Package 'BOIN'

October 12, 2022

Type Package

Title Bayesian Optimal INterval (BOIN) Design for Single-Agent and Drug- Combination Phase I Clinical Trials

Version 2.7.2

Date 2021-01-19

Author Ying Yuan and Suyu Liu

Maintainer Ying Yuan <yyuan@mdanderson.org>

Imports Iso

Description The Bayesian optimal interval (BOIN) design is a novel phase I clinical trial design for finding the maximum tolerated dose (MTD). It can be used to design both single-agent and drug-combination trials. The BOIN design is motivated by the top priority and concern of clinicians when testing a new drug, which is to effectively treat patients and minimize the chance of exposing them to subtherapeutic or overly toxic doses. The prominent advantage of the BOIN design is that it achieves simplicity and superior performance at the same time. The BOIN design is algorithm-based and can be implemented in a simple way similar to the traditional 3+3 design. The BOIN design yields an average performance that is comparable to that of the continual reassessment method (CRM, one of the best model-based designs) in terms of selecting the MTD, but has a substantially lower risk of assigning patients to subtherapeutic or overly toxic doses. For tutorial, please check Yan et al. (2020) <doi:10.18637/jss.v094.i13>.

License GPL-2

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-19 17:40:06 UTC

R topics documented:

get.boundary   *Generate the optimal dose escalation and deescalation boundaries for*
conducting the trial.

Description

Use this function to generate the optimal dose escalation and deescalation boundaries for conducting
the trial.

Usage

```
get.boundary(
  target,
  ncohort,
  cohortsize,
  n.earlystop = 100,
  p.saf = 0.6 * target,
  p.tox = 1.4 * target,
  cutoff.eli = 0.95,
  extrasafe = FALSE,
  offset = 0.05
)
```

Arguments

target	the target DLT rate
ncohort	the total number of cohorts
cohortsize	the cohort size
n.earlystop	the early stopping parameter. If the number of patients treated at the current dose reaches n.earlystop, stop the trial and select the MTD based on the observed data. The default value n.earlystop=100 essentially turns off the type of early stopping.
p.saf	the highest toxicity probability that is deemed subtherapeutic (i.e., below the MTD) such that dose escalation should be made. The default value is p.saf = 0.6 * target.

p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is `p.tox=1.4*target`.
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value (`cutoff.eli=0.95`) for general use.
extrasafe	set `extrasafe=TRUE` to impose a more strict stopping rule for extra safety, expressed as the stopping boundary value in the result .
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when `extrasafe=TRUE`. A larger value leads to a more strict stopping rule. The default value (`offset=0.05`) generally works well.

Details

The dose escalation and deescalation boundaries are all we need to run a phase I trial when using the BOIN design. The decision of which dose to administer to the next cohort of patients does not require complicated computations, but only a simple comparison of the observed DLT rate at the current dose with the dose escalation and deescalation boundaries. If the observed DLT rate at the current dose is smaller than or equal to the escalation boundary, we escalate the dose; if the observed toxicity rate at the current dose is greater than or equal to the deescalation boundary, we deescalate the dose; otherwise, we retain the current dose. The dose escalation and deescalation boundaries are chosen to minimize the probability of assigning patients to subtherapeutic or overly toxic doses, thereby optimizing patient ethics. `get.boundary()` also outputs the elimination boundary, which is used to avoid treating patients at overly toxic doses based on the following Bayesian safety rule: if $Pr(p_j > \phi|m_j, n_j) > 0.95$ and $n_j \geq 3$, dose levels j and higher are eliminated from the trial, where p_j is the toxicity probability of dose level j, ϕ is the target DLT rate, and m_j and n_j are the number of toxicities and patients treated at dose level j. The trial is terminated if the lowest dose is eliminated.

The BOIN design has two built-in stopping rules: (1) stop the trial if the lowest dose is eliminated due to toxicity, and no dose should be selected as the MTD; and (2) stop the trial and select the MTD if the number of patients treated at the current dose reaches `n.earlystop`. The first stopping rule is a safety rule to protect patients from the case in which all doses are overly toxic. The rationale for the second stopping rule is that when there is a large number (i.e., `n.earlystop`) of patients assigned to a dose, it means that the dose-finding algorithm has approximately converged. Thus, we can stop the trial early and select the MTD to save the sample size and reduce the trial duration. For some applications, investigators may prefer a more strict safety stopping rule than rule (1) for extra safety when the lowest dose is overly toxic. This can be achieved by setting `extrasafe=TRUE`, which imposes the following more strict safety stopping rule: stop the trial if (i) the number of patients treated at the lowest dose >=3, and (ii) $Pr(toxicity\ rate\ of\ the\ lowest\ dose > \texttt{target}|data) > \texttt{cutoff.eli} - \texttt{offset}$. As a tradeoff, the strong stopping rule will decrease the MTD selection percentage when the lowest dose actually is the MTD.

Value

`get.boundary()` returns a list object, including the dose escalation and de-escalation boundaries `$lambda_e` and `$lambda_d` and the corresponding decision tables `$boundary_tab` and `$full_boundary_tab`. If `extrasafe=TRUE`, the function also returns a (more strict) safety stopping boundary `$stop_boundary`.

Note

We should avoid setting the values of `p.saf` and `p.tox` very close to the `target`. This is because the small sample sizes of typical phase I trials prevent us from differentiating the target DLT rate from the rates close to it. In addition, in most clinical applications, the target DLT rate is often a rough guess, and finding a dose level with a DLT rate reasonably close to the target rate will still be of interest to the investigator. The default values provided by `get.boundary()` are generally reasonable for most clinical applications.

Author(s)

Suyu Liu and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Yuan Y., Hess K.R., Hilsenbeck S.G. and Gilbert M.R. (2016). Bayesian Optimal Interval Design: A Simple and Well-performing Design for Phase I Oncology Trials, *Clinical Cancer Research*, 22, 4291-4301.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
## get the dose escalation and deescalation boundaries for BOIN design with
## the target DLT rate of 0.3, maximum sample size of 30, and cohort size of 3
bound <- get.boundary(target=0.3, ncohort=10, cohortsize=3)
summary(bound) # get the descriptive summary of the boundary
plot(bound)    # plot the flowchart of the design with boundaries
```

---

get.oc   *Generate operating characteristics for single agent trials*

Description

Obtain the operating characteristics of the BOIN design for single agent trials by simulating trials.

Usage

```
get.oc(target, p.true, ncohort, cohortsize, n.earlystop=100,
            startdose=1, titration=FALSE, p.saf=0.6*target, p.tox=1.4*target,
            cutoff.eli=0.95,extrasafe=FALSE, offset=0.05, boundMTD=FALSE,
            ntrial=1000, seed=6)
```

Arguments

target	the target DLT rate
p.true	a vector containing the true toxicity probabilities of the investigational dose levels.
ncohort	the total number of cohorts
cohortsize	the cohort size
n.earlystop	the early stopping parameter. If the number of patients treated at the current dose reaches n.earlystop, stop the trial and select the MTD based on the observed data. The default value n.earlystop=100 essentially turns off this type of early stopping.
startdose	the starting dose level for the trial
titration	set titration=TRUE to perform dose escalation with cohort size = 1 to accelerate dose escalation at the begining of the trial.
p.saf	the highest toxicity probability that is deemed subtherapeutic (i.e. below the MTD) such that dose escalation should be undertaken. The default value is p.saf=0.6*target.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is p.tox=1.4*target).
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value of (cutoff.eli=0.95) for general use.
extrasafe	set extrasafe=TRUE to impose a more stringent stopping rule
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when extrasafe=TRUE. A larger value leads to a more strict stopping rule. The default value offset=0.05 generally works well.
boundMTD	set boundMTD=TRUE to impose the condition: the isotonic estimate of toxicity probability for the selected MTD must be less than de-escalation boundary.
ntrial	the total number of trials to be simulated
seed	the random seed for simulation

Details

The operating characteristics of the BOIN design are generated by simulating trials under the pre-specified true toxicity probabilities of the investigational doses. If titration=TRUE, we perform dose escalation with cohort size = 1 at the begining of the trial: starting from startdose, if no toxicity is observed, we escalate the dose; otherwise, the titration is completed and we switch to cohort size = cohortsize. Titration accelerates the dose escalation and is useful when low doses are believed to be safe.

The BOIN design has two built-in stopping rules: (1) stop the trial if the lowest dose is eliminated due to toxicity, and no dose should be selected as the MTD; and (2) stop the trial and select the MTD if the number of patients treated at the current dose reaches n.earlystop. The first stopping rule is a safety rule to protect patients from the case in which all doses are overly toxic. The rationale for the second stopping rule is that when there is a large number (i.e., n.earlystop) of patients assigned to a dose, it means that the dose-finding algorithm has approximately converged. Thus, we can stop the trial early and select the MTD to save sample size and reduce the trial duration. For some applications, investigators may prefer a more strict safety stopping rule than rule (1) for extra safety when the lowest dose is overly toxic. This can be achieved by setting extrasafe=TRUE, which imposes the following more strict safety stopping rule: stop the trial if (i) the number of patients treated at the lowest dose >=3, and (ii) $Pr(toxicity\ rate\ of\ the\ lowest\ dose > \texttt{target}|data) > \texttt{cutoff.eli} - \texttt{offset}$. As a tradeoff, the strong stopping rule will decrease the MTD selection percentage when the lowest dose actually is the MTD.

Value

get.oc() returns the operating characteristics of the BOIN design as a list, including: (1) selection percentage at each dose level ($selpercent), (2) the number of patients treated at each dose level ($npatients), (3) the number of toxicities observed at each dose level ($ntox), (4) the average number of toxicities ($totaltox), (5) the average number of patients ($totaln), (6) the percentage of early stopping without selecting the MTD ($percentstop), (7) risk of overdosing 60% or more of patients ($overdose60), (8) risk of overdosing 80% or more of patients ($overdose80), (9) data.frame ($simu.setup) containing simulation parameters, such as target, p.true, etc.

Note

We should avoid setting the values of p.saf and p.tox very close to the target. This is because the small sample sizes of typical phase I trials prevent us from differentiating the target DLT rate from the rates close to it. The default values provided by get.oc() are strongly recommended, and generally yield excellent operating characteristics.

Author(s)

Suyu Liu, Yanhong Zhou, and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Yuan Y., Hess K.R., Hilsenbeck S.G. and Gilbert M.R. (2016) Bayesian Optimal Interval Design: A Simple and Well-performing Design for Phase I Oncology Trials, *Clinical Cancer Research*, 22, 4291-4301.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: <http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf>

Examples

```
## get the operating characteristics for BOIN single agent trial
oc <- get.oc(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6),
ncohort=20, cohortsize=3, ntrial=1000)

summary(oc) # summarize design operating characteristics
plot(oc)  # plot flowchart of the BOIN design and design operating characteristics, including
          # selection percentage, number of patients, and observed toxicities at each dose


## perform titration at the begining of the trial to accelerate dose escalation
oc <- get.oc(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6),
titration=TRUE, ncohort=20, cohortsize=3, ntrial=1000)

summary(oc)         # summarize design operating characteristics
plot(oc)  # plot flowchart of the BOIN design and design operating characteristics
```

---

get.oc.comb | *Generate operating characteristics for drug combination trials*

Description

Obtain the operating characteristics of the BOIN design or waterfall design for drug combination trials. The BOIN design is to find a MTD, and the waterfall design is to find the MTD contour (i.e., multple MTDs in the dose matrix)

Usage

```
get.oc.comb(target, p.true, ncohort, cohortsize, n.earlystop=NULL, startdose=c(1, 1),
            titration=FALSE,p.saf=0.6*target, p.tox=1.4*target, cutoff.eli=0.95,
                extrasafe=FALSE,offset=0.05, ntrial=1000, mtd.contour=FALSE,
                boundMTD=FALSE, seed=6)
```

Arguments

target	the target DLT rate
p.true	a J*K matrix (J<=K) containing the true toxicity probabilities of combinations with J dose levels of agent A and K dose levels of agent B
ncohort	a 1*J vector specifying the number of cohorts for each of J subtrials if mtd.contour=TRUE; Otherwise, a scalar specifying the total number of cohorts for the trial.
cohortsize	the cohort size
n.earlystop	the early stopping parameter. If the number of patients treated at the current dose reaches n.earlystop, stop the trial or subtrial and select the MTD based on the observed data. When the waterfall design is used to find the MTD contour, n.earlystop=12 by default.

startdose	the starting dose combination level for drug combination trial
titration	set `titration=TRUE` to perform dose escalation with cohort size = 1 to accelerate dose escalation at the begining of the trial.
p.saf	the highest toxicity probability that is deemed subtherapeutic (i.e. below the MTD) such that dose escalation should be undertaken. The default value is `p.saf=0.6*target`.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is `p.tox=1.4*target`.
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value of (`cutoff.eli=0.95`) for general use.
extrasafe	set `extrasafe=TRUE` to impose a more stringent stopping rule
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when `extrasafe=TRUE`. A larger value leads to a more strict stopping rule. The default value `offset=0.05` generally works well.
ntrial	the total number of trials to be simulated
mtd.contour	set `mtd.contour=TRUE` to select the MTD contour (claiming multiple MTDs). Otherwise, BOIN design is used to search for a single MTD.
boundMTD	set `boundMTD=TRUE` to impose the condition: the isotonic estimate of toxicity probability for the selected MTD must be less than de-escalation boundary.
seed	the random seed for simulation

Details

The operating characteristics of the BOIN design or waterfall design are generated by simulating trials under the prespecified true toxicity probabilities of the investigational dose combinations. If `titration=TRUE`, we perform dose escalation with cohort size = 1 at the begining of the trial: starting from `startdose`, if no toxicity is observed, we escalate the dose; otherwise, the titration is completed and we switch to cohort size = `cohortsize`. Titration accelerates the dose escalation and is useful when low doses are believed to be safe.

The BOIN and waterfall designs have two built-in stopping rules: (1) stop the trial/subtrial if the lowest dose is eliminated due to toxicity, and no dose should be selected as the MTD; and (2) stop the trial/subtrial and select the MTD if the number of patients treated at the current dose reaches `n.earlystop`. The first stopping rule is a safety rule to protect patients from the case in which all doses are overly toxic. The rationale for the second stopping rule is that when there is a large number (i.e., `n.earlystop`) of patients assigned to a dose, it means that the dose-finding algorithm has approximately converged. Thus, we can stop the trial/subtrial early and select the MTD to save sample size and reduce the trial duration.

For some applications, investigators may prefer a more strict safety stopping rule than rule (1) for extra safety when the lowest dose is overly toxic. This can be achieved by setting `extrasafe=TRUE`, which imposes the following more strict safety stopping rule: stop the trial if (i) the number of patients treated at the lowest dose >=3, and (ii) $Pr(toxicity\ rate\ of\ the\ lowest\ dose\ >\ target|data) > $ `cutoff.eli` $-$ `offset`. As a tradeoff, the strong stopping rule will decrease the MTD selection percentage when the lowest dose actually is the MTD.

Value

`get.oc.comb()` returns the operating characteristics of the BOIN combination or waterfall design as a list. For the BOIN combination design, including: (1) true toxicity probability at each dose level (`$p.true`), (2) selection percentage at each dose level (`$selpercent`), (3) the number of patients treated at each dose level (`$npatients`) (4) the number of toxicities observed at each dose level (`$ntox`), (5) the total number of patients in the trial (`$totaln`), (6) the total number of toxicities observed for the trial (`$totaltox`) (7) the pecentage of correct selection (`$pcs`), (8) the total percentage of patients treated at the MTD (`$npercent`). (9) the percentage of early stopping without selecting the MTD (`$percentstop`) For the the waterfall design, including: (1) true toxicity probability at each dose level (`$p.true`), (2) selection percentage of dose combinations (`$selpercent`), (3) the number of patients treated at each dose combination (`$npatients`) (4) the number of toxicities observed at each dose combination (`$ntox`), (5) the total number of patients in the trial (`$totaln`), (6) the total number of toxicities observed for the trial (`$totaltox`) (7) the total percentage of correct selection at the MTD contour (`$pcs.contour`), (8) the total percentage of patients treated at MTD contour (`$npercent.contour`) (9) the total percentage of patients treated above MTD contour (`$npercent.above.contour`) (10) the total percentage of patients treated below MTD contour (`$npercent.below.contour`)

Note

We should avoid setting the values of `p.saf` and `p.tox` very close to the `target`. This is because the small sample sizes of typical phase I trials prevent us from differentiating the target DLT rate from the rates close to it. The default values provided by `get.oc()` are strongly recommended, and generally yield excellent operating characteristics.

Author(s)

Suyu Liu, Liangcai Zhang, Yanhong Zhou, and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Lin R. and Yin, G. (2017). Bayesian Optimal Interval Designs for Dose Finding in Drug-combination Trials, *Statistical Methods in Medical Research*, 26, 2155-2167.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Zhang L. and Yuan, Y. (2016). A Simple Bayesian Design to Identify the Maximum Tolerated Dose Contour for Drug Combination Trials, *Statistics in Medicine*, 35, 4924-4936.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
###### drug-combination trial ######

##### combination trial to find a single MTD  ######

## get the operating characteristics for BOIN design
p.true <- matrix(c(0.01,0.03,0.10,0.20,0.30,
                   0.03,0.05,0.15,0.30,0.60,
                   0.08,0.10,0.30,0.60,0.75), byrow=TRUE, ncol=5)

oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=20, cohortsize=3,
   n.earlystop=12, startdose=c(1,1), ntrial=100)

summary(oc.comb)
plot(oc.comb)


## get the operating characteristics with titration for BOIN design
oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=20, cohortsize=3,
   n.earlystop=12, startdose=c(1,1), titration=TRUE, ntrial=100)
summary(oc.comb)
plot(oc.comb)


##### combination trial to find the MTD contour ######

## find the MTD contour using waterfall design
oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=c(10,5,5), cohortsize=3,
   n.earlystop=12, startdose=c(1,1), ntrial=100, mtd.contour=TRUE)

summary(oc.comb)
plot(oc.comb)
```

---

next.comb	*Determine the dose combination for the next cohort of new patients for drug-combination trials that aim to find a MTD*

Description

Determine the dose combination for the next cohort of new patients for drug-combination trials that aim to find a MTD

Usage

```
next.comb(target, npts, ntox, dose.curr, n.earlystop=100,
                p.saf=0.6*target, p.tox=1.4*target, cutoff.eli=0.95,
                extrasafe=FALSE, offset=0.05)
```

Arguments

target	the target DLT rate
npts	a J*K matrix (J<=K) containing the number of patients treated at each dose combination
ntox	a J*K matrix (J<=K) containing the number of patients experienced dose-limiting toxicity at each dose combination
dose.curr	the current dose combination
n.earlystop	the early stopping parameter. If the number of patients treated at the current dose reaches n.earlystop, stop the trial and select the MTD based on the observed data. The default value n.earlystop=100 essentially turns off this type of early stopping.
p.saf	the highest toxicity probability that is deemed subtherapeutic (i.e. below the MTD) such that dose escalation should be undertaken. The default value is p.saf=0.6*target.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is p.tox=1.4*target.
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value of (cutoff.eli=0.95) for general use.
extrasafe	set extrasafe=TRUE to impose a more stringent stopping rule
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when extrasafe=TRUE. A larger value leads to a more strict stopping rule. The default value offset=0.05 generally works well.

Details

This function is used to determine dose combination for conducting combination trials. Given the currently observed data, next.comb() determines dose combination for treating the next cohort of new patients. The currently observed data include: the number of patients treated at each dose combination (i.e., npts), the number of patients who experienced dose-limiting toxicities at each dose combination (i.e., ntox), and the level of current dose (i.e., dose.curr).

Value

the recommended dose for treating the next cohort of patients ($next_dc).

Author(s)

Suyu Liu and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Lin R. and Yin, G. (2017). Bayesian Optimal Interval Designs for Dose Finding in Drug-combination Trials, *Statistical Methods in Medical Research*, 26, 2155-2167.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
## determine the dose combination for the next cohort of new patients
n <- matrix(c(3, 0, 0, 0, 0, 7, 6, 0, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0), ncol=5, byrow=TRUE)
nxt.comb <- next.comb(target=0.3, npts=n, ntox=y, dose.curr=c(2, 2))
summary(nxt.comb)
```

---

next.subtrial   *Determine the starting dose and the dose-searching space for next sub-trial in waterfall design*

Description

Determine the starting dose and the dose-searching space for next subtrial after the current subtrial is completed when using the waterfall design

Usage

```
next.subtrial(
  target,
  npts,
  ntox,
  p.saf = 0.6 * target,
  p.tox = 1.4 * target,
  cutoff.eli = 0.95,
  extrasafe = FALSE,
  offset = 0.05
)
```

Arguments

target	the target DLT rate
npts	a J*K matrix (J<=K) containing the number of patients treated at each dose combination
ntox	a J*K matrix (J<=K) containing the number of patients who experienced dose-limiting toxicities at each dose combination

p.saf	the highest toxicity probability that is deemed subtherapeutic (i.e. below the MTD) such that dose escalation should be undertaken. The default value is p.saf=0.6*target.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is p.tox=1.4*target.
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value of (cutoff.eli=0.95) for general use.
extrasafe	set extrasafe=TRUE to impose a more stringent stopping rule
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when extrasafe=TRUE. A larger value leads to a more strict stopping rule. The default value offset=0.05 generally works well.

Details

For the waterfall design, this function is used to obtain the starting dose and dose-searching space for the next subtrial when the current subtrial is completed. The input data include: the number of patients treated at each dose combination (i.e., npts), the number of patients who experienced dose-limiting toxicities at each dose combination (i.e., ntox).

Value

next.subtrial() returns a list object, containing the starting dose ($starting_dose) and the dose-searching space for the the next subtrial ($next_subtrial).

Author(s)

Suyu Liu, Liangcai Zhang and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Lin R. and Yin, G. (2017). Bayesian Optimal Interval Designs for Dose Finding in Drug-combination Trials, *Statistical Methods in Medical Research*, 26, 2155-2167.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Zhang L. and Yuan, Y. (2016). A Simple Bayesian Design to Identify the Maximum Tolerated Dose Contour for Drug Combination Trials, *Statistics in Medicine*, 35, 4924-4936.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
## determine the starting dose and dose searching space for next subtrial
n <- matrix(c(6, 0, 0, 0,
              6, 10, 12, 0,
              9, 12, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0,
              1, 1, 4, 0,
              2, 3, 0, 0), ncol=4, byrow=TRUE)
nxt.trial <- next.subtrial(target=0.3, npts=n, ntox=y)
summary(nxt.trial)
```

---

plot.boin   *Plot the flowchart and simulation results for BOIN designs*

Description

Plot the objects returned by other functions, including (1) flowchart of BOIN design; (2) operating characteristics of the design, including selesction percentage and the number of patients treated at each dose; (3) the estimate of toxicity probability for each dose and corresponding 95% credible interval

Usage

```
## S3 method for class 'boin'
plot(x, ..., name = deparse(substitute(x)))
```

Arguments

x	the object returned by other functions
...	ignored arguments
name	the name of the object to be plotted. User doesn't need to input this parameter.

Value

plot() returns a figure or a series of figures depending on the object entered

Author(s)

Suyu Liu, Liangcai Zhang, Yanhong Zhou, and Ying Yuan

Examples

```
###### single-agent trial ######

## get dose escalation and deescalation boundaries for conducting the trial
bound <- get.boundary(target=0.3, ncohort=10, cohortsize=3)
plot(bound)


## get the operating characteristics for BOIN single agent trial
oc <- get.oc(target=0.3, p.true=c(0.05,0.15,0.3,0.45,0.6),
   ncohort=10, cohortsize=3, ntrial=1000)
summary(oc)
plot(oc)


## select the MTD based on the trial data
n <- c(3, 3, 15, 9, 0)
y <- c(0, 0, 4, 4, 0)
selmtd <- select.mtd(target=0.3, npts=n, ntox=y)
summary(selmtd)
plot(selmtd)


###### drug-combination trial ######

##### combination trial to find a single MTD  ######

## get the operating characteristics for BOIN combination trial
p.true <- matrix(c(0.01,0.03,0.10,0.20,0.30,
                0.03,0.05,0.15,0.30,0.60,
                0.08,0.10,0.30,0.60,0.75), byrow=TRUE, ncol=5)

oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=20, cohortsize=3, n.earlystop=12,
     startdose=c(1,1),ntrial=100)
summary(oc.comb)
plot(oc.comb)


## select a MTD based on the trial data
n <- matrix(c(3, 5, 0, 0, 0, 7, 6, 15, 0, 0, 0, 0, 4, 0, 0), ncol=5, byrow=TRUE)
y <- matrix(c(0, 1, 0, 0, 0, 1, 1, 4, 0, 0, 0, 0, 2, 0, 0), ncol=5, byrow=TRUE)
sel.comb <- select.mtd.comb(target=0.3, npts=n, ntox=y)
summary(sel.comb)
plot(sel.comb)


##### combination trial to find a MTD contour (e.g., multiple MTDs)  #####

## get the operating characteristics for BOIN waterfall design
p.true <- matrix(c(0.01, 0.10, 0.20, 0.30,
                0.03, 0.15, 0.30, 0.60,
```

```
                  0.08, 0.30, 0.60, 0.75), byrow=TRUE, ncol=4)

oc.comb2 <- get.oc.comb(target=0.3, p.true, ncohort=c(8,6,6), cohortsize=3, n.earlystop=12,
        startdose=c(1,1), ntrial=100, mtd.contour=TRUE)
summary(oc.comb2)
plot(oc.comb2)


## select the MTD contour based on the trial data
n <- matrix(c(6, 9, 24, 0,  6, 24, 9, 0,  12, 18, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1,  5, 0,  1,  5, 4, 0,  1, 5, 0, 0), ncol=4, byrow=TRUE)
sel.comb2 <- select.mtd.comb(target=0.3, npts=n, ntox=y, mtd.contour=TRUE)
summary(sel.comb2)
plot(sel.comb2)
```

---

print.boin	*Generate descriptive summary for objects returned by other functions*

Description

Generate descriptive summary for objects returned by other functions.

Usage

```
## S3 method for class 'boin'
print(x, ...)
```

Arguments

x   the object returned by other functions

...   ignored arguments

Details

print() prints the objects returned by other functions.

Value

print() prints the objects returned by other functions.

Author(s)

Suyu Liu, Liangcai Zhang, Yanhong Zhou, and Ying Yuan

Examples

```
###### single-agent trial ######

## sprint the object returned by get.boundary()
bound <- get.boundary(target=0.3, ncohort=10, cohortsize=3)
print(bound)


## print the object returned by get.oc()
oc.single <- get.oc(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6), ncohort=10,
cohortsize=3, ntrial=1000)
print(oc.single)


## print the object returned by select.mtd()
n <- c(3, 3, 15, 9, 0)
y <- c(0, 0, 4, 4, 0)
selmtd <- select.mtd(target=0.3, npts=n, ntox=y)
print(selmtd)


###### drug-combination trial######

###### drug-combiation trial to find a single MTD ######

## print the object returned by next.comb()
n <- matrix(c(3, 0, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
nxt.comb <- next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(1, 1))
print(nxt.comb)


## print the object returned by next.comb()
n <- matrix(c(3, 3, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
nxt.comb <- next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(1, 2))
print(nxt.comb)


## print the object returned by get.oc.comb() when mtd.contour=FALSE
p.true <- matrix(c(0.02,0.04,0.08,0.14,
                   0.08,0.25,0.42,0.48,
                   0.25,0.45,0.50,0.60), byrow=TRUE, ncol=4)

oc.comb <- get.oc.comb(target=0.25, p.true=p.true, ncohort=16, cohortsize=3,
            ntrial=100)
print(oc.comb)


## print the object returned by select.mtd.comb()
n <- matrix(c(6, 3, 0, 0, 6, 24, 9, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
```

```
y <- matrix(c(0, 0, 0, 0, 1, 5, 4, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
sel.comb <- select.mtd.comb(target=0.25, npts=n, ntox=y)
print(sel.comb)



###### drug-combiation trial to find the MTD contour ######

## print the object returned by next.subtrial()
n <- matrix(c(6, 0,  0, 0,
              6, 0, 0, 0,
              9, 12, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0,
              1, 0, 0, 0,
              2, 3, 0, 0), ncol=4, byrow=TRUE)
nxt.trial <- next.subtrial(target=0.3, npts=n, ntox=y)
print(nxt.trial)


## print the object returned by get.oc.comb() when mtd.contour=TRUE.
p.true <- matrix(c(0.01,0.03,0.10,0.20,0.30,
                   0.03,0.05,0.15,0.30,0.60,
                   0.08,0.10,0.30,0.60,0.75), byrow=TRUE, ncol=5)

oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=c(10,5,5), cohortsize=3,
   n.earlystop=12, startdose=c(1,1),ntrial=100, mtd.contour=TRUE)
print(oc.comb)


## print the object returned by select.mtd.comb()
n <- matrix(c(6, 9, 24, 0,
              6,  24, 9, 0,
              12, 18, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1, 5, 0,
              1, 5, 4, 0,
              1, 5, 0, 0), ncol=4, byrow=TRUE)
sel.comb2 <- select.mtd.comb(target=0.3, npts=n, ntox=y, mtd.contour=TRUE)
print(sel.comb2)
```

---

select.mtd   *Select the maximum tolerated dose (MTD) for single agent trials*

Description

Select the maximum tolerated dose (MTD) when the single-agent trial is completed

Usage

```
select.mtd(target, npts, ntox, cutoff.eli=0.95, extrasafe=FALSE, offset=0.05,
            boundMTD=FALSE,p.tox=1.4*target)
```

Arguments

target	the target DLT rate
npts	a vector containing the number of patients treated at each dose level
ntox	a vector containing the number of patients who experienced dose-limiting toxicity at each dose level
cutoff.eli	the cutoff to eliminate overly toxic doses for safety. We recommend the default value of (cutoff.eli=0.95) for general use.
extrasafe	set extrasafe=TRUE to impose a more strict stopping rule for extra safety
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when extrasafe=TRUE. A larger value leads to a more strict stopping rule. The default value offset=0.05 generally works well.
boundMTD	set boundMTD=TRUE to impose the condition: the isotonic estimate of toxicity probability for the selected MTD must be less than de-escalation boundary.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is p.tox=1.4*target.

Details

select.mtd() selects the MTD based on isotonic estimates of toxicity probabilities. select.mtd() selects as the MTD dose j^*, for which the isotonic estimate of the DLT rate is closest to the target. If there are ties, we select from the ties the highest dose level when the estimate of the DLT rate is smaller than the target, or the lowest dose level when the estimate of the DLT rate is greater than the target. The isotonic estimates are obtained by the pooled-adjacent-violators algorithm (PAVA) (Barlow, 1972).

Value

select.mtd() returns (1) target toxicity probability ($target), (2) selected MTD ($MTD), (3) isotonic estimate of the DLT probablity at each dose and associated 95% credible interval ($p_est), and (4) the probability of overdosing defined as $Pr(toxicity > \texttt{target}|data)$ ($p_overdose)

Note

The MTD selection and dose escalation/deescalation rule are two independent components of the trial design. When appropriate, another dose selection procedure (e.g., based on a fitted logistic model) can be used to select the MTD after the completion of the trial using the BOIN design.

Author(s)

Suyu Liu, Yanhong Zhou, and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Yuan Y., Hess K.R., Hilsenbeck S.G. and Gilbert M.R. (2016). Bayesian Optimal Interval Design: A Simple and Well-performing Design for Phase I Oncology Trials, *Clinical Cancer Research*, 22, 4291-4301.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
### select the MTD for BOIN single agent trial
n <- c(3, 3, 15, 9, 0)
y <- c(0, 0, 4, 4, 0)
selmtd <- select.mtd(target=0.3, npts=n, ntox=y)
summary(selmtd)
plot(selmtd)
```

---

select.mtd.comb	*Select the maximum tolerated dose (MTD) or MTD contour for drug combination trials*

Description

Select the maximum tolerated dose (MTD) or MTD contour after the drug combination trial is completed using the BOIN design or waterfall design

Usage

```
select.mtd.comb(
  target,
  npts,
  ntox,
  cutoff.eli = 0.95,
  extrasafe = FALSE,
  offset = 0.05,
  boundMTD = FALSE,
  p.tox = 1.4 * target,
  mtd.contour = FALSE
)
```

Arguments

target	the target DLT rate
npts	a J*K matrix (J<=K) containing the number of patients treated at each dose combination
ntox	a J*K matrix (J<=K) containing the number of patients experienced dose-limiting toxicity at each dose combination
cutoff.eli	the cutoff to eliminate an overly toxic dose for safety. We recommend the default value of (cutoff.eli=0.95) for general use.
extrasafe	set extrasafe=TRUE to impose a more strict stopping rule for extra safety
offset	a small positive number (between 0 and 0.5) to control how strict the stopping rule is when extrasafe=TRUE. A larger value leads to a more strict stopping rule. The default value offset=0.05 generally works well.
boundMTD	set boundMTD=TRUE to impose the condition: the isotonic estimate of toxicity probability for the selected MTD must be less than de-escalation boundary.
p.tox	the lowest toxicity probability that is deemed overly toxic such that deescalation is required. The default value is p.tox=1.4*target.
mtd.contour	set mtd.contour=TRUE to select the MTD contour, otherwise select a single MTD. The value of mtd.contour should be consistent with that in get.oc.comb().

Details

select.mtd.comb() selects a MTD or the MTD contour based on matrix isotonic estimates of toxicity probabilities, depending on mtd.contour is set as TRUE or FALSE. The (matrix) isotonic estimates are obtained by the R package (Iso::biviso).

Value

select.mtd.comb() returns returns (1) target toxicity probability ($target), (2) selected MTD or MTD contour ($MTD), (3) isotonic estimate of the DLT probablity at each dose ($p_est).

Note

The MTD selection and dose escalation/deescalation rule are two independent components of the trial design. When appropriate, another dose selection procedure (e.g., based on a fitted logistic model) can be used to select the MTD after the completion of the trial using the BOIN or waterfall design.

Author(s)

Suyu Liu, Liangcai Zhang, Yanhong Zhou, and Ying Yuan

References

Liu S. and Yuan, Y. (2015). Bayesian Optimal Interval Designs for Phase I Clinical Trials, *Journal of the Royal Statistical Society: Series C*, 64, 507-523.

Lin R. and Yin, G. (2017). Bayesian Optimal Interval Designs for Dose Finding in Drug-combination Trials, *Statistical Methods in Medical Research*, 26, 2155-2167.

Yan, F., Zhang, L., Zhou, Y., Pan, H., Liu, S. and Yuan, Y. (2020).BOIN: An R Package for Designing Single-Agent and Drug-Combination Dose-Finding Trials Using Bayesian Optimal Interval Designs. *Journal of Statistical Software*, 94(13),1-32.<doi:10.18637/jss.v094.i13>.

Zhang L. and Yuan, Y. (2016). A Simple Bayesian Design to Identify the Maximum Tolerated Dose Contour for Drug Combination Trials, *Statistics in Medicine*, 35, 4924-4936.

See Also

Tutorial: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/BOIN2.6_tutorial.pdf

Paper: http://odin.mdacc.tmc.edu/~yyuan/Software/BOIN/paper.pdf

Examples

```
### drug-combination trial to find a single MTD

## Select the MTD based on the data from a 3x5 combination trial
## matrix n contains the number of patients treated at each dose combination
## matrix y contains the number of patients experienced toxicity at each dose combination
n <- matrix(c(3, 5, 0, 0, 0, 7, 6, 15, 0, 0, 0, 0, 4, 0, 0), ncol=5, byrow=TRUE)
y <- matrix(c(0, 1, 0, 0, 0, 1, 1, 4, 0, 0, 0, 0, 2, 0, 0), ncol=5, byrow=TRUE)
sel.comb <- select.mtd.comb(target=0.3, npts=n, ntox=y)
summary(sel.comb)
plot(sel.comb)


### drug-combination trial to find the MTD contour

## Select the MTD contour based on the data from a 3x4 combination trial
## matrix n contains the number of patients treated at each dose combination
## matrix y contains the number of patients experienced toxicity at each dose combination
n <- matrix(c(6, 9, 24, 0,  6, 24, 9, 0,  12, 18, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1,  5, 0,  1,  5, 4, 0,   1, 5, 0, 0), ncol=4, byrow=TRUE)
sel.comb2 <- select.mtd.comb(target=0.3, npts=n, ntox=y, mtd.contour=TRUE)
summary(sel.comb2)
plot(sel.comb2)
```

---

summary.boin	*Generate descriptive summary for objects returned by other functions*

Description

Generate descriptive summary for objects returned by other functions.

Usage

```
## S3 method for class 'boin'
summary(object, ...)
```

Arguments

object   the object returned by other functions.

...   ignored arguments

Details

summary() prints the objects returned by other functions.

Value

summary() prints the objects returned by other functions.

Author(s)

Suyu Liu, Liangcai Zhang and Ying Yuan

Examples

```
###### single-agent trial ######

## summarize the object returned by get.boundary()
bound <- get.boundary(target=0.3, ncohort=10, cohortsize=3)
summary(bound)


## summarize the object returned by get.oc()
oc.single <- get.oc(target=0.3, p.true=c(0.05, 0.15, 0.3, 0.45, 0.6), ncohort=10,
cohortsize=3, ntrial=1000)
summary(oc.single)


## summarize the object returned by select.mtd()
n <- c(3, 3, 15, 9, 0)
y <- c(0, 0, 4, 4, 0)
selmtd <- select.mtd(target=0.3, npts=n, ntox=y)
summary(selmtd)


###### drug-combination trial######

###### drug-combiation trial to find a single MTD ######

## summarize the object returned by next.comb()
n <- matrix(c(3, 0, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
nxt.comb <- next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(1, 1))
summary(nxt.comb)


## summarize the object returned by next.comb()
```

```
n <- matrix(c(3, 3, 0, 0, 0, 0, 0, 0,  0, 0, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
nxt.comb <- next.comb(target=0.25, npts=n, ntox=y, dose.curr=c(1, 2))
summary(nxt.comb)


## summarize the object returned by get.oc.comb() when mtd.contour=FALSE
p.true <- matrix(c(0.02,0.04,0.08,0.14,
                   0.08,0.25,0.42,0.48,
                   0.25,0.45,0.50,0.60), byrow=TRUE, ncol=4)

oc.comb <- get.oc.comb(target=0.25, p.true=p.true, ncohort=16, cohortsize=3,
            ntrial=100)
summary(oc.comb)


## summarize the object returned by select.mtd.comb()
n <- matrix(c(6, 3, 0, 0, 6, 24, 9, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0, 1, 5, 4, 0, 0, 0, 0, 0), ncol=4, byrow=TRUE)
sel.comb <- select.mtd.comb(target=0.25, npts=n, ntox=y)
summary(sel.comb)



###### drug-combiation trial to find the MTD contour ######

## summarize the object returned by next.subtrial()
n <- matrix(c(6, 0,  0, 0,
              6, 0, 0, 0,
              9, 12, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 0, 0, 0,
              1, 0, 0, 0,
              2, 3, 0, 0), ncol=4, byrow=TRUE)
nxt.trial <- next.subtrial(target=0.3, npts=n, ntox=y)
summary(nxt.trial)


## summarize the object returned by get.oc.comb() when mtd.contour=TRUE.
p.true <- matrix(c(0.01,0.03,0.10,0.20,0.30,
                   0.03,0.05,0.15,0.30,0.60,
                   0.08,0.10,0.30,0.60,0.75), byrow=TRUE, ncol=5)

oc.comb <- get.oc.comb(target=0.3, p.true, ncohort=c(10,5,5), cohortsize=3,
    n.earlystop=12, startdose=c(1,1),ntrial=100, mtd.contour=TRUE)
summary(oc.comb)


## summarize the object returned by select.mtd.comb()
n <- matrix(c(6, 9, 24, 0,
              6,  24, 9, 0,
              12, 18, 0, 0), ncol=4, byrow=TRUE)
y <- matrix(c(0, 1, 5, 0,
              1, 5, 4, 0,
```

```
            1, 5, 0, 0), ncol=4, byrow=TRUE)
sel.comb2 <- select.mtd.comb(target=0.3, npts=n, ntox=y, mtd.contour=TRUE)
summary(sel.comb2)
```

Index