(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

k	The number of baskets.
p0	A common probability under the null hypothesis.
shape1	First common shape parameter of the beta prior.
shape2	Second common shape parameter of the beta prior.

Details

The class bma implements the Bayesian Model Averaging design by Pisoda et al. (2021). Functions for this class are mostly wrappers for functions of the bmabasket package.

Value

An S3 object of class bma

References

Psioda, M. A., Xu, J., Jiang, Q. I., Ke, C., Yang, Z., & Ibrahim, J. G. (2021). Bayesian adaptive basket trial design using model averaging. Biostatistics, 22(1), 19-34.

Examples

```
design_bma <- setup_bma(k = 3, p0 = 0.2)
```

setup_cpp *Setup Calibrated Power Prior Design Object*

Description

Setup Calibrated Power Prior Design Object

Usage

```
setup_cpp(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

k	The number of baskets.
p0	A common probability under the null hypothesis.
shape1	First common shape parameter of the beta prior.
shape2	Second common shape parameter of the beta prior.

Details

The class cpp implements a version of the power prior design, in which the amount of information that is shared between baskets is determined by the Kolmogorov-Smirnov test statistic between baskets (which is equivalent to the absolut difference in response rates).

Value

An S3 object of class cpp

References

Baumann, L., Sauer, L., & Kieser, M. (2024). A basket trial design based on power priors. arXiv:2309.06988.

Examples

```
design_cpp <- setup_cpp(k = 3, p0 = 0.2)
```

`setup_cppglobal` *Setup Global Calibrated Power Prior Design Object*

Description

Setup Global Calibrated Power Prior Design Object

Usage

```
setup_cppglobal(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

`k`	The number of baskets.
`p0`	A common probability under the null hypothesis.
`shape1`	First common shape parameter of the beta prior.
`shape2`	Second common shape parameter of the beta prior.

Details

The class `cppglobal` implements a version of the power prior design, in which the amount of information that is shared between baskets is determined by the Kolmogorov-Smirnov test statistic between basekts and a function based on response rate differences that quantifies the overall heterogeneity.

Value

An S3 object of class `cppglobal`

References

Baumann, L., Sauer, L., & Kieser, M. (2024). A basket trial design based on power priors. arXiv:2309.06988.

Examples

```
design_cppglobal <- setup_cppglobal(k = 3, p0 = 0.2)
```

setup_exnex *Setup EXNEX Design Object*

Description

Setup EXNEX Design Object

Usage

```
setup_exnex(
  k,
  p0,
  basket_mean = NULL,
  basket_sd = 100,
  mu_mean = NULL,
  mu_sd = 100
)
```

Arguments

k	The number of baskets.
p0	A common probability under the null hypothesis.
basket_mean	Mean of the normal prior distribution of the individual thetas (NEX part). See details.
basket_sd	Standard deviation of the normal prior distribution of the individual thetas (NEX part).
mu_mean	Mean of the normal prior exchangeability distribution for the mean of the thetas (EX part). See details.
mu_sd	Standard deviation of the normal prior exchangeability distribution for the mean of the thetas (EX part).

Details

The class exnex implements the EXNEX model proposed by Neuenschwander et al. (2016). Methods for this class are mostly wrappers for functions from the package bhmbasket.

In the EXNEX model the thetas of all baskets are modeled as a mixture of individual models and a Bayesian Hierarchical Model with a fixed mixture weight w. If mu_mean and basket_mean are NULL then they are set to logit(p0). Note that Neuenschwander et al. (2016) use different prior means and standard deviations. The default values here are used for better comparison with the BHM model (see setup_bhm).

Value

An S3 object of class exnex

References

Neuenschwander, B., Wandel, S., Roychoudhury, S., & Bailey, S. (2016). Robust exchangeability designs for early phase clinical trials with multiple strata. Pharmaceutical statistics, 15(2), 123-134.

Examples

```
design_exnex <- setup_exnex(k = 3, p0 = 0.2)
```

`setup_fujikawa` *Setup Fujikawa Design Object*

Description

Setup Fujikawa Design Object

Usage

```
setup_fujikawa(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

`k`	The number of baskets.
`p0`	A common probability under the null hypothesis.
`shape1`	First common shape parameter of the beta prior.
`shape2`	Second common shape parameter of the beta prior.

Details

The class `fujikawa` implements a design by Fujikawa et al. (2020) in which information is shared based on the pairwise similarity between baskets which is quantified using the Jensen-Shannon divergence between the individual posterior distributions between baskets.

Value

An S3 object of class `fujikawa`

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

```
design_fujikawa <- setup_fujikawa(k = 3, p0 = 0.2)
```

`setup_jsdglobal` *Setup Global JSD Design Object*

Description

Setup Global JSD Design Object

Usage

```
setup_jsdglobal(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

`k`	The number of baskets.
`p0`	A common probability under the null hypothesis.
`shape1`	First common shape parameter of the beta prior.
`shape2`	Second common shape parameter of the beta prior.

Details

The class `jsdglobal` implements a version of the power prior design, in which information is shared based on pairwise similarity and overall heterogeneity between baskets. Both pairwise similarity and overall heterogeneity are assessed using the Jensen-Shannon divergence.

Value

An S3 object of class `jsdglobal`

References

Baumann, L., Sauer, L., & Kieser, M. (2024). A basket trial design based on power priors. arXiv:2309.06988.

Examples

```
design_jsdglobal <- setup_jsdglobal(k = 3, p0 = 0.2)
```

`setup_mml` *Setup mml Design Object*

Description

Creates an object of class `mml`.

Usage

```
setup_mml(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

`k`	The number of baskets.
`p0`	A common probability under the null hypothesis.
`shape1`	First common shape parameter of the beta prior.
`shape2`	Second common shape parameter of the beta prior.

Details

The class `mml` implements a modified version of the empirical Bayes method by Gravestock & Held (2017) which was proposed for borrowing strength from an external study. In their approach, the sharing weight is found as the maximum of the marginal likelihood of the weight, given the external data set. This leads, however, to non-symmetric weights when applied to sharing in basket trials, i.e. Basket i would not share the information from Basket j as the other way round. Therefore, a symmetrised version is used, where the mean of the two weights resulting from sharing in both directions is used.

Value

An S3 object of class `mml`

References

Gravestock, I., & Held, L. (2017). Adaptive power priors with empirical Bayes for clinical trials. Pharmaceutical statistics, 16(5), 349-360.

Examples

```
design_mml <- setup_mml(k = 3, p0 = 0.2)
```

`setup_mmlglobal` *Setup mmlglobal Design Object*

Description

Creates an object of class `mmlglobal`.

Usage

```
setup_mmlglobal(k, p0, shape1 = 1, shape2 = 1)
```

Arguments

`k`	The number of baskets.
`p0`	A common probability under the null hypothesis.
`shape1`	First common shape parameter of the beta prior.
`shape2`	Second common shape parameter of the beta prior.

Details

The class `mmlglobal` implements an empirical Bayes method by Gravestock & Held (2019) which was proposed for borrowing strength from multiple external studies.

Value

An S3 object of class `mmlglobal`

References

Gravestock, I., & Held, L. (2019). Power priors based on multiple historical studies for binary outcomes. Biometrical Journal, 61(5), 1201-1218.

Baumann, L., Sauer, L., & Kieser, M. (2024). A basket trial design based on power priors. arXiv:2309.06988.

Examples

```
design_mmlglobal <- setup_mmlglobal(k = 3, p0 = 0.2)
```

toer *Calculate the Type 1 Error Rate for a Basket Trial Design*

Description

Calculate the Type 1 Error Rate for a Basket Trial Design

Usage

```
toer(
  design,
  n,
  p1 = NULL,
  lambda,
  design_params = list(),
  iter = 1000,
  data = NULL,
  ...
)
```

Arguments

design	An object created with one of the setup functions.
n	The sample size per basket.
p1	Probabilities under the alternative hypothesis. If NULL then the type 1 error rate under the global null hypothesis is calculated.
lambda	The posterior probability threshold.
design_params	A list of params that is specific to the class of design.
iter	The number of iterations in the simulation. Is ignored if data is specified.
data	A data matrix with k column with the number of responses for each basket. Has to be generated with get_data. If data is used, then iter is ignored.
...	Further arguments.

Value

A numeric value.

Examples

```
# Example for a basket trial with Fujikawa's Design
design <- setup_fujikawa(k = 3, p0 = 0.2)
toer(design = design, n = 20, p1 = c(0.2, 0.5, 0.5), lambda = 0.95,
  design_params = list(epsilon = 2, tau = 0), iter = 1000)
```

Index