Package 'tsdf'

May 8, 2026

Type Package

Title Two-/Three-Stage Designs for Phase 1&2 Clinical Trials

Version 1.1-9

Date 2026-04-25

Description Calculates Zhong's optimal two-/three-stage Phase II designs for single-arm trials, generates target-toxicity decision tables for two-/three-stage Phase I dose-finding, and supports dose-finding simulations using custom decision tables. The Phase II design is based on Zhong (2012) <doi:10.1016/j.cct.2012.07.006>.

License GPL-2

Encoding UTF-8

RoxygenNote 7.3.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Wenchuan Guo [aut, cre],
Jianan Hui [aut],
Bob Zhong [aut]

Maintainer Wenchuan Guo <wguo1017@gmail.com>

Repository CRAN

Date/Publication 2026-04-26 00:30:02 UTC

Contents

`adj.two` *Adjust Zhong's two-stage Phase II design*

Description

Adjust Zhong's two-stage design for over-running or under-running.

Usage

```
adj.two(n1, r1, s1, n2, alpha1, alpha2, beta, pc, pe, ...)
```

Arguments

`n1`	Sample size at stage 1.
`r1`	Inefficacy boundary at stage 1.
`s1`	Efficacy boundary at stage 1. If there is no early stopping for efficacy, `s1` should equal `n1`.
`n2`	Sample size at stage 2.
`alpha1`	Left-side overall type I error.
`alpha2`	right-side overall type I error.
`beta`	Type II error.
`pc`	A numeric vector of response rates. It should have length 1 or 2.
`pe`	Alternative hypothesis response rate.
`...`	Unused arguments.

Details

This function enumerates feasible second-stage boundaries after the first-stage sample size and boundaries have been fixed.

Value

An object of class `"opt.design"`, returned as a list containing:

`bdry`	The rejection boundaries.
`error`	The true type I and type II errors.
`n`	The sample size at each stage.
`complete`	The complete list of feasible designs.
`alpha1`	The input left-side type I error.

alpha2	The input right-side type I error.
beta	The input type II error.
pc	The input response-rate vector.
pe	The input alternative response rate.
stage	The number of stages in the selected design.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>, Jianan Hui <jiananhuistat@gmail.com>

Examples

```
n1 <- 22
r1 <- 6
s1 <- 22
n2 <- 24
pc <- 0.4
pe <- pc + 0.15
alpha1 <- 0.3
alpha2 <- 0.1
beta <- 0.2
out <- adj.two(n1, r1, s1, n2, alpha1, alpha2, beta, pc, pe)
```

---

dec.sim *Run dose-finding simulations*

Description

Run dose-finding simulations based on a customized decision table.

Usage

```
dec.sim(truep, decTable, start.level = 1, nsim = 1000)
```

Arguments

truep	A vector of length k, giving the true toxicity probabilities for the dose levels under study.
decTable	A customized decision table in the same format as the output of dec.table.
start.level	Starting dose level. Defaults to 1, the lowest dose level.
nsim	Number of simulated trials. Defaults to 1000.

Details

Assume there are d dose levels to be studied. Let n_i and m_i denote the cumulative number of patients treated and the cumulative number of DLTs observed at the current dose level, respectively. Let n_{max} be the maximum number of patients allowed at each dose level. The procedure is as follows:

Step 1 Update the cumulative numbers of DLTs (m_i) and total treated patients (n_i) at the current dose, then use the decision table to choose an action. If the decision is `"S"`, go to Step 2. If the decision is `"D"` or `"DU"`, go to Step 3. If the decision is `"E"`, go to Step 4.

Step 2 If $n_i = n_{max}$, declare dose i as the MTD. Otherwise, treat an additional cohort at the current dose and return to Step 1.

Step 3 If the current dose is the lowest dose, stop the trial and declare that the MTD is below the lowest dose. Otherwise, if the next lower dose has not yet reached n_{max}, treat an additional cohort there, move to that dose, and return to Step 1. If the next lower dose has already reached n_{max}, stop the trial and declare that dose as the MTD. If the decision is `"DU"`, record the current dose as unusable and do not treat additional patients there.

Step 4 If the current dose is the highest dose, stop the trial and declare that the MTD is above the highest dose. Otherwise, if the next higher dose has been marked `"DU"`, continue treating the current dose until it reaches n_{max}; if that limit is reached, declare the current dose as the MTD. If the next higher dose is available and has not yet reached n_{max}, treat an additional cohort there, move up to that dose, and return to Step 1. Otherwise, declare the current dose as the MTD.

Value

An object of class `"dec.sim"`. Use `summary.dec.sim` to obtain and print a summary table of the results. The returned object is a list containing:

`mtd`	A vector of dose levels giving the recommended maximum tolerated dose (MTD) at the end of the trial.
`mtd.prob`	A vector of length k giving the proportion of trials that selected each dose level as the MTD.
`over.prob`	A vector of length k giving the proportion of trials that selected each dose level as above the MTD.
`n.patients`	The average number of patients treated at each dose level.
`dlt`	The average number of DLTs observed at each dose level.
`truep`	The input true toxicity probabilities.
`start.level`	The input starting dose level.
`nsim`	The input number of simulated trials.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>

Examples

```
truep <- c(0.3, 0.45, 0.5, 0.6)
res <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
out <- dec.sim(truep, as.matrix(res$table), start.level = 2, nsim = 1000)
summary(out, pt = 0.3)
```

`dec.table` *Generate a dose-finding decision table*

Description

Generate a two- or three-stage dose-finding decision table.

Usage

```
dec.table(alpha.l, alpha.r, alpha.u, pt, n, sf.param = 4, pe.par = 0.25, ...)
```

Arguments

`alpha.l`	Left-side overall type I error. This controls the upper bound for dose escalation.
`alpha.r`	Right-side overall type I error. This controls the lower bound for dose de-escalation.
`alpha.u`	Right-side overall type I error used to determine the lower bound for `"DU"`.
`pt`	A numeric vector of target toxicities. It should have length 1, or length 2 when the target is an interval.
`n`	A vector of sample sizes at each stage. `sum(n)` is the total sample size. For A+B designs, `n` has length 2; for A+B+C designs, `n` has length 3.
`sf.param`	A single real value specifying the gamma parameter for the Hwang-Shih-DeCani spending function. The allowable range is [-40, 40]. Larger values spend more error early and leave less for later stages. Defaults to 4.
`pe.par`	Alternative hypothesis offset used to calculate power and the type II error. The alternative is defined as `pe = pt + pe.par`. Defaults to 0.25.
`...`	Unused arguments.

Details

An alpha-spending method is available for two- and three-stage designs. `dec.table` uses the Hwang-Shih-DeCani spending function.

Value

An object of class "dec.table", returned as a list containing:

table	The generated decision table.
alpha.two	A vector of true type I errors for the two-sided test.
alpha.one	A vector of true type I errors for the right-sided test.
beta	The true type II error, which depends on the alternative hypothesis.
E	A vector of "E" boundaries.
D	A vector of "D" boundaries.
DU	A vector of "DU" boundaries.
pt	The input target toxicity vector.
n	The input stage-wise sample sizes.
sf.param	The input alpha-spending parameter.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>

Examples

```
alpha.l <- 0.6
alpha.r <- 0.4
alpha.u <- 0.1
pt <- 0.3
# Print the decision table for a 3+3+3 design
n <- rep(3, 3)
dec.table(alpha.l, alpha.r, alpha.u, pt, n)$table
# 3+3 design
n <- rep(3, 2)
dec.table(alpha.l, alpha.r, alpha.u, pt, n)$table
```

---

opt.design *Zhong's 2- or 3-stage Phase II design*

Description

Calculate the optimal 2- or 3-stage design proposed by Bob Zhong.

Usage

```
opt.design(
  alpha1,
  alpha2,
  beta,
  pc,
  pe,
  stage = 2,
  stop.eff = FALSE,
  frac_n1 = NULL,
  frac_n2 = NULL,
  sf.param = NULL,
  show = FALSE,
  nmax = 100,
  n.choice = 1,
  ...
)
```

Arguments

alpha1	Left-side overall type I error.
alpha2	right-side overall type I error.
beta	Type II error.
pc	A numeric vector of response rates. It should have length 1 or 2.
pe	Alternative hypothesis response rate.
stage	Either 2 or 3. Defaults to 2.
stop.eff	Logical; if TRUE, the trial may stop early for efficacy at an interim analysis.
frac_n1	Proportion range for n1. For a two-stage design, the default is c(0.3, 0.6). For a three-stage design, the default is c(0.2, 0.3).
frac_n2	Proportion range for n2. Used only for three-stage designs. Defaults to c(0.2, 0.4).
sf.param	A single real value specifying the gamma parameter for the Hwang-Shih-DeCani spending function. The allowable range is [-40, 40]. Larger values spend more error early and leave less for later stages. For two-stage designs, the default is NULL (no alpha-spending). For three-stage designs, the default is 4.
show	Logical; if TRUE, the current total sample size is printed during the search.
nmax	Maximum sample size. Defaults to 100.
n.choice	Stopping criterion for the search over feasible designs. The search stops once the number of designs exceeds n.choice.
...	Unused arguments.

Details

In the two-stage design, n1 patients are treated in the first stage. At the end of stage 1, the trial either continues to stage 2 or stops early for inefficacy, depending on the number of observed responses. If the trial continues, an additional n2 patients are treated. The three-stage design extends the two-stage design by adding one interim stage between stages 1 and 2. The left-side rejection region is defined by response <= r_i for i = 1, 2, 3, and the right-side rejection region is defined by response > s. An alpha-spending method is available for both two- and three-stage designs. opt.design uses the Hwang-Shih-DeCani spending function; you can change the definition of HSD to use a different spending function.

Value

An object of class "opt.design", returned as a list containing:

bdry	The rejection boundaries.
error	The true type I and type II errors.
n	The sample size at each stage.
complete	The complete list of feasible designs.
alpha1	The input left-side type I error.
alpha2	The input right-side type I error.
beta	The input type II error.
pc	The input response-rate vector.
pe	The input alternative response rate.
sf.param	The input alpha-spending parameter.
stage	The number of stages in the selected design.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>, Jianan Hui <jiananhuistat@gmail.com>

References

Zhong. (2012) Single-arm Phase IIA clinical trials with go/no-go decisions. *Contemporary Clinical Trials*, **33**, 1272–1279.

Examples

```
alpha1 <- 0.15
alpha2 <- 0.10
beta <- 0.15
pc <- 0.25
pe <- pc + 0.20
# calculate optimal two-stage design without using alpha-spending
opt.design(alpha1, alpha2, beta, pc, pe, stage=2)

# calculate optimal two-stage design with Pocock-like spending function
opt.design(alpha1, alpha2, beta, pc, pe, stage = 2, sf.param = 1)
```

```
# calculate optimal three-stage design with an O'Brien-Fleming-like spending function
opt.design(alpha1, alpha2, beta, pc, pe, stage = 3, sf.param = -4)
```

`plot.dec.sim` *Plot simulation results from a "dec.sim" object*

Description

Available plot types are: true toxicity at each dose level (`type = "s"`); a bar plot of the probability of selecting each dose as the MTD (`type = "prob"`); a bar plot of the average number of patients treated at each dose level (`type = "np"`); and a bar plot of the average number of DLTs at each dose level (`type = "dlt"`). Setting `type = "all"` produces all four plots.

Usage

```
## S3 method for class 'dec.sim'
plot(
  x,
  pt,
  s = 1,
  type = c("all", "s", "prob", "np", "dlt"),
  label = TRUE,
  col = "cornflowerblue",
  text.col = "darkblue",
  cex = 1,
  ...
)
```

Arguments

`x`	An object of class `"dec.sim"` or `"sl.sim"`, returned by `dec.sim` or `sl.sim`.
`pt`	A vector of target toxicity values, one for each scenario.
`s`	The scenario to plot. Defaults to 1.
`type`	Plot type. See the description above.
`label`	Logical; if `TRUE`, values are displayed on the plot.
`col`	Graphical parameter `col`; see `par`.
`text.col`	Color used for text labels.
`cex`	Graphical parameter `cex`; see `par`.
`...`	Arguments passed to plotting functions.

Examples

```
# generate decision table
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
# Simulate trials from test data
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
out <- sl.sim(dt$table, test.file)
plot(out, pt=rep(0.3,2), s=1, type="all")
plot(out, pt=rep(0.3,2), s=2, type="prob")
plot(out, pt=rep(0.3,2), s=1, type="np")
plot(out, pt=rep(0.3,2), s=2, type="dlt")
```

`plot.dec.table` *Plot a decision table from a "dec.table" object*

Description

`plot` method for class `"dec.table"`.

Usage

```
## S3 method for class 'dec.table'
plot(x, ...)
```

Arguments

`x`	An object of class `"dec.table"`, typically returned by `dec.table`.
`...`	Unused arguments.

Details

`plot.dec.table` plots the decision boundaries.

Examples

```
truep <- c(0.3, 0.45, 0.5, 0.6)
out <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
plot(out)
```

print.dec.table *Print a decision table from a "dec.table" object*

Description

print method for class "dec.table".

Usage

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

Arguments

x	An object of class "dec.table", typically returned by dec.table.
...	Unused arguments.

Details

print.dec.table prints the decision table together with a legend.

Examples

```
print(dec.table(0.6,0.4,0.2,0.3,c(3,3,3)))
```

print.opt.design *Print Zhong's design from an "opt.design" object*

Description

print method for class "opt.design".

Usage

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

Arguments

x	An object of class "opt.design", typically returned by opt.design.
...	Unused arguments.

Examples

```
alpha1 <- 0.20
alpha2 <- 0.1
beta <- 0.20
pc <- 0.5
pt <- pc + 0.2
out <- opt.design(alpha1, alpha2, beta, pc, pt, stage = 2, sf.param = 1)
print(out)
```

---

sl.sim *Dose-finding simulations for a list of scenarios*

Description

Run dose-finding simulations based on a customized decision table for one or more scenarios read from a file.

Usage

```
sl.sim(decTable, file, header = TRUE, sep = ",", ...)
```

Arguments

decTable	A customized decision table in the same format as the output of dec.table.
file	The name of the file containing the scenario definitions. See read.table for details.
header	Logical; if TRUE, the file contains variable names in the first line. See read.table for details.
sep	The field separator character. Defaults to ",". See read.table for details.
...	Additional arguments passed to read.table.

Details

In each row of the input file, the parameters must be ordered as start.level, nsim, and truep. The dose-finding algorithm is described in dec.sim.

Value

An object of class "dec.sim" (for one scenario) or "sl.sim" (for multiple scenarios). Use summary to obtain and print a summary of the results. The returned object is a list containing:

mtd	A vector of dose levels giving the recommended maximum tolerated dose (MTD) at the end of each trial.
n.patients	The average number of patients treated at each dose level.
truep	The input true toxicity probabilities.
start.level	The input starting dose level.
nsim	The input number of simulated trials.

Author(s)

Wenchuan Guo <wguo1017@gmail.com>

Examples

```
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
# use a customized decision table
table.file <- system.file("extdata", "decTable.csv", package = "tsdf")
dec <- read.csv(table.file, row.names = 1, check.names = FALSE)
out1 <- sl.sim(as.matrix(dt$table), test.file)
out2 <- sl.sim(dec, test.file)
```

---

summary.dec.sim *Summarize simulation results from a "dec.sim" object*

Description

summary method for class "dec.sim".

Usage

```
## S3 method for class 'dec.sim'
summary(object, pt, ...)
```

Arguments

object	An object of class "dec.sim", returned by dec.sim or sl.sim.
pt	Target toxicity for each scenario.
...	Unused arguments.

Details

summary formats key statistics from dose-finding simulations. Given the target toxicity, it reports the probability of selecting each dose level as the MTD, the probability of selecting doses above the true MTD, the probability of selecting the true MTD, and the probability that subjects are treated at or below the true MTD. The true MTD is defined as the highest dose level with toxicity probability less than or equal to the target toxicity. If the target is below the smallest toxicity probability, the lowest dose level is treated as the MTD. For example, if the target is 0.3 and the true toxicity probabilities for five doses are 0.1, 0.25, 0.35, 0.40, and 0.50, then the MTD is dose 2.

Examples

```
test.file <- system.file("extdata", "testS.csv", package = "tsdf")
dt <- dec.table(0.6,0.4,0.2,0.3,c(3,3,3))
out <- sl.sim(dt$table, test.file)
pt <- c(0.3, 0.4)
summary(out, pt)
```

Index