Package 'CohortPlat'

July 21, 2025

Type Package

Title Simulation of Cohort Platform Trials for Combination Treatments

Version 1.0.5

Author Elias Laurin Meyer [aut, cre],
Peter Mesenbrink [ctb],
Cornelia Dunger-Baldauf [ctb],
Ekkehard Glimm [ctb],
Franz Koenig [ctb]

Maintainer Elias Laurin Meyer <elias.meyer@meduniwien.ac.at>

Description A collection of functions dedicated to simulating staggered entry platform trials whereby the treatment under investigation is a combination of two active compounds. In order to obtain approval for this combination therapy, superiority of the combination over the two active compounds and superiority of the two active compounds over placebo need to be demonstrated. A more detailed description of the design can be found in Meyer et al. <DOI:10.1002/pst.2194> and a manual in Meyer et al. <doi:10.48550/arXiv.2202.02182>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

RoxygenNote 7.1.2

Imports dplyr, purrr, ggplot2, plotly, tidyr, parallel, doParallel, foreach, openxlsx, forcats, epitools, zoo

Suggests knitr, rmarkdown, DT, gtools

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-14 09:30:02 UTC

Contents

`make_decision_trial` *Checks whether decision criteria are met and updates trial results accordingly.*

Description

Given a res_list object, checks the supplied decision criteria and saves the results in the res_list file.

Usage

```
make_decision_trial(
  res_list,
  which_cohort,
  test_strat = 3,
  sharing_type = "all",
  Bayes_Sup = NULL,
  Bayes_Fut = NULL,
  Bayes_SA_Sup = NULL,
  Bayes_SA_Fut = NULL,
  w = 0.5,
  P_Sup = NULL,
  P_Fut = NULL,
  Est_Sup_Fut = NULL,
  CI_Sup_Fut = NULL,
  interim,
  beta_prior = 0.5,
  ...
)
```

Arguments

`res_list`	List item containing individual cohort trial results so far in a format used by the other functions in this package
`which_cohort`	Current cohort that should be evaluated
`test_strat`	Testing strategy used; 1 = Combo vs. both monos, 2 = 1 + Add-on Mono vs. Placebo, 3 = 2 + Backbone mono vs. placebo
`sharing_type`	What backbone and placebo data should be used for comparisons; Default is "all". Other options are "concurrent" or "dynamic" or "cohort".

Bayes_Sup	List of matrices with rows corresponding to number of multiple Bayesian posterior two-arm combination criteria for superiority
Bayes_Fut	List of matrices with rows corresponding to number of multiple Bayesian posterior two-arm combination criteria for futility
Bayes_SA_Sup	List of matrices with rows corresponding to number of multiple Bayesian posterior single-arm combination criteria for superiority
Bayes_SA_Fut	List of matrices with rows corresponding to number of multiple Bayesian posterior single-arm combination criteria for futility
w	If dynamic borrowing, what is the prior choice for w. Default is 0.5.
P_Sup	List with sublists corresponding to number of multiple frequentist test-based combination criteria for superiority
P_Fut	List with sublists corresponding to number of multiple frequentist test-based combination criteria for futility
Est_Sup_Fut	List with sublists corresponding to number of multiple point estimate based combination criteria for superiority and futility
CI_Sup_Fut	List with sublists corresponding to number of multiple confidence interval based combination criteria for superiority and futility
interim	Is the analysis conducted an interim or a final analysis?
beta_prior	Prior parameter for all Beta Distributions. Default is 0.5.
...	Further arguments inherited from upper layer functions

Value

List containing original res_list and results of decision rules

Examples

```
# Example 1

res_list <- list(c(list(decision = rep("none", 2), alloc_ratio = c(1,1,1,1),
                  n_thresh = c(Inf, 210)),
           rep(list(list(rr = NULL, resp_bio = NULL, resp_hist = NULL, n = NULL)), 4)))

names(res_list)[1] <- paste0("Cohort", 1)
names(res_list[[1]])[4:7] <- c("Comb", "Mono", "Back", "Plac")
res_list[[1]][[4]]$rr <- 0.2
res_list[[1]][[5]]$rr <- 0.15
res_list[[1]][[6]]$rr <- 0.15
res_list[[1]][[7]]$rr <- 0.10

r141 <- rbinom(1, 70, prob = res_list[[1]][[4]]$rr)
res_list[[1]][[4]]$resp_bio <- gtools::permute(c(rep(1, r141), rep(0, 70 - r141)))
r151 <- rbinom(1, 70, prob = res_list[[1]][[5]]$rr)
res_list[[1]][[5]]$resp_bio <- gtools::permute(c(rep(1, r151), rep(0, 70 - r151)))
r161 <- rbinom(1, 70, prob = res_list[[1]][[6]]$rr)
res_list[[1]][[6]]$resp_bio <- gtools::permute(c(rep(1, r161), rep(0, 70 - r161)))
r171 <- rbinom(1, 70, prob = res_list[[1]][[7]]$rr)
```

```
res_list[[1]][[7]]$resp_bio <- gtools::permute(c(rep(1, r171), rep(0, 70 - r171)))
r142 <- rbinom(1, 70, prob = res_list[[1]][[4]]$rr)
res_list[[1]][[4]]$resp_hist <- gtools::permute(c(rep(1, r142), rep(0, 70 - r142)))
r152 <- rbinom(1, 70, prob = res_list[[1]][[5]]$rr)
res_list[[1]][[5]]$resp_hist <- gtools::permute(c(rep(1, r152), rep(0, 70 - r152)))
r162 <- rbinom(1, 70, prob = res_list[[1]][[6]]$rr)
res_list[[1]][[6]]$resp_hist <- gtools::permute(c(rep(1, r162), rep(0, 70 - r162)))
r172 <- rbinom(1, 70, prob = res_list[[1]][[7]]$rr)
res_list[[1]][[7]]$resp_hist <- gtools::permute(c(rep(1, r172), rep(0, 70 - r172)))

res_list[[1]][[4]]$n <- rep(1, 70)
res_list[[1]][[5]]$n <- rep(1, 70)
res_list[[1]][[6]]$n <- rep(1, 70)
res_list[[1]][[7]]$n <- rep(1, 70)

# Comparison Combo vs Mono
Bayes_Sup1 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup1[1,] <- c(0.00, 0.95, 0.90)
Bayes_Sup1[2,] <- c(0.10, 0.80, 0.75)
Bayes_Sup1[3,] <- c(0.15, 0.50, 1.00)
# Comparison Combo vs Backbone
Bayes_Sup2 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup2[1,] <- c(0.00, 0.95, 0.90)
Bayes_Sup2[2,] <- c(NA, NA, NA)
Bayes_Sup2[3,] <- c(NA, NA, NA)
# Comparison Mono vs Placebo
Bayes_Sup3 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup3[1,] <- c(0.00, 0.95, 0.90)
Bayes_Sup3[2,] <- c(0.10, 0.80, 0.75)
Bayes_Sup3[3,] <- c(NA, NA, NA)
#' # Comparison Backbone vs Placebo
Bayes_Sup4 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup4[1,] <- c(0.00, 0.95, 0.90)
Bayes_Sup4[2,] <- c(0.10, 0.80, 0.75)
Bayes_Sup4[3,] <- c(NA, NA, NA)
Bayes_Sup <- list(list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4),
                  list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4))

sharing_type <- "all"
interim <- TRUE
which_cohort <- 1
missing_prob <- 0.5
seed_missing <- 100

make_decision_trial(
res_list = res_list, which_cohort = which_cohort,
interim = interim, missing_prob = missing_prob,
Bayes_Sup = Bayes_Sup, sharing_type = sharing_type,
seed_missing = seed_missing,
)

# Multiple decision rules
```

```
# Vergleich Combo vs Mono
Bayes_Fut1 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut1[1,] <- c(NA, NA)
# Vergleich Combo vs Backbone
Bayes_Fut2 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut2[1,] <- c(NA, NA)
# Vergleich Mono vs Placebo
Bayes_Fut3 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut3[1,] <- c(0.00, 0.60)
Bayes_Fut4 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut4[1,] <- c(0.00, 0.60)
Bayes_Fut <- list(list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4),
                  list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4))

# Combo
Bayes_SA_Sup1 <- matrix(nrow = 1, ncol = 3)
Bayes_SA_Sup1[1,] <- c(0.20, 0.95, 0.90)
# Mono
Bayes_SA_Sup2 <- matrix(nrow = 1, ncol = 3)
Bayes_SA_Sup2[1,] <- c(0.15, 0.80, 0.75)
# Backbone
Bayes_SA_Sup3 <- matrix(nrow = 1, ncol = 3)
Bayes_SA_Sup3[1,] <- c(0.15, 0.80, 0.75)
# Placebo
Bayes_SA_Sup4 <- matrix(nrow = 1, ncol = 3)
Bayes_SA_Sup4[1,] <- c(0.15, 0.80, 0.75)

Bayes_SA_Sup <- list(list(Bayes_SA_Sup1, Bayes_SA_Sup2, Bayes_SA_Sup3, Bayes_SA_Sup4),
                     list(Bayes_SA_Sup1, Bayes_SA_Sup2, Bayes_SA_Sup3, Bayes_SA_Sup4))

## Combo
Bayes_SA_Fut1 <- matrix(nrow = 1, ncol = 2)
Bayes_SA_Fut1[1,] <- c(0.20, 0.50)
# Mono
Bayes_SA_Fut2 <- matrix(nrow = 1, ncol = 2)
Bayes_SA_Fut2[1,] <- c(0.15, 0.50)
# Backbone
Bayes_SA_Fut3 <- matrix(nrow = 1, ncol = 2)
Bayes_SA_Fut3[1,] <- c(0.15, 0.50)
# Placebo
Bayes_SA_Fut4 <- matrix(nrow = 1, ncol = 2)
Bayes_SA_Fut4[1,] <- c(0.15, 0.50)

Bayes_SA_Fut <- list(list(Bayes_SA_Fut1, Bayes_SA_Fut2, Bayes_SA_Fut3, Bayes_SA_Fut4),
                     list(Bayes_SA_Fut1, Bayes_SA_Fut2, Bayes_SA_Fut3, Bayes_SA_Fut4))

# Comparison Combo vs Mono
P_Sup1 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_sup = 0.025, p_prom = 0.10, p_adj = "B"))
# Comparison Combo vs Backbone
P_Sup2 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
```

```
p_sup = 0.025, p_prom = 0.10, p_adj = "B"))
# Comparison Mono vs Placebo
P_Sup3 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_sup = 0.050, p_prom = 0.10, p_adj = "B"))
P_Sup4 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_sup = 0.050, p_prom = 0.10, p_adj = "B"))
P_Sup <- list(list(P_Sup1, P_Sup2, P_Sup3, P_Sup4),
              list(P_Sup1, P_Sup2, P_Sup3, P_Sup4))

# Comparison Combo vs Mono
P_Fut1 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_fut = 0.5, p_adj = "none"))
# Comparison Combo vs Backbone
P_Fut2 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_fut = 0.5, p_adj = "none"))
# Comparison Mono vs Placebo
P_Fut3 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_fut = 0.5, p_adj = "none"))
# Comparison Backbone Placebo
P_Fut4 <- list(list(
testfun = function(x) stats::prop.test(x, alternative = "less", correct = FALSE),
p_fut = 0.5, p_adj = "none"))
P_Fut <- list(list(P_Fut1, P_Fut2, P_Fut3, P_Fut4),
              list(P_Fut1, P_Fut2, P_Fut3, P_Fut4))

# Comparison Combo vs Mono
Est_Sup_Fut1 <- list(list(est = "AR", p_hat_sup = 0.6, p_hat_fut = 0.1, p_hat_prom = 0.5))
# Comparison Combo vs Backbone
Est_Sup_Fut2 <- list(list(est = "RR", p_hat_sup = 1.25, p_hat_fut = 0.75, p_hat_prom = 1.5))
# Comparison Mono vs Placebo
Est_Sup_Fut3 <- list(list(est = "OR", p_hat_sup = 1.50, p_hat_fut = 0.75, p_hat_prom = 2))
Est_Sup_Fut4 <- list(list(est = "OR", p_hat_sup = 1.50, p_hat_fut = 0.75, p_hat_prom = 2))
Est_Sup_Fut <- list(list(Est_Sup_Fut1, Est_Sup_Fut2, Est_Sup_Fut3, Est_Sup_Fut4),
                    list(Est_Sup_Fut1, Est_Sup_Fut2, Est_Sup_Fut3, Est_Sup_Fut4))

# Comparison Combo vs Mono
CI_Sup_Fut1 <- list(list(est = "AR", ci = 0.95, p_hat_lower_sup = 0.35,
                   p_hat_upper_fut = 0.25, p_hat_lower_prom = 0.3))
# Comparison Combo vs Backbone
CI_Sup_Fut2 <- list(list(est = "RR", ci = 0.95, p_hat_lower_sup = 1.10,
                   p_hat_upper_fut = 1.10, p_hat_lower_prom = 1.05))
# Comparison Mono vs Placebo
CI_Sup_Fut3 <- list(list(est = "OR", ci = 0.95, p_hat_lower_sup = 1.20,
                   p_hat_upper_fut = 1.20, p_hat_lower_prom = 1.10))
CI_Sup_Fut4 <- list(list(est = "OR", ci = 0.95, p_hat_lower_sup = 1.20,
                   p_hat_upper_fut = 1.20, p_hat_lower_prom = 1.10))
CI_Sup_Fut <- list(list(CI_Sup_Fut1, CI_Sup_Fut2, CI_Sup_Fut3, CI_Sup_Fut4),
                   list(CI_Sup_Fut1, CI_Sup_Fut2, CI_Sup_Fut3, CI_Sup_Fut4))
```

```
make_decision_trial(res_list = res_list, which_cohort = which_cohort, interim = interim,
Bayes_Sup = Bayes_Sup, sharing_type = sharing_type,
Bayes_Fut = Bayes_Fut, Bayes_SA_Sup = Bayes_SA_Sup, Bayes_SA_Fut = Bayes_SA_Fut, P_Sup = P_Sup,
P_Fut = P_Fut, Est_Sup_Fut = Est_Sup_Fut, CI_Sup_Fut = CI_Sup_Fut
)
```

`plot_trial` *Plots the cohort trial study overview given stage data.*

Description

Given a res_list object, plots things like final study design, indicating which arms were discontinued after how many patients etc..

Usage

```
plot_trial(res_list, unit = "cohort")
```

Arguments

`res_list`	List item containing trial results so far in a format used by the other functions in this package
`unit`	What is unit of observation in response rate plots: N_cohort or N_total?

Examples

```
random <- TRUE

rr_comb <- c(1)
prob_comb_rr <- c(1)
rr_mono <- c(1,2)
prob_mono_rr <- c(0.2, 0.8)
rr_back <- c(2)
prob_back_rr <- c(1)
rr_plac <- c(0.10)
prob_plac_rr <- c(1)

rr_transform <- list(
  function(x) {return(c(0.90*(1 - x), (1-0.90)*(1-x), (1-0.90)*x, 0.90*x))}
)
prob_rr_transform <- c(1)

cohorts_max <- 20
trial_struc <- "all_plac"
safety_prob <- 0
sharing_type <- "dynamic"
sr_drugs_pos <- 7
```

```
n_int <- 100
n_fin <- 200
stage_data <- TRUE
cohort_random <- 0.02
target_rr <- c(0,0,1)
cohort_offset <- 0
random_type <- "risk_ratio"
sr_first_pos <- FALSE

# Vergleich Combo vs Mono
Bayes_Sup1 <- matrix(nrow = 1, ncol = 3)
Bayes_Sup1[1,] <- c(0.00, 0.90, 1.00)
# Vergleich Combo vs Backbone
Bayes_Sup2 <- matrix(nrow = 1, ncol = 3)
Bayes_Sup2[1,] <- c(0.00, 0.90, 1.00)
# Vergleich Mono vs Placebo
Bayes_Sup3 <- matrix(nrow = 1, ncol = 3)
Bayes_Sup3[1,] <- c(0.00, 0.80, 1.00)
Bayes_Sup4 <- matrix(nrow = 1, ncol = 3)
Bayes_Sup4[1,] <- c(0.00, 0.80, 1.00)
Bayes_Sup <- list(list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4),
             list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4))

# Vergleich Combo vs Mono
Bayes_Fut1 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut1[1,] <- c(0.00, 0.50)
# Vergleich Combo vs Backbone
Bayes_Fut2 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut2[1,] <- c(0.00, 0.50)
# Vergleich Mono vs Placebo
Bayes_Fut3 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut3[1,] <- c(0.00, 0.50)
Bayes_Fut4 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut4[1,] <- c(0.00, 0.50)
Bayes_Fut <- list(list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4),
                  list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4))

res_list <- simulate_trial(
n_int = n_int, n_fin = n_fin, trial_struc = trial_struc, random_type = random_type,
rr_comb = rr_comb, rr_mono = rr_mono, rr_back = rr_back, rr_plac = rr_plac,
rr_transform = rr_transform, random = random, prob_comb_rr = prob_comb_rr,
prob_mono_rr = prob_mono_rr, prob_back_rr = prob_back_rr, prob_plac_rr = prob_plac_rr,
stage_data = stage_data, cohort_random = cohort_random, cohorts_max = cohorts_max,
sr_drugs_pos = sr_drugs_pos, target_rr = target_rr, sharing_type = sharing_type,
safety_prob = safety_prob, Bayes_Sup = Bayes_Sup, prob_rr_transform = prob_rr_transform,
cohort_offset = cohort_offset, Bayes_Fut = Bayes_Fut, sr_first_pos = sr_first_pos
)

plot_trial(res_list, unit = "n")
```

`simulate_trial` *Simulates the cohort trial.*

Description

Simulates the cohort trial.

Usage

```
simulate_trial(
  n_int = 50,
  n_fin = 100,
  cohorts_start = 1,
  rr_comb,
  rr_mono,
  rr_back,
  rr_plac,
  rr_transform,
  random_type = NULL,
  trial_struc = "all_plac",
  random = FALSE,
  prob_comb_rr = NULL,
  prob_mono_rr = NULL,
  prob_back_rr = NULL,
  prob_plac_rr = NULL,
  prob_rr_transform = prob_rr_transform,
  stage_data = TRUE,
  cohort_random = NULL,
  cohorts_max = 4,
  sr_drugs_pos = 1,
  sr_pats = cohorts_max * (n_fin + 3 * cohorts_max),
  sr_first_pos = FALSE,
  target_rr = c(0, 0, 1),
  cohort_offset = 0,
  sharing_type = "all",
  safety_prob = 0,
  cohort_fixed = NULL,
  ...
)
```

Arguments

`n_int`	Sample size per cohort to conduct interim analysis
`n_fin`	Sample size per cohort at final
`cohorts_start`	Number of cohorts to start the platform with
`rr_comb`	Response rates of combination therapies

`rr_mono` Response rate of mono therapies

`rr_back` Response rates of backbone arms

`rr_plac` Response rate of the placebo

`rr_transform` Function transforming all the above response rates to a vector of four probabilities for the multinomial simulation First element is probability of both failures. Second element is probability of biomarker success and histology failure. Third element is probability of biomarker failure and histology success. Fourth element is probability of both success.

`random_type` How should the response rates be drawn randomly? Options are:

"absolute": Specify absolute response rates that will be drawn with a certain probability

"risk_difference": Specify absolute response rates for placebo which will be drawn randomly, plus specify vectors for absolute treatment effects of mono therapies over placebo and for combo over the mono therapies.

"risk_ratio": Specify absolute response rates for placebo which will be drawn randomly, plus specify vectors for relative treatment effects of mono therapies over placebo and for combo over the mono therapies.

"odds_ratios": Specify response rate for placebo, specify odds-ratios for mono therapies (via rr_back and rr_mono) and respective probabilities. On top, specify interaction for the combination therapy via rr_comb with prob_rr_comb. Set: odds_combo = odds_plac * or_mono1 * or_mono2 * rr_comb. If rr_comb > 1 -> synergistic, if rr_comb = 1 -> additive. If rr_comb < 1 -> antagonistic. Default is "NULL".

`trial_struc` Trial Structure: "all_plac" = all cohorts have placebo arm

"no_plac" = no cohort has placebo arm

"stop_post_mono" = all cohorts start with placebo arm, but after first mono has been declared successful, newly enrolled cohorts have no more placebo

"stop_post_back" = all cohorts start with placebo arm, but after first backbone has been declared successful, newly enrolled cohorts have no more placebo

`random` Should the response rates of the arms be randomly drawn from rr_exp? Default is FALSE.

`prob_comb_rr` If random == TRUE, what are the probabilities with which the elements of rr_comb should be drawn?

`prob_mono_rr` If random == TRUE, what are the probabilities with which the elements of rr_mono should be drawn?

`prob_back_rr` If random == TRUE, what are the probabilities with which the elements of rr_back should be drawn?

`prob_plac_rr` If random == TRUE, what are the probabilities with which the elements of rr_plac should be drawn?

`prob_rr_transform` If random == TRUE, what are the probabilities with which the elements of rr_transform should be drawn?

`stage_data` Should individual stage data be passed along? Default is TRUE

cohort_random	If not NULL, indicates that new arms/cohorts should be randomly started. For every patient, there is a cohort_random probability that a new cohort will be started.
cohorts_max	Maximum number of cohorts that are allowed to be added throughout the trial
sr_drugs_pos	Stopping rule for successful experimental arms; Default = 1
sr_pats	Stopping rule for total number of patients; Default = cohorts_max * n_fin + error term based on randomization
sr_first_pos	Stopping rule for first successful cohort; if TRUE, after first cohort was found to be successful, no further cohorts will be included but cohorts will finish evaluating, unless other stopping rules reached prior. Default is FALSE.
target_rr	What is target to declare a combo a positive? Vector of length 3 giving 1) the threshold by which the combo needs to be better than the monos and 2) the threhsold by which the monos need to be better than the placebo. The third element of the vector specifies the relation, choices are 1=="risk-difference", 2=="risk-ratio" and 3=="odds-ratio". By default: c(0,0, "risk-difference").
cohort_offset	Minimum number of patients between adding new cohorts
sharing_type	Which backbone and placebo data should be used for arm comparisons; Default is "all". Another option is "concurrent" or "dynamic" or "cohort".
safety_prob	Probability for a safety stop after every patient
cohort_fixed	If not NULL, fixed timesteps after which a cohort will be included
...	Further arguments to be passed to decision function, such as decision making criteria

Value

List containing: Responses and patients on experimental and control arm, total treatment successes and failures and final p-value

Examples

```
random <- TRUE

rr_comb <- c(0.25, 0.35, 0.4)
prob_comb_rr <- c(0.4, 0.4, 0.2)
rr_mono <- c(0.15, 0.20, 0.25)
prob_mono_rr <- c(0.2, 0.4, 0.4)
rr_back <- c(0.20, 0.25, 0.30)
prob_back_rr <- c(0.3, 0.4, 0.3)
rr_plac <- c(0.10, 0.12, 0.14)
prob_plac_rr <- c(0.25, 0.5, 0.25)

rr_transform <- list(
  function(x) {return(c(0.75*(1 - x), (1-0.75)*(1-x), (1-0.75)*x, 0.75*x))},
  function(x) {return(c(0.85*(1 - x), (1-0.85)*(1-x), (1-0.85)*x, 0.85*x))}
)
prob_rr_transform <- c(0.5, 0.5)
```

```
cohorts_max <- 5
trial_struc <- "stop_post_back"
safety_prob <- 0
sharing_type <- "concurrent"
sr_drugs_pos <- 5
sr_first_pos <- FALSE
n_int <- 50
n_fin <- 100
stage_data <- TRUE
cohort_random <- NULL
target_rr <- c(0,0,1)
cohort_offset <- 0
random_type <- "risk_difference"
cohort_fixed <- 5

# Vergleich Combo vs Mono
Bayes_Sup1 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup1[1,] <- c(0.00, 0.90, 1.00)
Bayes_Sup1[2,] <- c(0.05, 0.65, 1.00)
Bayes_Sup1[3,] <- c(0.10, 0.50, 1.00)
# Vergleich Combo vs Backbone
Bayes_Sup2 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup2[1,] <- c(0.05, 0.80, 1.00)
Bayes_Sup2[2,] <- c(NA, NA, NA)
Bayes_Sup2[3,] <- c(NA, NA, NA)
# Vergleich Mono vs Placebo
Bayes_Sup3 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup3[1,] <- c(0.00, 0.90, 1.00)
Bayes_Sup3[2,] <- c(0.05, 0.65, 1.00)
Bayes_Sup3[3,] <- c(NA, NA, NA)
Bayes_Sup4 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup4[1,] <- c(0.00, 0.90, 1.00)
Bayes_Sup4[2,] <- c(0.05, 0.65, 1.00)
Bayes_Sup4[3,] <- c(NA, NA, NA)
Bayes_Sup <- list(list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4),
             list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4))

# Vergleich Combo vs Mono
Bayes_Fut1 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut1[1,] <- c(0.00, 0.60)
# Vergleich Combo vs Backbone
Bayes_Fut2 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut2[1,] <- c(0.00, 0.60)
# Vergleich Mono vs Placebo
Bayes_Fut3 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut3[1,] <- c(0.00, 0.60)
Bayes_Fut4 <- matrix(nrow = 1, ncol = 2)
Bayes_Fut4[1,] <- c(0.00, 0.60)
Bayes_Fut <- list(list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4),
                  list(Bayes_Fut1, Bayes_Fut2, Bayes_Fut3, Bayes_Fut4))

a <- simulate_trial(
n_int = n_int, n_fin = n_fin, trial_struc = trial_struc, random_type = random_type,
```

```
rr_comb = rr_comb, rr_mono = rr_mono, rr_back = rr_back, rr_plac = rr_plac,
rr_transform = rr_transform, random = random, prob_comb_rr = prob_comb_rr,
prob_mono_rr = prob_mono_rr, prob_back_rr = prob_back_rr, prob_plac_rr = prob_plac_rr,
stage_data = stage_data, cohort_random = cohort_random, cohorts_max = cohorts_max,
sr_drugs_pos = sr_drugs_pos, target_rr = target_rr, sharing_type = sharing_type,
safety_prob = safety_prob, Bayes_Sup = Bayes_Sup, prob_rr_transform = prob_rr_transform,
cohort_offset = cohort_offset, sr_first_pos = sr_first_pos, Bayes_Fut = Bayes_Fut,
cohort_fixed = cohort_fixed
)
```

`trial_ocs` *Calculates the operating characteristics of the cohort trial*

Description

Given the trial specific design parameters, performs a number of simulations of the trial and saves the result in an Excel file

Usage

```
trial_ocs(
  iter,
  coresnum = 1,
  save = FALSE,
  path = NULL,
  filename = NULL,
  ret_list = FALSE,
  ret_trials = FALSE,
  plot_ocs = FALSE,
  export = NULL,
  ...
)
```

Arguments

`iter`	Number of program simulations that should be performed
`coresnum`	How many cores should be used for parallel computing
`save`	Indicator whether simulation results should be saved in an Excel file
`path`	Path to which simulation results will be saved; if NULL, then save to current path
`filename`	Filename of saved Excel file with results; if NULL, then name will contain design parameters
`ret_list`	Indicator whether function should return list of results
`ret_trials`	Indicator whether individual trial results should be saved as well
`plot_ocs`	Should OCs stability plots be drawn?
`export`	Should any other variables be exported to the parallel tasks?
`...`	All other design parameters for chosen program

Value

List containing: Responses and patients on experimental and control arm, total treatment successes and failures and final p-value

Examples

```
random <- TRUE

rr_comb <- c(0.40, 0.45, 0.50)
prob_comb_rr <- c(0.4, 0.4, 0.2)
rr_mono <- c(0.20, 0.25, 0.30)
prob_mono_rr <- c(0.2, 0.4, 0.4)
rr_back <- c(0.20, 0.25, 0.30)
prob_back_rr <- c(0.2, 0.4, 0.4)
rr_plac <- c(0.10, 0.12, 0.14)
prob_plac_rr <- c(0.25, 0.5, 0.25)

rr_transform <- list(
  function(x) {return(c(0.75*(1 - x), (1-0.75)*(1-x), (1-0.75)*x, 0.75*x))},
  function(x) {return(c(0.85*(1 - x), (1-0.85)*(1-x), (1-0.85)*x, 0.85*x))}
)
prob_rr_transform <- c(0.5, 0.5)

cohorts_max <- 4
safety_prob <- 0
sharing_type <- "all"
trial_struc <- "all_plac"
sr_drugs_pos <- 4
n_int <- 100
n_fin <- 200
stage_data <- TRUE
cohort_random <- 0.05
target_rr <- c(0,0,1)
cohort_offset <- 0
random_type <- "absolute"
sr_first_pos <- FALSE
missing_prob <- 0.1

# Vergleich Combo vs Mono
Bayes_Sup1 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup1[1,] <- c(0.05, 0.90, 1.00)
Bayes_Sup1[2,] <- c(0.05, 0.65, 1.00)
Bayes_Sup1[3,] <- c(0.10, 0.50, 1.00)
# Vergleich Combo vs Backbone
Bayes_Sup2 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup2[1,] <- c(0.05, 0.90, 1.00)
Bayes_Sup2[2,] <- c(NA, NA, NA)
Bayes_Sup2[3,] <- c(NA, NA, NA)
# Vergleich Mono vs Placebo
Bayes_Sup3 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup3[1,] <- c(0.00, 0.90, 1.00)
Bayes_Sup3[2,] <- c(NA, NA, NA)
```

```
Bayes_Sup3[3,] <- c(NA, NA, NA)
# Vergleich Back vs Placebo
Bayes_Sup4 <- matrix(nrow = 3, ncol = 3)
Bayes_Sup4[1,] <- c(0.00, 0.90, 1.00)
Bayes_Sup4[2,] <- c(NA, NA, NA)
Bayes_Sup4[3,] <- c(NA, NA, NA)
Bayes_Sup <- list(list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4),
             list(Bayes_Sup1, Bayes_Sup2, Bayes_Sup3, Bayes_Sup4))

ocs <- trial_ocs(
n_int = n_int, n_fin = n_fin, random_type = random_type,
rr_comb = rr_comb, rr_mono = rr_mono, rr_back = rr_back, rr_plac = rr_plac,
rr_transform = rr_transform, random = random, prob_comb_rr = prob_comb_rr,
prob_mono_rr = prob_mono_rr, prob_back_rr = prob_back_rr, prob_plac_rr = prob_plac_rr,
stage_data = stage_data, cohort_random = cohort_random, cohorts_max = cohorts_max,
sr_drugs_pos = sr_drugs_pos, target_rr = target_rr, sharing_type = sharing_type,
sr_first_pos = sr_first_pos, safety_prob = safety_prob, Bayes_Sup = Bayes_Sup,
prob_rr_transform = prob_rr_transform, cohort_offset = cohort_offset,
trial_struc = trial_struc, missing_prob = missing_prob,
iter = 10, coresnum = 1, save = FALSE, ret_list = TRUE, plot_ocs = TRUE
)

ocs[[3]]
```

Index