Package 'crm12Comb'

September 16, 2024

Type Package

Title Phase I/II CRM Based Drug Combination Design

Version 0.1.7

Maintainer Junying Wang <junying.wang@stonybrook.edu>

Description Implements the adaptive designs for integrated phase I/II trials of drug combinations via continual reassessment method (CRM) to evaluate toxicity and efficacy simultaneously for each enrolled patient cohort based on Bayesian inference. It supports patients assignment guidance in a single trial using current enrolled data, as well as conducting extensive simulation studies to evaluate operating characteristics before the trial starts. It includes various link functions such as empiric, one-parameter logistic, two-parameter logistic, and hyperbolic tangent, as well as considering multiple prior distributions of the parameters like normal distribution, gamma distribution and exponential distribution to accommodate diverse clinical scenarios. Method using Bayesian framework with empiric link function is described in: Wages and Conaway (2014) <doi:10.1002/sim.6097>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

VignetteBuilder knitr

Imports dplyr, ggplot2, ggforce

Suggests knitr, rmarkdown

Depends R (>= 3.5.0)

NeedsCompilation no

Author Junying Wang [cre, aut],
Song Wu [aut],
Jie Yang [aut]

Repository CRAN

Date/Publication 2024-09-16 14:30:01 UTC

Contents

`doseComb_to_mat` *Title: Get 6 complete orderings for toxicity and efficacy for drug combinations*

Description

Step 1: convert dose combinations into matrix

Usage

```
doseComb_to_mat(doseComb, type)
```

Arguments

`doseComb`	Either a numeric matrix or a numeric vector or a numeric list.
`type`	A string. condition 1: detailed dose combinations are directly given in matrix -> type = "comb" dose combinations dose A dose B condition 2: directly given the number of levels for the two doses in vector -> type = "matrix" Step 2: get the 6 orderings

Value

outMat -> Either a list or a matrix. Note: each array refers to the index of drug combinations orderings)

References

Wages NA, Conaway MR. Specifications of a continual reassessment method design for phase I trials of combined drugs. Pharmaceutical statistics. 2013 Jul;12(4):217-24. doi:10.1002/pst.1575

`efficacy_est` *Efficacy estimation*

Description

Estimate efficacy using Bayesian inference for each enrolled patient or cohort of patients given the current accumulated data and toxicity estimation.

Usage

```
efficacy_est(Dat, AR, I, K, K_prob, efficacy_skeleton, Nphase,
             model, para_prior,
         theta_mean, theta_sd, theta_intcpt_lgst1, theta_shape, theta_inverse_scale,
             alphaT_mean, alphaT_sd, alphaT_shape, alphaT_inverse_scale,
             seed=NULL, seed_rand=NULL, seed_max=NULL)
```

Arguments

`Dat`	A data frame for current data with three columns (DoseLevel, DLT and ORR).
`AR`	A vector for acceptable set.
`I`	Number of dose combinations.
`K`	Number of efficacy orderings.
`K_prob`	A vector of length K denoting prior probabilities of efficacy orderings (sum is 1).
`efficacy_skeleton`	A list of vector with length K containing efficacy orderings.
`Nphase`	Number of patients for determination of randomization phase (current number of patients less than Nphase) or maximization phase (current number of patients larger than or equal to Nphase).
`model`	A character string to specify the model used, must be one of "empiric", "tanh", "logistic" or "logistic2".
`para_prior`	A character string to specify the prior distribution used for parameters, must be one of "normal" or "gamma" (when model is either "empiric" or "logistic" or "logistic2") or "exponential" (when model is "tanh").

`theta_mean` The mean of parameter used when prior="exponential" or "normal", otherwise need to specify NULL.

`theta_sd` The standard deviation of parameter used when prior="normal", otherwise need to specify NULL.

`theta_intcpt_lgst1`
A constant value of intercept from a one-parameter logistic model only used when model="logistic" (suggested value is 3), otherwise need to specify NULL.

`theta_shape` The shape parameter used when prior="gamma", otherwise need to specify NULL.

`theta_inverse_scale`
The scale parameter used when prior="gamma", otherwise need to specify NULL.

`alphaT_mean` The mean of intercept parameter of two-parameter logistic model only used when model="logistic2" and prior="normal", otherwise need to specify NULL.

`alphaT_sd` The standard deviation of intercept parameter of two-parameter logistic model only used when model="logistic2" and prior="normal", otherwise need to specify NULL.

`alphaT_shape` The shape parameter of intercept parameter from a two-parameter logistic model only used when model="logistic2" and prior="gamma", otherwise need to specify NULL.

`alphaT_inverse_scale`
The scale parameter of intercept parameter from a two-parameter logistic model only used when model="logistic2" and prior="gamma", otherwise need to specify NULL.

`seed` An integer for the seed to generate random numbers used for equal ordering prior probabilities, default is NULL.

`seed_rand` An integer for the seed to generate random numbers used in randomization phase, default is NULL.

`seed_max` An integer for the seed to generate random numbers used in maximization phase, default is NULL.

Details

The efficacy estimation is based on Bayesian framework by calculating the likelihood function under each orderings, and select the ordering with maximum posterior probability. Then, estimated parameters can be obtained, which will be used for efficacy estimation based on the corresponding link function (specified in $model$ statement).

Value

A list is returned containing the following components:

`di` Number of dose level for next enrolled patient or cohort of patients

`K_prob` A vector for posterier density of efficacy orderings and will be used as prior information for next enrolled patient or cohort of patients.

References

Wages, N. A., & Conaway, M. R. (2014). Phase I/II adaptive design for drug combination oncology trials. Statistics in medicine, 33(12), 1990-2003. doi:10.1002/sim.6097

See Also

priorSkeletons, get_ordering, toxicity_est, randomization_phase, maximization_phase

Examples

```
### follow same steps as the example in function toxicity_est()
# Generate a data including 3 columns: DoseLevel, DLT, ORR (DLT and ORR are binary outcomes)
currDat <- data.frame(sample(1:9, 6, replace=TRUE), rbinom(6, 1, 0.2), rbinom(6, 1, 0.5))
names(currDat) <- c("DoseLevel", "DLT", "ORR")

# Generate toxicity and efficacy skeleton
DLT_skeleton_p <- priorSkeletons(updelta = 0.045, target = 0.3, npos= 5, ndose = 9,
                            model = "logistic", prior = "normal", beta_mean = 0, a0 = 3)
eff_skeleton_p <- priorSkeletons(updelta = 0.045, target = 0.5, npos= 5, ndose = 9,
                            model = "logistic", prior = "normal", beta_mean = 0, a0 = 3)

# Obtain 6 complete orderings for toxicity skeleton and efficacy skeleton
orderings <- get_ordering(doseComb_forMat=c(3,3), type_forMat="matrix")
DLT_skeleton_l <- lapply(orderings, function(or){DLT_skeleton_p[or]})
eff_skeleton_l <- lapply(orderings, function(or){eff_skeleton_p[or]})

# estimate toxicity
tox <- toxicity_est(Dat=currDat, I=9, M=6, M_prob=rep(1/6, 6),
                    DLT_skeleton=DLT_skeleton_l, DLT_thresh=0.33,
                    model="logistic", para_prior="normal",
                    beta_mean=0, beta_sd=1, intcpt_lgst1=3,
                    beta_shape=NULL, beta_inverse_scale=NULL,
                    alpha_mean=NULL, alpha_sd=NULL,
                    alpha_shape=NULL, alpha_inverse_scale=NULL,
                    seed=42)

### efficacy estimation
eff <- efficacy_est(Dat=currDat, AR=tox$AR, I=9, K=6, K_prob=rep(1/6, 6),
                    efficacy_skeleton=eff_skeleton_l, Nphas=20,
                    model="logistic", para_prior="normal",
                    theta_mean=0, theta_sd=1, theta_intcpt_lgst1=3,
                    theta_shape=NULL, theta_inverse_scale=NULL,
                    alphaT_mean=NULL, alphaT_sd=NULL,
                    alphaT_shape=NULL, alphaT_inverse_scale=NULL,
                    seed=1, seed_rand=2, seed_max=3)
```

`empiric_GammaPriorLikelihood`

empiric model with gamma prior

Description

empiric model with gamma prior

Usage

```
empiric_GammaPriorLikelihood(beta, beta_shape, beta_inverse_scale, x, y)
```

Arguments

beta	parameter.
beta_shape	A number.
beta_inverse_scale	A number.
x	A numeric vector.
y	A numeric vector.

Value

l -> likelihood function

`empiric_NormalPriorLikelihood` *empiric model with normal prior*

Description

empiric model with normal prior

Usage

```
empiric_NormalPriorLikelihood(beta, beta_mean, beta_sd, x, y)
```

Arguments

beta	parameter.
beta_mean	A number.
beta_sd	A number.
x	A numeric vector.
y	A numeric vector.

Value

l -> likelihood function

enroll_patient_plot *Plot patient enrollment for single trial*

Description

This function is used to generate the plot of patient enrollment with toxicity (red) and efficacy (green) outcomes of a single trial.

Usage

```
enroll_patient_plot(data)
```

Arguments

data	A data frame with 3 columns DoseLevel, DLT for toxicity outcome, and ORR for efficacy outcome.

Value

Returns a ggplot object.

Examples

```
# input the scenario with pre-defined true toxicity and efficacy probabilities
scenario <- matrix(c(0.02, 0.05,
                     0.04, 0.10,
                     0.08, 0.15,
                     0.12, 0.32,
                     0.06, 0.10,
                     0.10, 0.15,
                     0.14, 0.25,
                     0.20, 0.35,
                     0.12, 0.18,
                     0.16, 0.22,
                     0.22, 0.35,
                     0.25, 0.40,
                     0.20, 0.24,
                     0.24, 0.35,
                     0.35, 0.45,
                     0.40, 0.50), ncol=2, byrow = TRUE)

# generate skeletons
DLT_skeleton <- priorSkeletons(updelta=0.025, target=0.3, npos=10, ndose=16,
                               model = "empiric", prior = "normal", beta_mean=0)
Efficacy_skeleton <- priorSkeletons(updelta=0.025, target=0.5, npos=10, ndose=16,
                                    model = "empiric", prior = "normal", beta_mean=0)

# simulate 1 trial under the same model and prior distribution
simRes <- SIM_phase_I_II(nsim=1, Nmax=40, DoseComb=scenario, input_doseComb_forMat=c(4,4),
                         input_type_forMat="matrix", input_Nphase=20,
```

```
                          input_DLT_skeleton=DLT_skeleton,
                          input_efficacy_skeleton=Efficacy_skeleton,
                          input_DLT_thresh=0.3, input_efficacy_thresh=0.3,
                          input_cohortsize=1, input_corr=0,
                          input_early_stopping_safety_thresh=0.33,
                          input_early_stopping_futility_thresh=0.2,
                          input_model="empiric", input_para_prior="normal",
                          input_beta_mean=0, input_beta_sd=1,
                          input_theta_mean=0, input_theta_sd=1)

enroll_patient_plot(simRes$datALL[[1]])
```

`examples_results` *Output dataset for examples given list of inputs*

Description

The dataset contains 1296 rows (6 scenarios, 4 different combinations of link functions and prior distributions, 2 sets of skeletons, 3 maximum number of patients, 3 toxicity and efficacy correlations, and 3 subset number of patients) that represent each condition by 1000 simulations, along with 15 columns that contain 9 operating characteristics, other 6 columns including Scenario, Model for link function and prior distribution, N for maximum number of patients, Skeleton, Nphase for subset number of patients, and corr for toxicity and efficacy correlation to separate each condition.

Usage

```
examples_results
```

`get_ordering` *Complete orderings for combinations of two drugs*

Description

This function is to obtain complete orderings of both toxicity and efficacy for all drug combinations when considering the partial ordering issue for two combined drugs.

Usage

```
get_ordering(doseComb_forMat, type_forMat)
```

Arguments

`doseComb_forMat`
: For 2 drugs, either a matrix with columns for all combinations and rows for two drugs when type_forMat is "comb", or a vector of length 2 indicating the number of levels of each drug when type_forMat is "matrix" should be input. For more than 2 drugs, a list of all possible orderings should be input when type_forMat is "self".

`type_forMat`
: A character string to indicate the input type of dose combinations. The type_forMat is either "comb" for inputting all tested dose combinations for 2 drugs, "matrix" for entering number of levels of two drugs so that a matrix of combinations can be constructed, or "self" for more than 3 drugs with directly input of all possible orderings.

Details

Dose-toxicity and dose-efficacy curves for single drug are assumed as monotonically increasing as drug increases. After combined two drugs and traversing all combinations, the number of possible orderings exponentially expands when increasing dose levels for single drug or the dimensions of drug combinations, which will mask the information provided from orderings. Here, 6 typical complete orderings are utilized from practical design.

Take the 3×3 matrix as an example so that there are total of 9 does combination.

$$\begin{array}{cc|ccc} & & & B & \\ & & 1 & 2 & 3 \\ & 1 & d_1 & d_2 & d_3 \\ A & 2 & d_4 & d_5 & d_6 \\ & 3 & d_7 & d_8 & d_9 \end{array}$$

Across rows:

$$\pi_T(d_1) \leq \pi_T(d_2) \leq \pi_T(d_3) \leq \pi_T(d_4) \leq \pi_T(d_5) \leq \pi_T(d_6) \leq \pi_T(d_7) \leq \pi_T(d_8) \leq \pi_T(d_9)$$

Up columns:

$$\pi_T(d_1) \leq \pi_T(d_4) \leq \pi_T(d_7) \leq \pi_T(d_2) \leq \pi_T(d_5) \leq \pi_T(d_8) \leq \pi_T(d_3) \leq \pi_T(d_6) \leq \pi_T(d_9)$$

Up diagonals:

$$\pi_T(d_1) \leq \pi_T(d_2) \leq \pi_T(d_4) \leq \pi_T(d_3) \leq \pi_T(d_5) \leq \pi_T(d_7) \leq \pi_T(d_6) \leq \pi_T(d_8) \leq \pi_T(d_9)$$

Down diagonals:

$$\pi_T(d_1) \leq \pi_T(d_4) \leq \pi_T(d_2) \leq \pi_T(d_7) \leq \pi_T(d_5) \leq \pi_T(d_3) \leq \pi_T(d_8) \leq \pi_T(d_6) \leq \pi_T(d_9)$$

Alternating down-up diagonals:

$$\pi_T(d_1) \leq \pi_T(d_2) \leq \pi_T(d_4) \leq \pi_T(d_7) \leq \pi_T(d_5) \leq \pi_T(d_3) \leq \pi_T(d_6) \leq \pi_T(d_8) \leq \pi_T(d_9)$$

Alternating up-down diagonals:

$$\pi_T(d_1) \leq \pi_T(d_4) \leq \pi_T(d_2) \leq \pi_T(d_3) \leq \pi_T(d_5) \leq \pi_T(d_7) \leq \pi_T(d_8) \leq \pi_T(d_6) \leq \pi_T(d_9)$$

Finally, obtain the unique orderings if there are any duplicates.

Value

A list of vectors are returned, with each vector representing a specific order of dose combinations in ascending order (dose combinations are denoted by natrual numbers).

References

Wages, N. A., & Conaway, M. R. (2013). Specifications of a continual reassessment method design for phase I trials of combined drugs. Pharmaceutical statistics, 12(4), 217-224. doi:10.1002/pst.1575

Examples

```
# Input all dose combinations
doseComb <- matrix(c(10, 3, 20, 3, 30, 3, 40, 3, 20, 5, 20, 7), byrow=FALSE, nrow=2)
orderings <- get_ordering(doseComb_forMat = doseComb, type_forMat = "comb")

# Input number of dose levels
doseLevels <- c(2,3)
orderings <- get_ordering(doseComb_forMat = doseLevels, type_forMat = "matrix")

# Input customized orderings
orders <- list(c(1,2,3,4,5), c(2,1,4,3,5), c(1,2,4,3,5), c(2,1,3,4,5))
orderings <- get_ordering(doseComb_forMat = orders, type_forMat = "self")

# Extreme case: only one ordering will be input
orders <- list(c(1,2,3,4,5,6,7))
orderings <- get_ordering(doseComb_forMat = orders, type_forMat = "self")
```

---

logisticOnePara_GammaPriorLikelihood

one-parameter logistic model with gamma prior

Description

one-parameter logistic model with gamma prior

Usage

```
logisticOnePara_GammaPriorLikelihood(
  alpha1,
  alpha1_shape,
  alpha1_inverse_scale,
  intcpt,
  x,
  y
)
```

Arguments

alpha1	parameter.
alpha1_shape	A number.
alpha1_inverse_scale	A number.
intcpt	A number.
x	A numeric vector.
y	A numeric vector.

Value

l -> likelihood function

---

logisticOnePara_NormalPriorLikelihood

one-parameter logistic model with normal prior

Description

one-parameter logistic model with normal prior

Usage

```
logisticOnePara_NormalPriorLikelihood(
  alpha1,
  alpha1_mean,
  alpha1_sd,
  intcpt,
  x,
  y
)
```

Arguments

alpha1	parameter.
alpha1_mean	A number.
alpha1_sd	A number.
intcpt	A number.
x	A numeric vector.
y	A numeric vector.

Value

l -> likelihood function

`logisticTwoPara_GammaPriorLikelihood` *two-parameter logistic model with gamma prior*

Description

two-parameter logistic model with gamma prior

Usage

```
logisticTwoPara_GammaPriorLikelihood(
  alpha0,
  alpha1,
  alpha0_shape,
  alpha0_inverse_scale,
  alpha1_shape,
  alpha1_inverse_scale,
  x,
  y
)
```

Arguments

`alpha0`	parameter.
`alpha1`	parameter.
`alpha0_shape`	A number.
`alpha0_inverse_scale`	A number.
`alpha1_shape`	A number.
`alpha1_inverse_scale`	A number.
`x`	A numeric vector.
`y`	A numeric vector.

Value

l -> likelihood function

`logisticTwoPara_NormalPriorLikelihood` *two-parameter logistic model with normal prior*

Description

two-parameter logistic model with normal prior

Usage

```
logisticTwoPara_NormalPriorLikelihood(
  alpha0,
  alpha1,
  alpha0_mean,
  alpha0_sd,
  alpha1_mean,
  alpha1_sd,
  x,
  y
)
```

Arguments

`alpha0`	parameter.
`alpha1`	parameter.
`alpha0_mean`	A number.
`alpha0_sd`	A number.
`alpha1_mean`	A number.
`alpha1_sd`	A number.
`x`	A numeric vector.
`y`	A numeric vector.

Value

l -> likelihood function

maximization_phase *Maximization phase*

Description

This function is used to perform maximization to select the dose level with maximum efficacy probability for next patient or cohort of patients allocation when the current sample size is greater than or equal to a pre-specified number.

Usage

```
maximization_phase(pE_est, seed_m=NULL)
```

Arguments

pE_est	A vector of estimated efficacy probability in the acceptable set.
seed_m	An integer for the seed to generate random numbers used in maximization phase, default is NULL.

Details

If several dose combinations have the same maximum estimated efficacy probability, then randomly select one dose level for next enrolled patient or cohort of patients.

Value

A number is returned indicating the dose level for next patient or cohort of patients allocation.

Examples

```
# Assume the estimated prbabilities for each dose combination in the acceptable set as:
p_est <- c(0.1, 0.2, 0.3, 0.4)
# Dose level for next enrolled patient or cohort of patients is:
d <- maximization_phase(p_est)
```

ODC_plot *Plot optimal combination dose selections*

Description

This function is used to generate the plot of optimal combination dose (ODC) selections among a number simulation trials.

Usage

```
ODC_plot(SimsRes)
```

Arguments

SimsRes	A S4 class stores simulation results obtained from function SIM_phase_I_II.

Value

Returns a ggplot object.

Examples

```
# input the scenario with pre-defined true toxicity and efficacy probabilities
scenario <- matrix(c(0.02, 0.05,
                     0.04, 0.10,
                     0.08, 0.15,
                     0.12, 0.32,
                     0.06, 0.10,
                     0.10, 0.15,
                     0.14, 0.25,
                     0.20, 0.35,
                     0.12, 0.18,
                     0.16, 0.22,
                     0.22, 0.35,
                     0.25, 0.40,
                     0.20, 0.24,
                     0.24, 0.35,
                     0.35, 0.45,
                     0.40, 0.50), ncol=2, byrow = TRUE)

# generate skeletons
DLT_skeleton <- priorSkeletons(updelta=0.025, target=0.3, npos=10, ndose=16,
                               model = "empiric", prior = "normal", beta_mean=0)
Efficacy_skeleton <- priorSkeletons(updelta=0.025, target=0.5, npos=10, ndose=16,
                                    model = "empiric", prior = "normal", beta_mean=0)

# simulate 1 trial under the same model and prior distribution
simRes <- SIM_phase_I_II(nsim=1, Nmax=40, DoseComb=scenario, input_doseComb_forMat=c(4,4),
                         input_type_forMat="matrix", input_Nphase=20,
                         input_DLT_skeleton=DLT_skeleton,
                         input_efficacy_skeleton=Efficacy_skeleton,
                         input_DLT_thresh=0.3, input_efficacy_thresh=0.3,
                         input_cohortsize=1, input_corr=0,
                         input_early_stopping_safety_thresh=0.33,
                         input_early_stopping_futility_thresh=0.2,
                         input_model="empiric", input_para_prior="normal",
                         input_beta_mean=0, input_beta_sd=1,
                         input_theta_mean=0, input_theta_sd=1)

ODC_plot(simRes)
```

patient_allocation_plot *Plot patient allocation for a single trial*

Description

This function is used to generate the plot of patient allocation by dose combinations of a single trial.

Usage

```
patient_allocation_plot(data)
```

Arguments

data	A data frame with 3 columns DoseLevel, DLT for toxicity outcome, and ORR for efficacy outcome.

Value

Returns a ggplot object.

Examples

```
# input the scenario with pre-defined true toxicity and efficacy probabilities
scenario <- matrix(c(0.02, 0.05,
                     0.04, 0.10,
                     0.08, 0.15,
                     0.12, 0.32,
                     0.06, 0.10,
                     0.10, 0.15,
                     0.14, 0.25,
                     0.20, 0.35,
                     0.12, 0.18,
                     0.16, 0.22,
                     0.22, 0.35,
                     0.25, 0.40,
                     0.20, 0.24,
                     0.24, 0.35,
                     0.35, 0.45,
                     0.40, 0.50), ncol=2, byrow = TRUE)

# generate skeletons
DLT_skeleton <- priorSkeletons(updelta=0.025, target=0.3, npos=10, ndose=16,
                               model = "empiric", prior = "normal", beta_mean=0)
Efficacy_skeleton <- priorSkeletons(updelta=0.025, target=0.5, npos=10, ndose=16,
                                    model = "empiric", prior = "normal", beta_mean=0)

# simulate 1 trial under the same model and prior distribution
simRes <- SIM_phase_I_II(nsim=1, Nmax=40, DoseComb=scenario, input_doseComb_forMat=c(4,4),
                         input_type_forMat="matrix", input_Nphase=20,
```

```
                          input_DLT_skeleton=DLT_skeleton,
                          input_efficacy_skeleton=Efficacy_skeleton,
                          input_DLT_thresh=0.3, input_efficacy_thresh=0.3,
                          input_cohortsize=1, input_corr=0,
                          input_early_stopping_safety_thresh=0.33,
                          input_early_stopping_futility_thresh=0.2,
                          input_model="empiric", input_para_prior="normal",
                          input_beta_mean=0, input_beta_sd=1,
                          input_theta_mean=0, input_theta_sd=1)

patient_allocation_plot(simRes$datALL[[1]])
```

`priorSkeletons` *Generate the skeletons of toxicity and efficacy*

Description

This function is used to generate skeletons of toxicity and efficacy. This is a modifed version based on `getprior`, which keep the same procedure using empiric and one-parameter logistic models assumed normal priors with $mean = 0$ and further add multiple models with various prior distributions including hyperbolic tangent model with exponential prior, empiric/one-parameter logistic models with normal prior and self-input mean values as well as with gamma prior, and two-parameter logistic model with normal/gamma priors.

Usage

```
priorSkeletons(updelta, target, npos, ndose,
               model = "empiric", prior = "normal",
               alpha_mean=NULL, beta_mean=0, a0 = 3,
               alpha_shape=NULL, alpha_inverse_scale=NULL,
               beta_shape=NULL, beta_inverse_scale=NULL)
```

Arguments

`updelta`	The half-width of the indifference intervals.
`target`	The target DLT rate.
`npos`	The prior guess of the position of MTD.
`ndose`	The number of testing doses.
`model`	A character string to specify the model used. The default model is "empiric". Other models include hyperbolic tangent model specified by "tanh", one-parameter logistic model specified by "logistic", and two-parameter logistic model specified by "logistic2".
`prior`	A character sting to specify the prior distribution of parameter. The default prior is "normal" used together with the model="empiric". Other prior distributions include "exponential" when model="tanh", "gamma" when model="empiric", "normal" and "gamma" when model="logistic" and "logistic2".

`alpha_mean` The mean of intercept parameter of two-parameter logistic model only used when model="logistic2" and prior="normal", otherwise will be ignored.

`beta_mean` The mean of parameter used when prior="exponential" or "normal", otherwise will be ignored.

`a0` A constant value of intercept from a one-parameter logistic model only used when model="logistic" with default value 3, otherwise will be ignored.

`alpha_shape` The shape parameter of intercept parameter only used when model="logistic2" and prior="gamma", otherwise will be ignored.

`alpha_inverse_scale` The scale parameter of intercept parameter only used when model="logistic2" and prior="gamma", otherwise will be ignored.

`beta_shape` The shape parameter used when prior="gamma", otherwise will be ignored.

`beta_inverse_scale` The scale parameter used when prior="gamma", otherwise will be ignored.

Value

A vector of length $ndose$ is returned.

Note

The skeletons can be either specified by clinical researchers based on history information or directly generated based on this function given specific model and prior distribution.

References

Lee, S. M., & Cheung, Y. K. (2009). Model calibration in the continual reassessment method. Clinical Trials, 6(3), 227-238. doi:10.1177/1740774509105076

Examples

```
# generate skeleton based on empiric model with normal prior
prior <- priorSkeletons(updelta = 0.01, target = 0.25, npos= 5, ndose = 9, beta_mean = 0)

# generate skeleton based on one-parameter logistic model with normal prior
prior <- priorSkeletons(updelta = 0.01, target = 0.25, npos= 5, ndose = 9,
                        model = "logistic", prior = "normal", beta_mean = 0, a0 = 3)
```

---

randomization_phase *Adaptive randomization*

Description

This function is used to perform adaptive randomization for next patient or cohort of patients allocation when the current sample size is less than a pre-specified number.

Usage

```
randomization_phase(pE_est, seed_r=NULL)
```

Arguments

pE_est	A vector of estimated efficacy probability in the acceptable set.
seed_r	An integer for the seed to generate random numbers used in randomization phase, default is NULL.

Details

The dose combination for next patient or cohort of patients allocation is d_i with probability

$$R_i = \frac{\hat{\pi}_E(d_i)}{\sum_i \hat{\pi}_E(d_i)}.$$

Value

A number is returned indicating the dose level for next patient or cohort of patients allocation.

Examples

```
# Assume the estimated prbabilities for each dose combination in the acceptable set as:
p_est <- c(0.1, 0.2, 0.3, 0.4)
# Dose level for next enrolled patient or cohort of patients is:
d <- randomization_phase(p_est)
```

---

rBin2Corr *Generate correlated binary variables*

Description

Generate correlated bivariate binary outcomes of toxicity and efficacy for a cohort number of patients.

Usage

```
rBin2Corr(cohortsize, pT, pE, psi, seed=NULL)
```

Arguments

cohortsize	Number of patients in each cohort.
pT	Toxicity probability.
pE	Efficacy probability.
psi	Association parameter for efficacy and toxicity, where psi=0 means toxicity and efficacy is independent.
seed	An integer for the seed to generate random numbers, default is NULL.

Details

The formula for generating correlated binary variables is

$$\pi_{i,j} = (\pi_E)^i(1-\pi_E)^{1-i}(\pi_T)^j(1-\pi_T)^{1-j} + (-1)^{i+j}\pi_E(1-\pi_E)\pi_T(1-\pi_T)\left(\frac{e^\psi - 1}{e^\psi + 1}\right),$$

where $i, j = 0, 1$, so that four probabilities can be calculated for the possible combinations of (toxicity, efficacy) including $(1,1), (0,0), (0,1), (1,0)$ given π_T and π_E. Multinomial distribution `rmultinom` is further used to generate bivariate binary outcomes (number equals to cohortsize) based on the four calculated probabilities.

Value

Return a $cohortsize \times 2$ matrix with columns corresponding to toxicity and efficacy, and rows for each observations of binary outcome with 0 for no toxicity (no efficacy) and 1 for toxicity (efficacy) at the first (second) column.

References

Murtaugh, P. A., & Fisher, L. D. (1990). Bivariate binary models of efficacy and toxicity in dose-ranging trials. Communications in Statistics-Theory and Methods, 19(6), 2003-2020. doi:10.1080/03610929008830305

Thall, P. F., & Cook, J. D. (2004). Dose-finding based on efficacy–toxicity trade-offs. Biometrics, 60(3), 684-693. doi:10.1111/j.0006341X.2004.00218.x

Examples

```
rBin2Corr(cohortsize = 1, pT = 0.2, pE = 0.4, psi = 0, seed=12)
```

`sample_plot` *Sample plot for a given output results*

Description

This function is used to generate the plot of relationships between outcomes and total average sample size, number of patients for determination of randomization phase, skeleton, and Association parameter for efficacy and toxicity binary outcome by different scenarios and link functions.

Usage

```
sample_plot(dat, outcome, outname, N = NULL, nR = NULL, Skeleton = NULL, corr = NULL)
```

Arguments

dat	Input data used for plot.
outcome	Column name for the outcome used in the plot.
outname	A string for the name of outcome.
N	Maximum sample size, if not fixed, use the default value NULL.
nR	Number of patients for determination of randomization phase, if not fixed, use the default value NULL.
Skeleton	Two skeletons with number 1 and 2, if not fixed, use the default value NULL.
corr	Association parameter for efficacy and toxicity, if not fixed, use the default value NULL.

Details

4 settings with multiple inputs: need to fix three and plot each outcome vs. the remaining one by 6 scenarios:

1. N: maximum sample size -> 40, 50, 60
2. nR: subset sample size -> 10, 20, 30
3. Skeleton: two sets of skeletons for toxicity and efficacy
4. corr: correlation between toxicity and efficacy binary outcomes -> 0, -2.049, 0.814

Value

Returns a ggplot object.

Examples

```
# load the data stored in the crm12Comb package
data(examples_results, package = "crm12Comb")

# fix the number of patients for determination of randomization phase, skeleton,
# and Association parameter for efficacy and toxicity binary outcome
# plot the relationship between
# "Probability of ODC as target combinations" vs. total average sample size
sample_plot(examples_results, outcome = "prob_target",
  outname = "Probability of ODC as target combinations",
  N = NULL, nR = 20, Skeleton = 1, corr = 0)
```

SIM_phase_I_II *Single simulation of phase I/II adaptive design for drug combinations based on CRM design*

Description

This function is to realize the simulations of whole process of combined phase I and phase II design for drug combinations with early stopping rules and flexible user-defined model or prior distribution, returning operating characters for performace evaluations and comparisons.

Usage

```
SIM_phase_I_II(nsim, Nmax, DoseComb, input_doseComb_forMat,
               input_type_forMat, input_Nphase,
               input_DLT_skeleton, input_efficacy_skeleton,
               input_DLT_thresh=0.3, input_efficacy_thresh=0.3,
               input_M_prob=NULL, input_K_prob=NULL,
               input_cohortsize=1, input_corr=0,
               input_early_stopping_safety_thresh=0.33,
               input_early_stopping_futility_thresh=0.2,
               input_model="empiric", input_para_prior="normal",
               input_beta_mean=0, input_beta_sd=1,
               input_intcpt_lgst1=3,
               input_beta_shape=1, input_beta_inverse_scale=1,
               input_theta_mean=0, input_theta_sd=1,
               input_theta_intcpt_lgst1=3,
               input_theta_shape=1, input_theta_inverse_scale=1,
               input_alpha_mean=3, input_alpha_sd=1,
               input_alpha_shape=1, input_alpha_inverse_scale=1,
               input_alphaT_mean=3, input_alphaT_sd=1,
               input_alphaT_shape=1, input_alphaT_inverse_scale=1,
               random_seed=42)
```

Arguments

nsim	Number of simulation trials.
Nmax	Number of maximum enrolled patients for each trial.
DoseComb	A matrix with rows of length equaling number of dose combinations and columns of length equaling number of drugs for pre-defined true toxicity and true efficacy probabilities corresponding to each dose combinations.
input_doseComb_forMat	For 2 drugs, either a matrix with columns for all combinations and rows for two drugs when input_type_forMat is "comb", or a vector of length 2 indicating the number of levels of each drug when input_type_forMat is "matrix" should be input. For more than 2 drugs, a list of all possible orderings should be input when input_type_forMat is "self".
input_type_forMat	A character string to indicate the input type of dose combinations. The type_forMat is either "comb" for inputting all tested dose combinations for 2 drugs, "matrix" for entering number of levels of two drugs so that a matrix of combinations can be constructed, or "self" for more than 3 drugs with directly input of all possible orderings.
input_Nphase	Number of patients for determination of randomization phase (current number of patients less than Nphase) or maximization phase (current number of patients larger than or equal to Nphase).
input_DLT_skeleton	A vector with same length of dose combinations for toxicity skeleton for each dose combination.

`input_efficacy_skeleton`
: A vector with same length of dose combinations for efficacy skeleton for each dose combination.

`input_DLT_thresh`
: DLT threshold to define acceptable set.

`input_efficacy_thresh`
: Efficacy threshold to define target combinations.

`input_M_prob`
: A vector with same length of possible toxicity orderings denoting prior probabilities of toxicity orderings (sum is 1).

`input_K_prob`
: A vector with same length of possible efficacy orderings denoting prior probabilities of efficacy orderings (sum is 1).

`input_cohortsize`
: Number of patients in each cohort.

`input_corr`
: Association parameter for efficacy and toxicity, where 0 means toxicity and efficacy is independent.

`input_early_stopping_safety_thresh`
: Safety threshold for early stopping condition.

`input_early_stopping_futility_thresh`
: Futility threshold for early stopping condition.

`input_model`
: A character string to specify the model used, must be one of "empiric", "tanh", "logistic" or "logistic2".

`input_para_prior`
: A character string to specify the prior distribution used for parameters, must be one of "normal" or "gamma" (when input_model is either "empiric" or "logistic" or "logistic2") or "exponential" (when input_model is "tanh").

`input_beta_mean`
: The mean of parameter used when input_para_prior="exponential" or "normal", otherwise will be ignored.

`input_beta_sd`
: The standard deviation of parameter used when input_para_prior="normal", otherwise will be ignored.

`input_intcpt_lgst1`
: A constant value of intercept from a one-parameter logistic model only used when input_model="logistic" (suggested value is 3), otherwise will be ignored.

`input_beta_shape`
: The shape parameter used when input_para_prior="gamma", otherwise will be ignored.

`input_beta_inverse_scale`
: The scale parameter used when input_para_prior="gamma", otherwise will be ignoredL.

`input_theta_mean`
: The mean of parameter used when input_para_prior="exponential" or "normal", otherwise will be ignored.

`input_theta_sd`
: The standard deviation of parameter used when input_para_prior="normal", otherwise will be ignored.

`input_theta_intcpt_lgst1`
: A constant value of intercept from a one-parameter logistic model only used when input_model="logistic" (default value is 3), otherwise will be ignored.

`input_theta_shape`
: The shape parameter used when input_para_prior="gamma", otherwise will be ignored.

`input_theta_inverse_scale`
: The scale parameter used when input_para_prior="gamma", otherwise will be ignored.

`input_alpha_mean`
: The mean of intercept parameter of two-parameter logistic model only used when model="logistic2" and input_para_prior="normal", otherwise will be ignored.

`input_alpha_sd`
: The standard deviation of intercept parameter of two-parameter logistic model only used when input_model="logistic2" and input_para_prior="normal", otherwise will be ignored.

`input_alpha_shape`
: The shape parameter of intercept parameter from a two-parameter logistic model only used when input_model="logistic2" and input_para_prior="gamma", otherwise will be ignored.

`input_alpha_inverse_scale`
: The scale parameter of intercept parameter from a two-parameter logistic model only used when input_model="logistic2" and input_para_prior="gamma", otherwise will be ignored.

`input_alphaT_mean`
: The mean of intercept parameter of two-parameter logistic model only used when model="logistic2" and input_para_prior="normal", otherwise will be ignored.

`input_alphaT_sd`
: The standard deviation of intercept parameter of two-parameter logistic model only used when input_model="logistic2" and input_para_prior="normal", otherwise will be ignored.

`input_alphaT_shape`
: The shape parameter of intercept parameter from a two-parameter logistic model only used when input_model="logistic2" and input_para_prior="gamma", otherwise will be ignored.

`input_alphaT_inverse_scale`
: The scale parameter of intercept parameter from a two-parameter logistic model only used when input_model="logistic2" and input_para_prior="gamma", otherwise will be ignored.

`random_seed`
: An integer for the start seed to generate random numbers, default is 42.

Details

This function is to realize the whole process of combined phase I and phase II adaptive design for drug combinations based on CRM amonng number of simulation trials. For each trial, basic steps include starting the trial for first patient or cohor of patients allocation, toxicity and efficacy

estimations by current data, adaptive randomization or maximization phase for next patient or cohort of patients allocation, updating current data of patients enrollment, determining early stoppings for safety or fuility, and selecting optimal dose combination (ODC) after reaching maximum sample size or stopping early.

Value

A list of operating characteristics is returned containing the following components:

`prob_safe`	Probability of simulation trials with ODC identified as safe/ineffective combinations.
`prob_target`	Probability of simulation trials with ODC identified as target combinations.
`prob_toxic`	Probability of simulation trials with ODC identified as toxic combinations.
`mean_SS`	Average number of patients enrolled.
`mean_ODC`	Average proportion of patients allocated to target ODS(s).
`prob_stop_safety`	Probability of simulation trials stopping early for safety.
`prob_stop_futility`	Probability of simulation trials stopping early for futility.
`mean_DLT`	Average observed DLT rate.
`mean_ORR`	Average observed response rate.
`Npatient`	A vector of number of patients enrolled for all simulation trials.
`ODC`	A vector of ODC for all simulation trials.
`prop_ODC`	A list of vectors storing dose levels for each patients allocation.
`datALL`	A list of data frames storing data of (doselevel, toxicity, efficacy) for each patient.

References

Wages, N. A., & Conaway, M. R. (2014). Phase I/II adaptive design for drug combination oncology trials. Statistics in medicine, 33(12), 1990-2003. doi:10.1002/sim.6097

See Also

`priorSkeletons`, `get_ordering`, `toxicity_est`, `efficacy_est`, `rBin2Corr`, `binom.test`

Examples

```
# input the scenario with pre-defined true toxicity and efficacy probabilities
scenario <- matrix(c(0.02, 0.05,
                     0.04, 0.10,
                     0.08, 0.15,
                     0.12, 0.32,
                     0.06, 0.10,
                     0.10, 0.15,
                     0.14, 0.25,
                     0.20, 0.35,
```

```
                      0.12, 0.18,
                      0.16, 0.22,
                      0.22, 0.35,
                      0.25, 0.40,
                      0.20, 0.24,
                      0.24, 0.35,
                      0.35, 0.45,
                      0.40, 0.50), ncol=2, byrow = TRUE)

# generate skeletons
DLT_skeleton <- priorSkeletons(updelta=0.025, target=0.3, npos=10, ndose=16,
                               model = "empiric", prior = "normal", beta_mean=0)
Efficacy_skeleton <- priorSkeletons(updelta=0.025, target=0.5, npos=10, ndose=16,
                                    model = "empiric", prior = "normal", beta_mean=0)

# simulate 1 trial under the same model and prior distribution
simRes <- SIM_phase_I_II(nsim=1, Nmax=40, DoseComb=scenario, input_doseComb_forMat=c(4,4),
                         input_type_forMat="matrix", input_Nphase=20,
                         input_DLT_skeleton=DLT_skeleton,
                         input_efficacy_skeleton=Efficacy_skeleton,
                         input_DLT_thresh=0.3, input_efficacy_thresh=0.3,
                         input_cohortsize=1, input_corr=0,
                         input_early_stopping_safety_thresh=0.33,
                         input_early_stopping_futility_thresh=0.2,
                         input_model="empiric", input_para_prior="normal",
                         input_beta_mean=0, input_beta_sd=1,
                         input_theta_mean=0, input_theta_sd=1)
```

`tanh_ExpPriorLikelihood`

Title: Bayesian likelihood inference

Description

Title: Bayesian likelihood inference

Usage

```
tanh_ExpPriorLikelihood(beta, beta_mean, x, y)
```

Arguments

`beta`	parameter.
`beta_mean`	A number.
`x`	A numeric vector.
`y`	A numeric vector.

Value

l -> likelihood function

References

: R package dfcrm Hyperbolic tangent model with exponential prior

---

`toxicity_est` *Toxicity estimation*

Description

Estimate toxicity using Bayesian inference for each enrolled patient or cohort of patients given the current accumulated data.

Usage

```
toxicity_est(Dat, I, M, M_prob, DLT_skeleton, DLT_thresh,
             model, para_prior,
             beta_mean, beta_sd, intcpt_lgst1, beta_shape, beta_inverse_scale,
             alpha_mean, alpha_sd, alpha_shape, alpha_inverse_scale, seed=NULL)
```

Arguments

`Dat`	A data frame for current data with three columns (DoseLevel, DLT and ORR).
`I`	Number of dose combinations.
`M`	Number of toxicity orderings.
`M_prob`	A vector of length M denoting prior probabilities of toxicity orderings (sum is 1).
`DLT_skeleton`	A list of vector with length M containing toxicity orderings.
`DLT_thresh`	DLT threshold to define acceptable set.
`model`	A character string to specify the model used, must be one of "empiric", "tanh", "logistic" or "logistic2".
`para_prior`	A character string to specify the prior distribution used for parameters, must be one of "normal" or "gamma" (when model is either "empiric" or "logistic" or "logistic2") or "exponential" (when model is "tanh").
`beta_mean`	The mean of parameter used when prior="exponential" or "normal", otherwise need to specify NULL.
`beta_sd`	The standard deviation of parameter used when prior="normal", otherwise need to specify NULL.
`intcpt_lgst1`	A constant value of intercept from a one-parameter logistic model only used when model="logistic" (suggested value is 3), otherwise need to specify NULL.
`beta_shape`	The shape parameter used when prior="gamma", otherwise need to specify NULL.

`beta_inverse_scale`
: The scale parameter used when prior="gamma", otherwise need to specify NULL.

`alpha_mean`
: The mean of intercept parameter of two-parameter logistic model only used when model="logistic2" and prior="normal", otherwise need to specify NULL.

`alpha_sd`
: The standard deviation of intercept parameter of two-parameter logistic model only used when model="logistic2" and prior="normal", otherwise need to specify NULL.

`alpha_shape`
: The shape parameter of intercept parameter from a two-parameter logistic model only used when model="logistic2" and prior="gamma", otherwise need to specify NULL.

`alpha_inverse_scale`
: The scale parameter of intercept parameter from a two-parameter logistic model only used when model="logistic2" and prior="gamma", otherwise need to specify NULL.

`seed`
: An integer for the seed to generate random numbers used for equal ordering prior probabilities, default is NULL.

Details

The toxicity estimation is based on Bayesian framework by calculating the likelihood function under each orderings, and select the ordering with maximum posterior probability. Then, estimated parameters can be obtained, which will be used for toxicity estimation based on the corresponding link function (specified in *model* statement).

Value

A list is returned containing the following components:

`AR`
: A vector for dose levels belonging to acceptable set (estimated DLT rate less than pre-specified DLT threshold).

`M_prob`
: A vector for posterier density of toxicity orderings and will be used as prior information for next enrolled patient or cohort of patients.

References

Wages, N. A., & Conaway, M. R. (2014). Phase I/II adaptive design for drug combination oncology trials. Statistics in medicine, 33(12), 1990-2003. doi:10.1002/sim.6097

See Also

`priorSkeletons`, `get_ordering`

Examples

```
# Generate a data including 3 columns: DoseLevel, DLT, ORR (DLT and ORR are binary outcomes)
currDat <- data.frame(sample(1:9, 6, replace=TRUE), rbinom(6, 1, 0.2), rbinom(6, 1, 0.5))
names(currDat) <- c("DoseLevel", "DLT", "ORR")

# Generate toxicity skeleton
```

```
DLT_skeleton_p <- priorSkeletons(updelta = 0.045, target = 0.3, npos= 5, ndose = 9,
                            model = "logistic", prior = "normal", beta_mean = 0, a0 = 3)

# Obtain 6 complete orderings for toxicity skeleton
orderings <- get_ordering(doseComb_forMat=c(3,3), type_forMat="matrix")
DLT_skeleton_l <- lapply(orderings, function(or){DLT_skeleton_p[or]})

# estimate toxicity
tox <- toxicity_est(Dat=currDat, I=9, M=6, M_prob=rep(1/6, 6),
                    DLT_skeleton=DLT_skeleton_l, DLT_thresh=0.3,
                    model="logistic", para_prior="normal",
                    beta_mean=0, beta_sd=1, intcpt_lgst1=3,
                    beta_shape=NULL, beta_inverse_scale=NULL,
                    alpha_mean=NULL, alpha_sd=NULL,
                    alpha_shape=NULL, alpha_inverse_scale=NULL,
                    seed=42)
```

Index