Package: drugdevelopR (via r-universe)

August 28, 2024

Type Package

Title Utility-Based Optimal Phase II/III Drug Development Planning

Version 1.0.1

Author Stella Erdmann [aut], Johannes Cepicka [aut], Marietta Kirchner [aut], Meinhard Kieser [aut], Lukas D. Sauer [aut, cre] (<https://orcid.org/0000-0002-1340-9994>)

Maintainer Lukas D. Sauer <sauer@imbi.uni-heidelberg.de>

Description Plan optimal sample size allocation and go/no-go decision rules for phase II/III drug development programs with time-to-event, binary or normally distributed endpoints when assuming fixed treatment effects or a prior distribution for the treatment effect, using methods from Kirchner et al. (2016) <doi:10.1002/sim.6624> and Preussler (2020). Optimal is in the sense of maximal expected utility, where the utility is a function taking into account the expected cost and benefit of the program. It is possible to extend to more complex settings with bias correction (Preussler S et al. (2020) <doi:10.1186/s12874-020-01093-w>), multiple phase III trials (Preussler et al. (2019) <doi:10.1002/bimj.201700241>), multi-arm trials (Preussler et al. (2019) <doi:10.1080/19466315.2019.1702092>), and multiple endpoints (Kieser et al. (2018) <doi:10.1002/pst.1861>).

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends R (>= 3.5.0), doParallel, parallel, foreach, iterators

Imports mvtnorm, cubature, msm, MASS, stats, progressr

URL https://github.com/Sterniii3/drugdevelopR, https://sterniii3.github.io/drugdevelopR/

BugReports https://github.com/Sterniii3/drugdevelopR/issues

Suggests rmarkdown, knitr, testthat (>= 3.0.0), covr, kableExtra, magrittr, devtools

VignetteBuilder knitr

Config/testthat/parallel true

Config/testthat/edition 3

Repository https://sterniii3.r-universe.dev

RemoteUrl https://github.com/sterniii3/drugdevelopr

RemoteRef HEAD

RemoteSha 807bc3a33f854431d48d73833a7f50ed4e588197

Contents

`optimal_bias` *Optimal phase II/III drug development planning for time-to-event endpoints when discounting phase II results*

Description

The function `optimal_bias` of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules including methods for discounting of phase II results for time-to-event endpoints (Preussler et. al, 2020). The discounting may be necessary as programs that proceed to phase III can be overoptimistic about the treatment effect (i.e. they are biased). The assumed true treatment effects can be assumed fixed (planning is then also possible via user friendly R Shiny App: bias) or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_bias(
  w,
  hr1,
  hr2,
  id1,
  id2,
  d2min,
  d2max,
  stepd2,
  hrgomin,
  hrgomax,
  stephrgo,
  adj = "both",
  lambdamin = NULL,
  lambdamax = NULL,
  steplambda = NULL,
  alphaCImin = NULL,
  alphaCImax = NULL,
  stepalphaCI = NULL,
  alpha,
  beta,
  xi2,
  xi3,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  b1,
  b2,
  b3,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
hr1	first assumed true treatment effect on HR scale for prior distribution, see the vignette on priors as well as the Shiny app for more details concerning the definition of a prior distribution.
hr2	second assumed true treatment effect on HR scale for prior distribution

id1	amount of information for hr1 in terms of number of events
id2	amount of information for hr2 in terms of number of events
d2min	minimal number of events for phase II
d2max	maximal number of events for phase II
stepd2	stepsize for the optimization over d2
hrgomin	minimal threshold value for the go/no-go decision rule
hrgomax	maximal threshold value for the go/no-go decision rule
stephrgo	stepsize for the optimization over HRgo
adj	choose type of adjustment: "multiplicative", "additive", "both" or "all". When using "both", res[1,] contains the results using the multiplicative method and res[2,] contains the results using the additive method. When using "all", there are also res[3,] and res[4,], containing the results of a multiplicative and an additive method which do not only adjust the treatment effect but also the threshold value for the decision rule.
lambdamin	minimal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
lambdamax	maximal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
steplambda	stepsize for the adjustment parameter lambda
alphaCImin	minimal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
alphaCImax	maximal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
stepalphaCI	stepsize for alphaCI
alpha	one-sided significance level
beta	1-beta power for calculation of the number of events for phase III by Schoenfeld (1981) formula
xi2	event rate for phase II
xi3	event rate for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g., no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g., no constraint
steps1	lower boundary for effect size category "small" in HR scale, default: 1
stepm1	lower boundary for effect size category "medium" in HR scale = upper boundary for effect size category "small" in HR scale, default: 0.95

`stepl1`	lower boundary for effect size category "large" in HR scale = upper boundary for effect size category "medium" in HR scale, default: 0.85
`b1`	expected gain for effect size category "small" in 10^5 $
`b2`	expected gain for effect size category "medium" in 10^5 $
`b3`	expected gain for effect size category "large" in 10^5 $
`fixed`	choose if true treatment effects are fixed or random, if TRUE hr1 is used as fixed effect
`num_cl`	number of clusters used for parallel computing, default: 1

Value

The output of the function is a `data.frame` object containing the optimization results:

Method Type of adjustment: "multipl." (multiplicative adjustment of effect size), "add." (additive adjustment of effect size), "multipl2." (multiplicative adjustment of effect size and threshold), "add2." (additive adjustment of effect size and threshold)

Adj optimal adjustment parameter (lambda or alphaCI according to Method)

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

HRgo optimal threshold value for the decision rule to go to phase III

d2 optimal total number of events for phase II

d3 total expected number of events for phase III; rounded to next natural number

d total expected number of events in the program; d = d2 + d3

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Preussler, S., Kirchner, M., Goette, H., Kieser, M. (2020). Optimal designs for phase II/III drug development programs including methods for discounting of phase II results. Submitted to peer-review journal.

Schoenfeld, D. (1981). The asymptotic properties of nonparametric tests for comparing survival distributions. Biometrika, 68(1), 316-319.

Examples

```
# Activate progress bar (optional)
## Not run:
progressr::handlers(global = TRUE)

## End(Not run)
# Optimize

optimal_bias(w = 0.3,                        # define parameters for prior
 hr1 = 0.69, hr2 = 0.88, id1 = 210, id2 = 420,    # (https://web.imbi.uni-heidelberg.de/prior/)
  d2min = 20, d2max = 100, stepd2 = 5,              # define optimization set for d2
  hrgomin = 0.7, hrgomax = 0.9, stephrgo = 0.05,    # define optimization set for HRgo
  adj = "both",                                     # choose type of adjustment
  lambdamin = 0.2, lambdamax = 1, steplambda = 0.05, # define optimization set for lambda
  alphaCImin = 0.025, alphaCImax = 0.5,
  stepalphaCI = 0.025,                             # define optimization set for alphaCI
 alpha = 0.025, beta = 0.1, xi2 = 0.7, xi3 = 0.7,  # drug development planning parameters
 c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,          # fixed/variable costs for phase II/III
 K = Inf, N = Inf, S = -Inf,                       # set constraints
 steps1 = 1,                                       # define lower boundary for "small"
 stepm1 = 0.95,                                    # "medium"
 stepl1 = 0.85,                                    # and "large" effect size categories
 b1 = 1000, b2 = 2000, b3 = 3000,                  # define expected benefits
 fixed = FALSE,                                  # true treatment effects are fixed/random
 num_cl = 1)                                     # number of coresfor parallelized computing
```

`optimal_bias_binary` *Optimal phase II/III drug development planning when discounting phase II results with binary endpoint*

Description

The function `optimal_bias_binary` of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules including methods for discounting of phase II results for binary endpoints (Preussler et. al, 2020). The discounting may be necessary as programs that proceed to phase III can be overoptimistic about the treatment effect (i.e. they are biased). The assumed true treatment effects can be assumed fixed

or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_bias_binary(
  w,
  p0,
  p11,
  p12,
  in1,
  in2,
  n2min,
  n2max,
  stepn2,
  rrgomin,
  rrgomax,
  steprrgo,
  adj = "both",
  lambdamin = NULL,
  lambdamax = NULL,
  steplambda = NULL,
  alphaCImin = NULL,
  alphaCImax = NULL,
  stepalphaCI = NULL,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  b1,
  b2,
  b3,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
p0	assumed true rate of control group, see here for details

p11	assumed true rate of treatment group, see here for details
p12	assumed true rate of treatment group, see here for details
in1	amount of information for p11 in terms of sample size, see here for details
in2	amount of information for p12 in terms of sample size, see here for details
n2min	minimal total sample size for phase II; must be an even number
n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
rrgomin	minimal threshold value for the go/no-go decision rule
rrgomax	maximal threshold value for the go/no-go decision rule
steprrgo	step size for the optimization over RRgo
adj	choose type of adjustment: "multiplicative", "additive", "both" or "all". When using "both", res[1,] contains the results using the multiplicative method and res[2,] contains the results using the additive method. When using "all", there are also res[3,] and res[4,], containing the results of a multiplicative and an additive method which do not only adjust the treatment effect but also the threshold value for the decision rule.
lambdamin	minimal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
lambdamax	maximal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
steplambda	stepsize for the adjustment parameter lambda
alphaCImin	minimal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
alphaCImax	maximal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
stepalphaCI	stepsize for alphaCI
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the number of events for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small" in RR scale, default: 1
stepm1	lower boundary for effect size category "medium" in RR scale = upper boundary for effect size category "small" in RR scale, default: 0.95

`stepl1`	lower boundary for effect size category "large" in RR scale = upper boundary for effect size category "medium" in RR scale, default: 0.85
`b1`	expected gain for effect size category "small"
`b2`	expected gain for effect size category "medium"
`b3`	expected gain for effect size category "large"
`fixed`	choose if true treatment effects are fixed or random, if TRUE p11 is used as fixed effect for p1
`num_cl`	number of clusters used for parallel computing, default: 1

Value

The output of the function is a `data.frame` object containing the optimization results:

Method Type of adjustment: "multipl." (multiplicative adjustment of effect size), "add." (additive adjustment of effect size), "multipl2." (multiplicative adjustment of effect size and threshold), "add2." (additive adjustment of effect size and threshold)

Adj optimal adjustment parameter (lambda or alphaCI according to Method)

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

RRgo optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_bias_binary(w = 0.3,                 # define parameters for prior
  p0 = 0.6, p11 =  0.3, p12 = 0.5,
  in1 = 30, in2 = 60,                      # (https://web.imbi.uni-heidelberg.de/prior/)
  n2min = 20, n2max = 100, stepn2 = 10,              # define optimization set for n2
  rrgomin = 0.7, rrgomax = 0.9, steprrgo = 0.05,     # define optimization set for RRgo
  adj = "both",                                      # choose type of adjustment
  alpha = 0.025, beta = 0.1,                       # drug development planning parameters
  lambdamin = 0.2, lambdamax = 1, steplambda = 0.05,  # define optimization set for lambda
  alphaCImin = 0.025, alphaCImax = 0.5,
  stepalphaCI = 0.025,                              # define optimization set for alphaCI
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,        # fixed and variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                        # set constraints
  steps1 = 1,                                       # define lower boundary for "small"
  stepm1 = 0.95,                                     # "medium"
  stepl1 = 0.85,                                    # and "large" effect size categories
  b1 = 1000, b2 = 2000, b3 = 3000,                   # define expected benefits
  fixed = TRUE,                                    # true treatment effects are fixed/random
  num_cl = 1)                                     # number of cores for parallelized computing
```

`optimal_bias_normal` *Optimal phase II/III drug development planning when discounting phase II results with normally distributed endpoint*

Description

The function `optimal_bias_normal` of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules including methods for discounting of phase II results for normally distributed endpoints (Preussler et. al, 2020). The discounting may be necessary as programs that proceed to phase III can be overoptimistic about the treatment effect (i.e. they are biased). The assumed true treatment effects can be assumed fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_bias_normal(
  w,
  Delta1,
  Delta2,
  in1,
  in2,
```

```
  a,
  b,
  n2min,
  n2max,
  stepn2,
  kappamin,
  kappamax,
  stepkappa,
  adj = "both",
  lambdamin = NULL,
  lambdamax = NULL,
  steplambda = NULL,
  alphaCImin = NULL,
  alphaCImax = NULL,
  stepalphaCI = NULL,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 0,
  stepm1 = 0.5,
  stepl1 = 0.8,
  b1,
  b2,
  b3,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
Delta1	assumed true prior treatment effect measured as the standardized difference in means, see here for details
Delta2	assumed true prior treatment effect measured as the standardized difference in means, see here for details
in1	amount of information for Delta1 in terms of sample size, see here for details
in2	amount of information for Delta2 in terms of sample size, see here for details
a	lower boundary for the truncation of the prior distribution
b	upper boundary for the truncation of the prior distribution
n2min	minimal total sample size for phase II; must be an even number

n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
kappamin	minimal threshold value kappa for the go/no-go decision rule
kappamax	maximal threshold value kappa for the go/no-go decision rule
stepkappa	step size for the optimization over the threshold value kappa
adj	choose type of adjustment: "multiplicative", "additive", "both" or "all". When using "both", res[1,] contains the results using the multiplicative method and res[2,] contains the results using the additive method. When using "all", there are also res[3,] and res[4,], containing the results of a multiplicative and an additive method which do not only adjust the treatment effect but also the threshold value for the decision rule.
lambdamin	minimal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
lambdamax	maximal multiplicative adjustment parameter lambda (i.e. use estimate with a retention factor)
steplambda	stepsize for the adjustment parameter lambda
alphaCImin	minimal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
alphaCImax	maximal additive adjustment parameter alphaCI (i.e. adjust the lower bound of the one-sided confidence interval)
stepalphaCI	stepsize for alphaCI
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small", default: 0
stepm1	lower boundary for effect size category "medium" = upper boundary for effect size category "small" default: 0.5
stepl1	lower boundary for effect size category "large" = upper boundary for effect size category "medium", default: 0.8
b1	expected gain for effect size category "small" in 10^5 $
b2	expected gain for effect size category "medium" in 10^5 $
b3	expected gain for effect size category "large" in 10^5 $
fixed	choose if true treatment effects are fixed or following a prior distribution, if TRUE Delta1 is used as fixed effect
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a `data.frame` object containing the optimization results:

Method Type of adjustment: "multipl." (multiplicative adjustment of effect size), "add." (additive adjustment of effect size), "multipl2." (multiplicative adjustment of effect size and threshold), "add2." (additive adjustment of effect size and threshold)

Adj optimal adjustment parameter (lambda or alphaCI according to Method)

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

Kappa optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_bias_normal(w=0.3,             # define parameters for prior
 Delta1 = 0.375, Delta2 = 0.625, in1=300, in2=600,   # (https://web.imbi.uni-heidelberg.de/prior/)
  a = 0.25, b = 0.75,
  n2min = 20, n2max = 100, stepn2 = 10,                # define optimization set for n2
  kappamin = 0.02, kappamax = 0.2, stepkappa = 0.02,   # define optimization set for kappa
  adj = "both",                                          # choose type of adjustment
  lambdamin = 0.2, lambdamax = 1, steplambda = 0.05,   # define optimization set for lambda
  alphaCImin = 0.025, alphaCImax = 0.5,
  stepalphaCI = 0.025,                                 # define optimization set for alphaCI
  alpha = 0.025, beta = 0.1,                          # drug development planning parameters
```

```
c2 = 0.675, c3 = 0.72, c02 = 15, c03 = 20,      # fixed and variable costs for phase II/III
K = Inf, N = Inf, S = -Inf,                       # set constraints
steps1 = 0,                                     # define lower boundary for "small"
stepm1 = 0.5,                                     # "medium"
stepl1 = 0.8,                                   # and "large" effect size categories
b1 = 3000, b2 = 8000, b3 = 10000,                 # define expected benefits
fixed = TRUE,                                  # true treatment effects are fixed/random
num_cl = 1)                                   # number of coresfor parallelized computing
```

`optimal_binary` *Optimal phase II/III drug development planning with binary endpoint*

Description

The `optimal_binary` function of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules for binary endpoints. In this case, the treatment effect is measured by the risk ratio (RR). The assumed true treatment effects can be assumed to be fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_binary(
  w,
  p0,
  p11,
  p12,
  in1,
  in2,
  n2min,
  n2max,
  stepn2,
  rrgomin,
  rrgomax,
  steprrgo,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
```

```
  stepl1 = 0.85,
  b1,
  b2,
  b3,
  gamma = 0,
  fixed = FALSE,
  skipII = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
p0	assumed true rate of control group, see here for details
p11	assumed true rate of treatment group, see here for details
p12	assumed true rate of treatment group, see here for details
in1	amount of information for p11 in terms of sample size, see here for details
in2	amount of information for p12 in terms of sample size, see here for details
n2min	minimal total sample size for phase II; must be an even number
n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
rrgomin	minimal threshold value for the go/no-go decision rule
rrgomax	maximal threshold value for the go/no-go decision rule
steprrgo	step size for the optimization over RRgo
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the number of events for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small" in RR scale, default: 1
stepm1	lower boundary for effect size category "medium" in RR scale = upper boundary for effect size category "small" in RR scale, default: 0.95
stepl1	lower boundary for effect size category "large" in RR scale = upper boundary for effect size category "medium" in RR scale, default: 0.85

b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
gamma	to model different populations in phase II and III choose gamma != 0, default: 0, see here for details
fixed	choose if true treatment effects are fixed or random, if TRUE p11 is used as fixed effect for p1
skipII	skipII choose if skipping phase II is an option, default: FALSE; if TRUE, the program calculates the expected utility for the case when phase II is skipped and compares it to the situation when phase II is not skipped. The results are then returned as a two-row data frame, res[1,] being the results when including phase II and res[2,] when skipping phase II. res[2,] has an additional parameter, res[2,]$median_prior_RR, which is the assumed effect size used for planning the phase III study when the phase II is skipped.
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a data.frame object containing the optimization results:

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

RRgo optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking cat(comment()) of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run:
progressr::handlers(global = TRUE)

## End(Not run)
# Optimize

optimal_binary(w = 0.3,                              # define parameters for prior
  p0 = 0.6, p11 =  0.3, p12 = 0.5,
  in1 = 30, in2 = 60,                         # (https://web.imbi.uni-heidelberg.de/prior/)
  n2min = 20, n2max = 100, stepn2 = 4,               # define optimization set for n2
  rrgomin = 0.7, rrgomax = 0.9, steprrgo = 0.05,    # define optimization set for RRgo
  alpha = 0.025, beta = 0.1,                       # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,       # fixed and variable costs for phase II/III,
  K = Inf, N = Inf, S = -Inf,                        # set constraints
  steps1 = 1,                                        # define lower boundary for "small"
  stepm1 = 0.95,                                     # "medium"
  stepl1 = 0.85,                                # and "large" treatment effect size categories
  b1 = 1000, b2 = 2000, b3 = 3000,                   # define expected benefits
  gamma = 0,                                       # population structures in phase II/III
  fixed = FALSE,                                  # true treatment effects are fixed/random
  skipII = FALSE,                                 # choose if skipping phase II is an option
  num_cl = 2)                                   # number of cores for parallelized computing
```

`optimal_multiarm` *Optimal phase II/III drug development planning for multi-arm programs with time-to-event endpoint*

Description

The function `optimal_multiarm` of the drugdevelopR package enables planning of multi-arm phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules (Preussler et. al, 2019) for time-to-event endpoints. So far, only three-arm trials with two treatments and one control are supported. The assumed true treatment effects are assumed fixed (planning is also possible via user-friendly R Shiny App: multiarm). Fast computing is enabled by parallel programming.

Usage

```
optimal_multiarm(
  hr1,
  hr2,
  ec,
  n2min,
  n2max,
  stepn2,
```

```
  hrgomin,
  hrgomax,
  stephrgo,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  b1,
  b2,
  b3,
  strategy,
  num_cl = 1
)
```

Arguments

hr1	assumed true treatment effect on HR scale for treatment 1
hr2	assumed true treatment effect on HR scale for treatment 2
ec	control arm event rate for phase II and III
n2min	minimal total sample size in phase II, must be divisible by 3
n2max	maximal total sample size in phase II, must be divisible by 3
stepn2	stepsize for the optimization over n2, must be divisible by 3
hrgomin	minimal threshold value for the go/no-go decision rule
hrgomax	maximal threshold value for the go/no-go decision rule
stephrgo	step size for the optimization over HRgo
alpha	one-sided significance level/family-wise error rate
beta	type-II error rate for any pair, i.e. `1 - beta` is the (any-pair) power for calculation of the number of events for phase III
c2	variable per-patient cost for phase II
c3	variable per-patient cost for phase III
c02	fixed cost for phase II
c03	fixed cost for phase III
K	constraint on the costs of the program, default: `Inf`, e.g. no constraint
N	constraint on the total expected sample size of the program, default: `Inf`, e.g. no constraint

S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small" in HR scale, default: 1
stepm1	lower boundary for effect size category "medium" in HR scale = upper boundary for effect size category "small" in HR scale, default: 0.95
stepl1	lower boundary for effect size category "large" in HR scale = upper boundary for effect size category "medium" in HR scale, default: 0.85
b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
strategy	choose strategy: 1 (only the best promising candidate), 2 (all promising candidates) or 3 (both strategies)
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a data.frame object containing the optimization results:

Strategy Strategy, 1: "only best promising" or 2: "all promising"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

HRgo optimal threshold value for the decision rule to go to phase III

d2 optimal total number of events for phase II

d3 total expected number of events for phase III; rounded to next natural number

d total expected number of events in the program; d = d2 + d3

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg2 probability of a successful program with two arms in phase III

sProg3 probability of a successful program with three arms in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking cat(comment()) of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Preussler, S., Kirchner, M., Goette, H., Kieser, M. (2019). Optimal Designs for Multi-Arm Phase II/III Drug Development Programs. Submitted to peer-review journal.

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_multiarm(hr1 = 0.75, hr2 = 0.80,    # define assumed true HRs
  ec = 0.6,                                    # control arm event rate
  n2min = 30, n2max = 90, stepn2 = 6,          # define optimization set for n2
  hrgomin = 0.7, hrgomax = 0.9, stephrgo = 0.05,    # define optimization set for HRgo
  alpha = 0.025, beta = 0.1,                    # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,     # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                  # set constraints
  steps1 = 1,                                  # define lower boundary for "small"
  stepm1 = 0.95,                               # "medium"
  stepl1 = 0.85,                               # and "large" effect size categories
  b1 = 1000, b2 = 2000, b3 = 3000,             # define expected benefit
  strategy = 1,                                # choose strategy: 1, 2 or 3
  num_cl = 1)                                  # number of cores for parallelized computing
```

`optimal_multiarm_binary` *Optimal phase II/III drug development planning for multi-arm programs with binary endpoint*

Description

The `optimal_multiarm_binary` function enables planning of multi-arm phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules. For binary endpoints the treatment effect is measured by the risk ratio (RR). So far, only three-arm trials with two treatments and one control are supported. The assumed true treatment effects can be assumed fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_multiarm_binary(
  p0,
  p11,
  p12,
```

```
  n2min,
  n2max,
  stepn2,
  rrgomin,
  rrgomax,
  steprrgo,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  b1,
  b2,
  b3,
  strategy,
  num_cl = 1
)
```

Arguments

p0	assumed true rate of the control group
p11	assumed true rate of the treatment arm 1
p12	assumed true rate of treatment arm 2
n2min	minimal total sample size in phase II, must be divisible by 3
n2max	maximal total sample size in phase II, must be divisible by 3
stepn2	stepsize for the optimization over n2, must be divisible by 3
rrgomin	minimal threshold value for the go/no-go decision rule
rrgomax	maximal threshold value for the go/no-go decision rule
steprrgo	step size for the optimization over RRgo
alpha	one-sided significance level/family-wise error rate
beta	type-II error rate for any pair, i.e. `1 - beta` is the (any-pair) power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II
c3	variable per-patient cost for phase III
c02	fixed cost for phase II
c03	fixed cost for phase III
K	constraint on the costs of the program, default: `Inf`, e.g. no constraint

N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small" in RR scale, default: 1
stepm1	lower boundary for effect size category "medium" in RR scale = upper boundary for effect size category "small" in RR scale, default: 0.95
stepl1	lower boundary for effect size category "large" in RR scale = upper boundary for effect size category "medium" in RR scale, default: 0.85
b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
strategy	choose strategy: 1 (only the best promising candidate), 2 (all promising candidates) or 3 (both strategies)
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a data.frame object containing the optimization results:

Strategy Strategy, 1: "only best promising" or 2: "all promising"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

RRgo optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg2 probability of a successful program with two arms in phase III

sProg3 probability of a successful program with three arms in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking cat(comment()) of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_multiarm_binary( p0 = 0.6,
  p11 =  0.3, p12 = 0.5,
  n2min = 20, n2max = 100, stepn2 = 4,              # define optimization set for n2
  rrgomin = 0.7, rrgomax = 0.9, steprrgo = 0.05,    # define optimization set for RRgo
  alpha = 0.025, beta = 0.1,                       # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,        # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                       # set constraints
  steps1 = 1,                                       # define lower boundary for "small"
  stepm1 = 0.95,                                    # "medium"
  stepl1 = 0.85,                                   # and "large" effect size categories
  b1 = 1000, b2 = 2000, b3 = 3000,                  # define expected benefits
  strategy = 1, num_cl = 1)                    # number of cores for parallelized computing
```

`optimal_multiarm_normal` *Optimal phase II/III drug development planning for multi-arm programs with normally distributed endpoint*

Description

The `optimal_multiarm_normal` function enables planning of multi-arm phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules. For normally distributed endpoints, the treatment effect is measured by the standardized difference in means (Delta). So far, only three-arm trials with two treatments and one control are supported. The assumed true treatment effects can be assumed fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_multiarm_normal(
  Delta1,
  Delta2,
  n2min,
  n2max,
  stepn2,
  kappamin,
  kappamax,
  stepkappa,
  alpha,
  beta,
  c2,
```

```
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 0,
  stepm1 = 0.5,
  stepl1 = 0.8,
  b1,
  b2,
  b3,
  strategy,
  num_cl = 1
)
```

Arguments

Delta1	assumed true treatment effect as the standardized difference in means for treatment arm 1
Delta2	assumed true treatment effect as the standardized difference in means for treatment arm 2
n2min	minimal total sample size in phase II, must be divisible by 3
n2max	maximal total sample size in phase II, must be divisible by 3
stepn2	stepsize for the optimization over n2, must be divisible by 3
kappamin	minimal threshold value kappa for the go/no-go decision rule
kappamax	maximal threshold value kappa for the go/no-go decision rule
stepkappa	step size for the optimization over the threshold value kappa
alpha	one-sided significance level/family-wise error rate
beta	type-II error rate for any pair, i.e. `1 - beta` is the (any-pair) power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II
c3	variable per-patient cost for phase III
c02	fixed cost for phase II
c03	fixed cost for phase III
K	constraint on the costs of the program, default: `Inf`, e.g. no constraint
N	constraint on the total expected sample size of the program, default: `Inf`, e.g. no constraint
S	constraint on the expected probability of a successful program, default: `-Inf`, e.g. no constraint
steps1	lower boundary for effect size category "small", default: 0
stepm1	lower boundary for effect size category "medium" = upper boundary for effect size category "small" default: 0.5

stepl1	lower boundary for effect size category "large" = upper boundary for effect size category "medium", default: 0.8
b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
strategy	choose strategy: 1 (only the best promising candidate), 2 (all promising candidates) or 3 (both strategies)
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a `data.frame` object containing the optimization results:

Strategy Strategy, 1: "only best promising" or 2: "all promising"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

Kappa optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg2 probability of a successful program with two arms in phase III

sProg3 probability of a successful program with three arms in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_multiarm_normal(Delta1 = 0.375, Delta2 = 0.625,
  n2min = 20, n2max = 100, stepn2 = 4,                # define optimization set for n2
  kappamin = 0.02, kappamax = 0.2, stepkappa = 0.02,  # define optimization set for kappa
```

```
  alpha = 0.025, beta = 0.1,                          # drug development planning parameters
  c2 = 0.675, c3 = 0.72, c02 = 15, c03 = 20,          # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                             # set constraints
  steps1 = 0,                                          # define lower boundary for "small"
  stepm1 = 0.5,                                          # "medium"
  stepl1 = 0.8,                                       # and "large" effect size categories
  b1 = 3000, b2 = 8000, b3 = 10000,                       # define expected benefits
  strategy = 1,
  num_cl = 1)                                    # number of cores for parallelized computing
```

---

optimal_multiple_normal

Optimal phase II/III drug development planning for programs with multiple normally distributed endpoints

---

Description

The function optimal_multiple_normal of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules for two-arm trials with two normally distributed endpoints and one control group (Preussler et. al, 2019).

Usage

```
optimal_multiple_normal(
  Delta1,
  Delta2,
  in1,
  in2,
  sigma1,
  sigma2,
  n2min,
  n2max,
  stepn2,
  kappamin,
  kappamax,
  stepkappa,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 0,
```

```
  stepm1 = 0.5,
  stepl1 = 0.8,
  b1,
  b2,
  b3,
  rho,
  fixed,
  relaxed = FALSE,
  num_cl = 1
)
```

Arguments

Delta1	assumed true treatment effect for endpoint 1 measured as the difference in means
Delta2	assumed true treatment effect for endpoint 2 measured as the difference in means
in1	amount of information for Delta1 in terms of number of events
in2	amount of information for Delta2 in terms of number of events
sigma1	variance of endpoint 1
sigma2	variance of endpoint 2
n2min	minimal total sample size in phase II, must be divisible by 3
n2max	maximal total sample size in phase II, must be divisible by 3
stepn2	stepsize for the optimization over n2, must be divisible by 3
kappamin	minimal threshold value kappa for the go/no-go decision rule
kappamax	maximal threshold value kappa for the go/no-go decision rule
stepkappa	step size for the optimization over the threshold value kappa
alpha	one-sided significance level/family-wise error rate
beta	type-II error rate for any pair, i.e. `1 - beta` is the (any-pair) power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small", default: 0
stepm1	lower boundary for effect size category "medium" = upper boundary for effect size category "small" default: 0.5
stepl1	lower boundary for effect size category "large" = upper boundary for effect size category "medium", default: 0.8

b1	expected gain for effect size category "small" in 10^5 $
b2	expected gain for effect size category "medium" in 10^5 $
b3	expected gain for effect size category "large" in 10^5 $
rho	correlation between the two endpoints
fixed	assumed fixed treatment effect
relaxed	relaxed or strict decision rule
num_cl	number of clusters used for parallel computing, default: 1

Details

For this setting, the drug development program is defined to be successful if it proceeds from phase II to phase III and all endpoints show a statistically significant treatment effect in phase III. For example, this situation is found in Alzheimer's disease trials, where a drug should show significant results in improving cognition (cognitive endpoint) as well as in improving activities of daily living (functional endpoint).

The effect size categories small, medium and large are applied to both endpoints. In order to define an overall effect size from the two individual effect sizes, the function implements two different combination rules:

- A strict rule (`relaxed = FALSE`) assigning a large overall effect in case both endpoints show an effect of large size, a small overall effect in case that at least one of the endpoints shows a small effect, and a medium overall effect otherwise, and
- A relaxed rule (`relaxed = TRUE`) assigning a large overall effect if at least one of the endpoints shows a large effect, a small effect if both endpoints show a small effect, and a medium overall effect otherwise.

Fast computing is enabled by parallel programming.

Monte Carlo simulations are applied for calculating utility, event count and other operating characteristics in this setting. Hence, the results are affected by random uncertainty.

Value

The output of the function is a `data.frame` object containing the optimization results:

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

Kappa optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Meinhard Kieser, Marietta Kirchner, Eva Dölger, Heiko Götte (2018). Optimal planning of phase II/III programs for clinical trials with multiple endpoints

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

set.seed(123) # This function relies on Monte Carlo integration
optimal_multiple_normal(Delta1 = 0.75,
  Delta2 = 0.80, in1=300, in2=600,                  # define assumed true HRs
  sigma1 = 8, sigma2= 12,                           # variances for both endpoints
  n2min = 30, n2max = 90, stepn2 = 10,              # define optimization set for n2
  kappamin = 0.05, kappamax = 0.2, stepkappa = 0.05, # define optimization set for HRgo
  alpha = 0.025, beta = 0.1,                        # planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,          # fixed/variable costs: phase II/III
  K = Inf, N = Inf, S = -Inf,                       # set constraints
  steps1 = 0,                                       # define lower boundary for "small"
  stepm1 = 0.5,                                     # "medium"
  stepl1 = 0.8,                                     # and "large" effect size categories
  b1 = 1000, b2 = 2000, b3 = 3000,                  # define expected benefit
  rho = 0.5, relaxed = TRUE,                        # strict or relaxed rule
  fixed = TRUE,                                     # treatment effect
  num_cl = 1)                                       # parallelized computing
```

`optimal_multiple_tte` *Optimal phase II/III drug development planning for programs with multiple time-to-event endpoints*

Description

The function `optimal_multiple_tte` of the drugdevelopR package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules (Preussler et. al, 2019) in a two-arm trial with two time-to-event endpoints.

Usage

```
optimal_multiple_tte(
  hr1,
  hr2,
  id1,
  id2,
  n2min,
  n2max,
  stepn2,
  hrgomin,
  hrgomax,
  stephrgo,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  b11,
  b21,
  b31,
  b12,
  b22,
  b32,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  rho,
  fixed = TRUE,
  num_cl = 1
)
```

Arguments

hr1	assumed true treatment effect on HR scale for endpoint 1 (e.g. OS)
hr2	assumed true treatment effect on HR scale for endpoint 2 (e.g. PFS)
id1	amount of information for hr1 in terms of number of events
id2	amount of information for hr2 in terms of number of events
n2min	minimal total sample size in phase II, must be divisible by 3
n2max	maximal total sample size in phase II, must be divisible by 3
stepn2	stepsize for the optimization over n2, must be divisible by 3
hrgomin	minimal threshold value for the go/no-go decision rule
hrgomax	maximal threshold value for the go/no-go decision rule

stephrgo	step size for the optimization over HRgo
alpha	one-sided significance level/family-wise error rate
beta	type-II error rate for any pair, i.e. 1 - beta is the (any-pair) power for calculation of the number of events for phase III
c2	variable per-patient cost for phase II in 10^5 $.
c3	variable per-patient cost for phase III in 10^5 $.
c02	fixed cost for phase II in 10^5 $.
c03	fixed cost for phase III in 10^5 $.
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
b11	expected gain for effect size category "small" if endpoint 1 is significant (and endpoint 2 may or may not be significant)
b21	expected gain for effect size category "medium" if endpoint 1 is significant (and endpoint 2 may or may not be significant)
b31	expected gain for effect size category "large" if endpoint 1 is significant (and endpoint 2 may or may not be significant)
b12	expected gain for effect size category "small" if endpoint 1 is not significant, but endpoint 2 is
b22	expected gain for effect size category "medium"if endpoint 1 is not significant, but endpoint 2 is
b32	expected gain for effect size category "large" if endpoint 1 is not significant, but endpoint 2 is
steps1	lower boundary for effect size category "small" in HR scale, default: 1
stepm1	lower boundary for effect size category "medium" in HR scale = upper boundary for effect size category "small" in HR scale, default: 0.95
stepl1	lower boundary for effect size category "large" in HR scale = upper boundary for effect size category "medium" in HR scale, default: 0.85
rho	correlation between the two endpoints
fixed	assumed fixed treatment effect
num_cl	number of clusters used for parallel computing, default: 1

Details

In this setting, the drug development program is defined to be successful if it proceeds from phase II to phase III and at least one endpoint shows a statistically significant treatment effect in phase III. For example, this situation is found in oncology trials, where overall survival (OS) and progression-free survival (PFS) are the two endpoints of interest.

The gain of a successful program may differ according to the importance of the endpoint that is significant. If endpoint 1 is significant (no matter whether endpoint 2 is significant or not), then the

gains `b11`, `b21` and `b31` will be used for calculation of the utility. If only endpoint 2 is significant, then `b12`, `b22` and `b32` will be used. This also matches the oncology example, where OS (i.e. endpoint 1) implicates larger expected gains than PFS alone (i.e. endpoint 2).

Fast computing is enabled by parallel programming.

Monte Carlo simulations are applied for calculating utility, event count and other operating characteristics in this setting. Hence, the results are affected by random uncertainty. The extent of uncertainty is discussed in (Kieser et al. 2018).

Value

The output of the function is a `data.frame` object containing the optimization results:

OP probability that one endpoint is significant

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

HRgo optimal threshold value for the decision rule to go to phase III

d2 optimal total number of events for phase II

d3 total expected number of events for phase III; rounded to next natural number

d total expected number of events in the program; d = d2 + d3

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Kieser, M., Kirchner, M. Dölger, E., Götte, H. (2018).Optimal planning of phase II/III programs for clinical trials with multiple endpoints, Pharm Stat. 2018 Sep; 17(5):437-457.

Preussler, S., Kirchner, M., Goette, H., Kieser, M. (2019). Optimal Designs for Multi-Arm Phase II/III Drug Development Programs. Submitted to peer-review journal.

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

set.seed(123) # This function relies on Monte Carlo integration
optimal_multiple_tte(hr1 = 0.75,
  hr2 = 0.80, id1 = 210, id2 = 420,          # define assumed true HRs
  n2min = 30, n2max = 90, stepn2 = 6,        # define optimization set for n2
  hrgomin = 0.7, hrgomax = 0.9, stephrgo = 0.05, # define optimization set for HRgo
  alpha = 0.025, beta = 0.1,                 # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,   # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                # set constraints
  steps1 = 1,                                # define lower boundary for "small"
  stepm1 = 0.95,                             # "medium"
  stepl1 = 0.85,                             # and "large" effect size categories
  b11 = 1000, b21 = 2000, b31 = 3000,
  b12 = 1000, b22 = 1500, b32 = 2000,        # define expected benefits (both scenarios)
  rho = 0.6, fixed = TRUE,                   # correlation and treatment effect
  num_cl = 1)                               # number of cores for parallelized computing
```

`optimal_multitrial` *Optimal phase II/III drug development planning where several phase III trials are performed for time-to-event endpoints*

Description

The function `optimal_multitrial` of the drugdevelopR package enables planning of phase II/III drug development programs with time-to-event endpoints for programs with several phase III trials (Preussler et. al, 2019). Its main output values are the optimal sample size allocation and optimal go/no-go decision rules. The assumed true treatment effects can be assumed to be fixed (planning is then also possible via user friendly R Shiny App: multitrial) or can be modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_multitrial(
  w,
  hr1,
  hr2,
  id1,
  id2,
  d2min,
  d2max,
  stepd2,
```

```
  hrgomin,
  hrgomax,
  stephrgo,
  alpha,
  beta,
  xi2,
  xi3,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  b1,
  b2,
  b3,
  case,
  strategy = TRUE,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution, see this Shiny application for the choice of weights
hr1	first assumed true treatment effect on HR scale for prior distribution
hr2	second assumed true treatment effect on HR scale for prior distribution
id1	amount of information for hr1 in terms of number of events
id2	amount of information for hr2 in terms of number of events
d2min	minimal number of events for phase II
d2max	maximal number of events for phase II
stepd2	step size for the optimization over d2
hrgomin	minimal threshold value for the go/no-go decision rule
hrgomax	maximal threshold value for the go/no-go decision rule
stephrgo	step size for the optimization over HRgo
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the number of events for phase III by Schoenfeld's formula (Schoenfeld 1981)
xi2	assumed event rate for phase II, used for calculating the sample size of phase II via n2 = d2/xi2
xi3	event rate for phase III, used for calculating the sample size of phase III in analogy to xi2

c2	variable per-patient cost for phase II in 10^5 $.
c3	variable per-patient cost for phase III in 10^5 $.
c02	fixed cost for phase II in 10^5 $.
c03	fixed cost for phase III in 10^5 $.
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
case	choose case: "at least 1, 2 or 3 significant trials needed for approval"
strategy	choose strategy: "conduct 1, 2, 3 or 4 trials in order to achieve the case's goal"; TRUE calculates all strategies of the selected case
fixed	choose if true treatment effects are fixed or random, if TRUE hr1 is used as a fixed effect and hr2 is ignored
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function is a data.frame object containing the optimization results:

Case Case: "number of significant trials needed"

Strategy Strategy: "number of trials to be conducted in order to achieve the goal of the case"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

HRgo optimal threshold value for the decision rule to go to phase III

d2 optimal total number of events for phase II

d3 total expected number of events for phase III; rounded to next natural number

d total expected number of events in the program; d = d2 + d3

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III (lower boundary in HR scale is set to 1, as proposed by IQWiG (2016))

sProg2 probability of a successful program with "medium" treatment effect in phase III (lower boundary in HR scale is set to 0.95, as proposed by IQWiG (2016))

sProg3 probability of a successful program with "large" treatment effect in phase III (lower boundary in HR scale is set to 0.85, as proposed by IQWiG (2016))

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

Effect sizes

In other settings, the definition of "small", "medium" and "large" effect sizes can be user-specified using the input parameters `steps1`, `stepm1` and `stepl1`. Due to the complexity of the multitrial setting, this feature is not included for this setting. Instead, the effect sizes were set to to predefined values as explained under sProg1, sProg2 and sProg3 in the *Value* section.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Preussler, S., Kieser, M., and Kirchner, M. (2019). Optimal sample size allocation and go/no-go decision rules for phase II/III programs where several phase III trials are performed. Biometrical Journal, 61(2), 357-378.

Schoenfeld, D. (1981). The asymptotic properties of nonparametric tests for comparing survival distributions. Biometrika, 68(1), 316-319.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_multitrial(w = 0.3,                # define parameters for prior
 hr1 = 0.69, hr2 = 0.88, id1 = 210, id2 = 420,    # (https://web.imbi.uni-heidelberg.de/prior/)
  d2min = 20, d2max = 100, stepd2 = 5,              # define optimization set for d2
  hrgomin = 0.7, hrgomax = 0.9, stephrgo = 0.05,    # define optimization set for HRgo
  alpha = 0.025, beta = 0.1, xi2 = 0.7, xi3 = 0.7,  # drug development planning parameters
 c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,        # fixed and variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                       # set constraints
  b1 = 1000, b2 = 2000, b3 = 3000,                 # expected benefit for each effect size
  case = 1, strategy = TRUE,                        # chose Case and Strategy
  fixed = TRUE,                                   # true treatment effects are fixed/random
  num_cl = 1)                                    # number of cores for parallelized computing
```

optimal_multitrial_binary *Optimal phase II/III drug development planning where several phase III trials are performed*

Description

The optimal_multitrial_binary function enables planning of phase II/III drug development programs with several phase III trials for the same binary endpoint. The main output values are optimal sample size allocation and go/no-go decision rules. For binary endpoints, the treatment effect is measured by the risk ratio (RR).

Usage

```
optimal_multitrial_binary(
  w,
  p0,
  p11,
  p12,
  in1,
  in2,
  n2min,
  n2max,
  stepn2,
  rrgomin,
  rrgomax,
  steprrgo,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  b1,
  b2,
  b3,
  case,
  strategy = TRUE,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
p0	assumed true rate of control group, see here for details
p11	assumed true rate of treatment group, see here for details
p12	assumed true rate of treatment group, see here for details
in1	amount of information for p11 in terms of sample size, see here for details
in2	amount of information for p12 in terms of sample size, see here for details
n2min	minimal total sample size for phase II; must be an even number
n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
rrgomin	minimal threshold value for the go/no-go decision rule
rrgomax	maximal threshold value for the go/no-go decision rule
steprrgo	step size for the optimization over RRgo
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the number of events for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
case	choose case: "at least 1, 2 or 3 significant trials needed for approval"
strategy	choose strategy: "conduct 1, 2, 3 or 4 trials in order to achieve the case's goal"; TRUE calculates all strategies of the selected case
fixed	choose if true treatment effects are fixed or random, if TRUE p11 is used as fixed effect for p1
num_cl	number of clusters used for parallel computing, default: 1

Details

The assumed true treatment effects can be assumed fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package.

Fast computing is enabled by parallel programming.

Value

The output of the function is a `data.frame` object containing the optimization results:

Case Case: "number of significant trials needed"

Strategy Strategy: "number of trials to be conducted in order to achieve the goal of the case"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

RRgo optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III (lower boundary in HR scale is set to 1, as proposed by IQWiG (2016))

sProg2 probability of a successful program with "medium" treatment effect in phase III (lower boundary in HR scale is set to 0.95, as proposed by IQWiG (2016))

sProg3 probability of a successful program with "large" treatment effect in phase III (lower boundary in HR scale is set to 0.85, as proposed by IQWiG (2016))

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking `cat(comment())` of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

Effect sizes

In other settings, the definition of "small", "medium" and "large" effect sizes can be user-specified using the input parameters `steps1`, `stepm1` and `stepl1`. Due to the complexity of the multitrial setting, this feature is not included for this setting. Instead, the effect sizes were set to to predefined values as explained under sProg1, sProg2 and sProg3 in the *Value* section.

References

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize
```

```
optimal_multitrial_binary(w = 0.3,         # define parameters for prior
  p0 = 0.6, p11 =  0.3, p12 = 0.5,
  in1 = 30, in2 = 60,                      # (https://web.imbi.uni-heidelberg.de/prior/)
  n2min = 20, n2max = 100, stepn2 = 4,           # define optimization set for n2
  rrgomin = 0.7, rrgomax = 0.9, steprrgo = 0.05,  # define optimization set for RRgo
  alpha = 0.025, beta = 0.1,                     # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,      # fixed and variable costs for phase II/III,
  K = Inf, N = Inf, S = -Inf,                    # set constraints
  b1 = 1000, b2 = 2000, b3 = 3000,              # expected benefit for a each effect size
  case = 1, strategy = TRUE,                     # chose Case and Strategy
  fixed = TRUE,                                 # true treatment effects are fixed/random
  num_cl = 1)                                  # number of cores for parallelized computing
```

`optimal_multitrial_normal` *Optimal phase II/III drug development planning where several phase III trials are performed*

Description

The `optimal_multitrial_normal` function enables planning of phase II/III drug development programs with several phase III trials for the same normally distributed endpoint. Its main output values are optimal sample size allocation and go/no-go decision rules. For normally distributed endpoints, the treatment effect is measured by the standardized difference in means (Delta). The assumed true treatment effects can be assumed fixed or modelled by a prior distribution.

Usage

```
optimal_multitrial_normal(
  w,
  Delta1,
  Delta2,
  in1,
  in2,
  a,
  b,
  n2min,
  n2max,
  stepn2,
  kappamin,
  kappamax,
  stepkappa,
  alpha,
  beta,
  c2,
  c3,
  c02,
```

```
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  b1,
  b2,
  b3,
  case,
  strategy = TRUE,
  fixed = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
Delta1	assumed true prior treatment effect measured as the standardized difference in means, see here for details
Delta2	assumed true prior treatment effect measured as the standardized difference in means, see here for details
in1	amount of information for Delta1 in terms of sample size, see here for details
in2	amount of information for Delta2 in terms of sample size, see here for details
a	lower boundary for the truncation of the prior distribution
b	upper boundary for the truncation of the prior distribution
n2min	minimal total sample size for phase II; must be an even number
n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
kappamin	minimal threshold value kappa for the go/no-go decision rule
kappamax	maximal threshold value kappa for the go/no-go decision rule
stepkappa	step size for the optimization over the threshold value kappa
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint

b1	expected gain for effect size category "small" in 10^5 $
b2	expected gain for effect size category "medium" in 10^5 $
b3	expected gain for effect size category "large" in 10^5 $
case	choose case: "at least 1, 2 or 3 significant trials needed for approval"
strategy	choose strategy: "conduct 1, 2, 3 or 4 trials in order to achieve the case's goal"; TRUE calculates all strategies of the selected case
fixed	choose if true treatment effects are fixed or following a prior distribution, if TRUE Delta1 is used as fixed effect
num_cl	number of clusters used for parallel computing, default: 1

Details

The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Value

The output of the function is a data.frame object containing the optimization results:

Case Case: "number of significant trials needed"

Strategy Strategy: "number of trials to be conducted in order to achieve the goal of the case"

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

Kappa optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III (lower boundary in HR scale is set to 0, as proposed by Cohen (1988))

sProg2 probability of a successful program with "medium" treatment effect in phase III (lower boundary in HR scale is set to 0.5, as proposed Cohen (1988))

sProg3 probability of a successful program with "large" treatment effect in phase III (lower boundary in HR scale is set to 0.8, as proposed Cohen (1988))

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking cat(comment()) of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

Effect sizes

In other settings, the definition of "small", "medium" and "large" effect sizes can be user-specified using the input parameters `steps1`, `stepm1` and `stepl1`. Due to the complexity of the multitrial setting, this feature is not included for this setting. Instead, the effect sizes were set to to predefined values as explained under sProg1, sProg2 and sProg3 in the *Value* section.

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_multitrial_normal(w = 0.3,           # define parameters for prior
  Delta1 = 0.375, Delta2 = 0.625,
  in1 = 300, in2 = 600,                    # (https://web.imbi.uni-heidelberg.de/prior/)
  a = 0.25, b = 0.75,
  n2min = 20, n2max = 100, stepn2 = 4,              # define optimization set for n2
  kappamin = 0.02, kappamax = 0.2, stepkappa = 0.02,  # define optimization set for kappa
  alpha = 0.025, beta = 0.1,                     # drug development planning parameters
  c2 = 0.675, c3 = 0.72, c02 = 15, c03 = 20,       # fixed and variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                       # set constraints
  b1 = 3000, b2 = 8000, b3 = 10000,                # expected benefit for each effect size
  case = 1, strategy = TRUE,                        # chose Case and Strategy
  fixed = TRUE,                                  # true treatment effects are fixed/random
  num_cl = 1)                                   # number of cores for parallelized computing
```

`optimal_normal` *Optimal phase II/III drug development planning with normally distributed endpoint*

Description

The function `optimal_normal` of the `drugdevelopR` package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules for normally distributed endpoints. The treatment effect is measured by the standardized difference in means. The assumed true treatment effects can be assumed to be fixed or modelled by a prior distribution. The R Shiny application prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_normal(
  w,
  Delta1,
  Delta2,
  in1,
  in2,
  a,
  b,
  n2min,
  n2max,
  stepn2,
  kappamin,
  kappamax,
  stepkappa,
  alpha,
  beta,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 0,
  stepm1 = 0.5,
  stepl1 = 0.8,
  b1,
  b2,
  b3,
  gamma = 0,
  fixed = FALSE,
  skipII = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution
Delta1	assumed true prior treatment effect measured as the standardized difference in means, see here for details
Delta2	assumed true prior treatment effect measured as the standardized difference in means, see here for details
in1	amount of information for Delta1 in terms of sample size, see here for details
in2	amount of information for Delta2 in terms of sample size, see here for details
a	lower boundary for the truncation of the prior distribution
b	upper boundary for the truncation of the prior distribution

n2min	minimal total sample size for phase II; must be an even number
n2max	maximal total sample size for phase II, must be an even number
stepn2	step size for the optimization over n2; must be an even number
kappamin	minimal threshold value kappa for the go/no-go decision rule
kappamax	maximal threshold value kappa for the go/no-go decision rule
stepkappa	step size for the optimization over the threshold value kappa
alpha	one-sided significance level
beta	type II error rate; i.e. `1 - beta` is the power for calculation of the sample size for phase III
c2	variable per-patient cost for phase II in 10^5 $
c3	variable per-patient cost for phase III in 10^5 $
c02	fixed cost for phase II in 10^5 $
c03	fixed cost for phase III in 10^5 $
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small", default: 0
stepm1	lower boundary for effect size category "medium" = upper boundary for effect size category "small" default: 0.5
stepl1	lower boundary for effect size category "large" = upper boundary for effect size category "medium", default: 0.8
b1	expected gain for effect size category "small" in 10^5 $
b2	expected gain for effect size category "medium" in 10^5 $
b3	expected gain for effect size category "large" in 10^5 $
gamma	to model different populations in phase II and III choose `gamma != 0`, default: 0, see here for details
fixed	choose if true treatment effects are fixed or following a prior distribution, if TRUE `Delta1` is used as fixed effect
skipII	choose if skipping phase II is an option, default: FALSE; if TRUE, the program calculates the expected utility for the case when phase II is skipped and compares it to the situation when phase II is not skipped. The results are then returned as a two-row data frame, `res[1, ]` being the results when including phase II and `res[2, ]` when skipping phase II. `res[2, ]` has an additional parameter, `res[2, ]$median_prior_Delta`, which is the assumed effect size used for planning the phase III study when the phase II is skipped.
num_cl	number of clusters used for parallel computing, default: 1

Value

The output of the function optimal_normal is a data.frame containing the optimization results:

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

Kappa optimal threshold value for the decision rule to go to phase III

n2 total sample size for phase II

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters.

Taking `cat(comment())` of the data.frame object lists the used optimization sequences, start and finish date of the optimization procedure.

References

Cohen, J. (1988). Statistical power analysis for the behavioral sciences.

Examples

```
# Activate progress bar (optional)
## Not run: progressr::handlers(global = TRUE)
# Optimize

optimal_normal(w=0.3,                        # define parameters for prior
 Delta1 = 0.375, Delta2 = 0.625, in1=300, in2=600,  # (https://web.imbi.uni-heidelberg.de/prior/)
  a = 0.25, b = 0.75,
  n2min = 20, n2max = 100, stepn2 = 4,               # define optimization set for n2
  kappamin = 0.02, kappamax = 0.2, stepkappa = 0.02, # define optimization set for kappa
  alpha = 0.025, beta = 0.1,                          # drug development planning parameters
  c2 = 0.675, c3 = 0.72, c02 = 15, c03 = 20,        # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                         # set constraints
  steps1 = 0,                                         # define lower boundary for "small"
  stepm1 = 0.5,                                       # "medium"
  stepl1 = 0.8,                                      # and "large" effect size categories
  b1 = 3000, b2 = 8000, b3 = 10000,                 # benefit for each effect size category
  gamma = 0,                                        # population structures in phase II/III
  fixed = FALSE,                                   # true treatment effects are fixed/random
  skipII = FALSE,                                     # skipping phase II
```

```
  num_cl = 1)                          # number of cores for parallelized computing
```

`optimal_tte` *Optimal phase II/III drug development planning with time-to-event endpoint*

Description

The function `optimal_tte` of the `drugdevelopR` package enables planning of phase II/III drug development programs with optimal sample size allocation and go/no-go decision rules for time-to-event endpoints (Kirchner et al., 2016). The assumed true treatment effects can be assumed to be fixed or modelled by a prior distribution. When assuming fixed true treatment effects, planning can also be done with the user-friendly R Shiny app basic. The app prior visualizes the prior distributions used in this package. Fast computing is enabled by parallel programming.

Usage

```
optimal_tte(
  w,
  hr1,
  hr2,
  id1,
  id2,
  d2min,
  d2max,
  stepd2,
  hrgomin,
  hrgomax,
  stephrgo,
  alpha,
  beta,
  xi2,
  xi3,
  c2,
  c3,
  c02,
  c03,
  K = Inf,
  N = Inf,
  S = -Inf,
  steps1 = 1,
  stepm1 = 0.95,
  stepl1 = 0.85,
  b1,
  b2,
  b3,
```

```
  gamma = 0,
  fixed = FALSE,
  skipII = FALSE,
  num_cl = 1
)
```

Arguments

w	weight for mixture prior distribution, see this Shiny application for the choice of weights
hr1	first assumed true treatment effect on HR scale for prior distribution
hr2	second assumed true treatment effect on HR scale for prior distribution
id1	amount of information for hr1 in terms of number of events
id2	amount of information for hr2 in terms of number of events
d2min	minimal number of events for phase II
d2max	maximal number of events for phase II
stepd2	step size for the optimization over d2
hrgomin	minimal threshold value for the go/no-go decision rule
hrgomax	maximal threshold value for the go/no-go decision rule
stephrgo	step size for the optimization over HRgo
alpha	one-sided significance level
beta	type II error rate; i.e. 1 - beta is the power for calculation of the number of events for phase III by Schoenfeld's formula (Schoenfeld 1981)
xi2	assumed event rate for phase II, used for calculating the sample size of phase II via n2 = d2/xi2
xi3	event rate for phase III, used for calculating the sample size of phase III in analogy to xi2
c2	variable per-patient cost for phase II in 10^5 $.
c3	variable per-patient cost for phase III in 10^5 $.
c02	fixed cost for phase II in 10^5 $.
c03	fixed cost for phase III in 10^5 $.
K	constraint on the costs of the program, default: Inf, e.g. no constraint
N	constraint on the total expected sample size of the program, default: Inf, e.g. no constraint
S	constraint on the expected probability of a successful program, default: -Inf, e.g. no constraint
steps1	lower boundary for effect size category "small" in HR scale, default: 1
stepm1	lower boundary for effect size category "medium" in HR scale = upper boundary for effect size category "small" in HR scale, default: 0.95
stepl1	lower boundary for effect size category "large" in HR scale = upper boundary for effect size category "medium" in HR scale, default: 0.85

b1	expected gain for effect size category "small"
b2	expected gain for effect size category "medium"
b3	expected gain for effect size category "large"
gamma	to model different populations in phase II and III choose gamma != 0, default: 0
fixed	choose if true treatment effects are fixed or random, if TRUE hr1 is used as a fixed effect and hr2 is ignored
skipII	choose if skipping phase II is an option, default: FALSE; if TRUE, the program calculates the expected utility for the case when phase II is skipped and compares it to the situation when phase II is not skipped. The results are then returned as a two-row data frame, res[1,] being the results when including phase II and res[2,] when skipping phase II. res[2,] has an additional parameter, res[2,]$median_prior_HR, which is the assumed hazards ratio used for planning the phase III study when the phase II is skipped. It is calculated as the exponential function of the median of the prior function.
num_cl	number of clusters used for parallel computing, default: 1

Format

data.frame containing the optimization results (see Value)

Value

The output of the function is a data.frame object containing the optimization results:

u maximal expected utility under the optimization constraints, i.e. the expected utility of the optimal sample size and threshold value

HRgo optimal threshold value for the decision rule to go to phase III

d2 optimal total number of events for phase II

d3 total expected number of events for phase III; rounded to next natural number

d total expected number of events in the program; d = d2 + d3

n2 total sample size for phase II; rounded to the next even natural number

n3 total sample size for phase III; rounded to the next even natural number

n total sample size in the program; n = n2 + n3

K maximal costs of the program (i.e. the cost constraint, if it is set or the sum K2+K3 if no cost constraint is set)

pgo probability to go to phase III

sProg probability of a successful program

sProg1 probability of a successful program with "small" treatment effect in phase III

sProg2 probability of a successful program with "medium" treatment effect in phase III

sProg3 probability of a successful program with "large" treatment effect in phase III

K2 expected costs for phase II

K3 expected costs for phase III

and further input parameters. Taking cat(comment()) of the data frame lists the used optimization sequences, start and finish date of the optimization procedure.

References

Kirchner, M., Kieser, M., Goette, H., & Schueler, A. (2016). Utility-based optimization of phase II/III programs. Statistics in Medicine, 35(2), 305-316.

IQWiG (2016). Allgemeine Methoden. Version 5.0, 10.07.2016, Technical Report. Available at https://www.iqwig.de/ueber-uns/methoden/methodenpapier/, assessed last 15.05.19.

Schoenfeld, D. (1981). The asymptotic properties of nonparametric tests for comparing survival distributions. Biometrika, 68(1), 316-319.

See Also

optimal_binary, optimal_normal, optimal_bias, optimal_multitrial and optimal_multiarm

Examples

```
# Activate progress bar (optional)
## Not run:
progressr::handlers(global = TRUE)

## End(Not run)
# Optimize

optimal_tte(w = 0.3,                    # define parameters for prior
 hr1 = 0.69, hr2 = 0.88, id1 = 210, id2 = 420,  # (https://web.imbi.uni-heidelberg.de/prior/)
  d2min = 20, d2max = 100, stepd2 = 5,            # define optimization set for d2
  hrgomin = 0.7, hrgomax = 0.9, stephrgo = 0.05,  # define optimization set for HRgo
  alpha = 0.025, beta = 0.1, xi2 = 0.7, xi3 = 0.7, # drug development planning parameters
  c2 = 0.75, c3 = 1, c02 = 100, c03 = 150,        # fixed/variable costs for phase II/III
  K = Inf, N = Inf, S = -Inf,                      # set constraints
  steps1 = 1,                                      # define lower boundary for "small"
  stepm1 = 0.95,                                   # "medium"
 stepl1 = 0.85,                                 # and "large" treatment effect size categories
 b1 = 1000, b2 = 2000, b3 = 3000,              # expected benefit for each effect size category
  gamma = 0,                                      # population structures in phase II/III
  fixed = FALSE,                                 # true treatment effects are fixed/random
  skipII = FALSE,                                  # skipping phase II
  num_cl = 1)                                    # number of cores for parallelized computing
```

Index