

Package: AdaptiveDesignSim (via r-universe)

September 23, 2026

Title Simulations of clinical trials with adaptive designs
Version 0.2.0
Description Simulates trials with adaptive designs and calculates
opering characteriscs.
License MIT + file LICENSE
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 8.0.0
Depends R (>= 2.10)
Imports pairwiseCI, stats, shiny
Config/pak/sysreqs cmake make libuv1-dev zlib1g-dev
Repository <https://dcr-unibe-ch.r-universe.dev>
Date/Publication 2026-06-11 16:06:49 UTC
RemoteUrl <https://github.com/dcr-unibe-ch/AdaptiveDesignSim>
RemoteRef HEAD
RemoteSha 5b77c1cadbb3c04c7c85fa707521705a3698eedd

Contents

ADeval	2
ADsim	3
launch_ADsim_app	5
Index	6

ADeval

ADeval

Description

Function to evaluate operating characteristics of trials simulated via ADsim

Usage

```
ADeval(simdat, zcutoff, finalm = c("z", "ci"), alpha = 0.025)
```

Arguments

<code>simdat</code>	matrix or data.frame of multiple runs of ADsim
<code>zcutoff</code>	a matrix (or data frame) with the cut-offs on the z-statistics at interim. One row per interim analysis and a column for stop for futility and efficacy (with NA if not planned)
<code>finalm</code>	Measure used for the final analysis, either z (z-statistics) or ci (confidence interval).
<code>alpha</code>	Alpha level used for the final analysis if <code>finalm=="z"</code> , otherwise the ci given in <code>simdat</code> is used.

Value

A list with the evaluation for each repetition (**rawdata**) and the summarized operating characteristics (**opchar**). **opchar** includes the average estimates in each group, effect (point estimate, **pe**) and (1-alpha) confidence limits (**lci**, **uci**), and

- **psig**: the probability for a significant trial (the power or type I error, depending on the assumptions).
- **pstop**: the probability for any stop and a stop for futility or efficacy (**pstop_fut**, **pstop_eff**).
- the average, minimum and maximum sample size (**avn**, **minn**, **maxn**).
- **bias**: the average of estimate minus true value.
- **sd**: the standard deviation of the point estimate.
- **mse**: the mean squared error, the average of the squared difference between estimate and true value.
- **coverage**: the proportion of confidence intervals including the true value.

Examples

```
#Simulate 1000 trials with one interim analysis
simdat <- lapply(1:1000, function(i)
  ADsim(n01 = c(100,100), par01 = c(0.2, 0.4), nia = 1))
simdat <- do.call(rbind, simdat)

# Define stop for futility stoppin (z<0) and efficacy stopping (z>2)
```

```
zcutoff<-matrix(c(0,2),1,2)
sr<-ADeval(simdat = simdat, zcutoff = zcutoff)
head(sr$rawdata)
sr$opchar
```

ADsim

ADsim

Description

Simulates trials with a binary endpoint and one or several interim analysis,

Usage

```
ADsim(
  n01,
  nia,
  tia = NULL,
  par01,
  simfun = "binom",
  estfun = "rd_wald",
  cilevel = 0.95,
  truefun = NULL,
  direct = c("higher", "lower")
)
```

Arguments

n01	Total sample size in control and experimental arm, numeric vector of length 2.
nia	Number of interim analysis, not including the final analysis.
tia	Optional vector with the time point(s) of the interim analyses, as fraction of the total (between 0 and 1).
par01	Parameters in control and experimental arm (which is used as argument for simfun).
simfun	Simulation function or one of the default options ("binom", "norm").
estfun	Function to calculate point estimate, CIs and z- statistic or one of the five defaults functions ("rd_wald", "lrr_wald", "rd_score", "lrr_score", "md_t").
cilevel	Level of the two-sided confidence interval for the default estfun.
truefun	Function to calculate the true point estimate from par01 (ignored if the default estfun are used).
direct	Direction of the effect, "higher" (default) or "lower" values are better.

Details

Parameters for simulation are defined with *par01*, the simulation function with *simfun*. For the default *simfun* "binom" *par01* must be a numeric vector of length 2 with the probabilities in control and experimental group. For the default *simfun* "norm", *par01* must be a 2x2 data frame with rows *mean* and *sd* and first column for control and second for experimental. User-specified *simfun*, must take *par01* as argument.

For estimation, five defaults can be specified via *estfun* - standard Wald-type approximation for the risk difference (*wald_rd*) and risk ratio (*wald_rr*), the score-based method for the risk difference suggested by Newcombe (*score_rd*), the score-based method for the risk ratio according to Koopman (*score_rr*), and Student's t-test for the mean difference (*md_t*). The effect is always given as experimental (second element of *par01*) vs control (first element of *par01*).

User-specified *estfun* must take the output from *simfun* and *cilevel* as argument, and produce a named numeric vector with point estimate (*pe*), confidence limits (*lci*, *uci*) and z-statistic (*z*). If a user-specified *estfun* is used, a function to calculate the true effect must be given via *truefun*.

The direction if the effect in *direct* only affects the sign of the z-statistic. So that positive values of the z-statistic are always associated with a benefit of the experimental arm.

Value

Numeric vector with true point estimate (*true_pe*), and for each interim stage (prefix *iax_*) and the final stage (prefix *fa_*), the sample size (suffix *n0* and *n1*), point estimate (suffix *pe*), confidence interval (suffix *lci* and *uci*) and z-statistic (*z*). And depending on the setting further group-specific parameters (e.g. the simulated number of successes and observed probabilities for both arms).

Examples

```
set.seed(1)
ADsim(n01 = c(100,100), par01 = c(0.2,0.4), nia = 1)

#Earlier interim analysis
set.seed(1)
ADsim(n01 = c(100,100), par01 = c(0.2,0.4), nia = 1, tia = 0.4)

#Difference effect estimation
set.seed(1)
ADsim(n01 = c(100,100), par01 = c(0.2,0.4), nia = 1, estfun = "lrr_wald")

#Assume higher would be better
set.seed(1)
ADsim(n01 = c(100,100), par01 = c(0.2,0.4), nia = 2, direct="lower")

#User defined function that reproduces the default setting
efun<-function(out,cilevel) {
  xs<-unlist(lapply(out,sum))
  ns<-unlist(lapply(out,length))
  pe<-diff(xs/ns)
  se<-sqrt(sum(xs/ns*(1-xs/ns)/ns))
  lci<-pe - qnorm(1-(1-cilevel)/2)*se
  uci<-pe + qnorm(1-(1-cilevel)/2)*se
```

```

z<-(-pe)/se
res<-c(pe,lci,uci,z)
names(res)<-c("pe","lci","uci","z")
return(res)
}

tfun <- function(x) diff(x)

set.seed(1)
ADsim(n01 = c(100,100), par01 = c(0.4,0.2), nia = 1, estfun = efun, truefun = tfun)

#Continuous outcome
par01<-data.frame(matrix(c(5,1,5.5,1),2,2))
rownames(par01)<-c("mean","sd")
ADsim(n01 = c(100,100), par01 = par01, nia = 1, estfun = "md_t", simfun="norm")

```

launch_ADSim_app

AdaptiveDesignSim shiny app

Description

The shiny app to AdaptiveDesignSim provides an intuitive way to

1. check the results of individual simulations (via forest plots) and
2. of operating characteristics over a range of conditions.

Usage

```
launch_ADSim_app()
```

Examples

```

# launch the app
## Not run:
launch_ADSim_app()

## End(Not run)

```

Index

ADeval, [2](#)

ADsim, [3](#)

launch_ADsim_app, [5](#)