Package 'boinet'

June 5, 2025

Version 1.2.0

Title Conduct Simulation Study of Bayesian Optimal Interval Design
with BOIN-ET Family

Description Bayesian optimal interval based on both efficacy and toxicity outcomes (BOIN-ET) design is a model-assisted oncology phase I/II trial design, aiming to establish an optimal biological dose accounting for efficacy and toxicity in the framework of dose-finding. Some extensions of BOIN-ET design are also available to allow for time-to-event efficacy and toxicity outcomes based on cumulative and pending data (time-to-event BOIN-ET: TITE-BOIN-ET), ordinal graded efficacy and toxicity outcomes (generalized BOIN-ET: gBOIN-ET), and their combination (TITE-gBOIN-ET). 'boinet' is a package to implement the BOIN-ET design family and supports the conduct of simulation studies to assess operating characteristics of BOIN-ET, TITE-BOIN-ET, gBOIN-ET, and TITE-gBOIN-ET, where users can choose design parameters in flexible and straightforward ways depending on their own application.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports Iso, mfp, copula, gt, tibble

Suggests testthat (>= 3.0.0), knitr, rmarkdown, dplyr (>= 1.0.0),
ggplot2,

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Yusuke Yamaguchi [aut, cre],
Kentaro Takeda [aut]

Maintainer Yusuke Yamaguchi <yamagubed@gmail.com>

Repository CRAN

Date/Publication 2025-06-05 21:10:02 UTC

Contents

---

boinet	*BOIN-ET: Bayesian Optimal Interval Design for Dose-Finding Based on Efficacy and Toxicity*

Description

Conducts simulation studies of the BOIN-ET (Bayesian Optimal Interval design for dose finding based on both Efficacy and Toxicity outcomes) design to evaluate its operating characteristics for identifying the optimal biological dose (OBD). The BOIN-ET design extends the Bayesian optimal interval (BOIN) design, which is nonparametric and thus does not require the assumption used in model-based designs, in order to identify an optimal dose based on both efficacy and toxicity outcomes.

Unlike traditional phase I designs that focus solely on toxicity to find the maximum tolerated dose (MTD), BOIN-ET addresses the modern need to balance safety and efficacy for targeted therapies, immunotherapies, and biologics where the efficacy of the drug does not always increase and could plateau at a lower dose.

Usage

```
boinet(
  n.dose, start.dose, size.cohort, n.cohort,
  toxprob, effprob,
  phi = 0.3, phi1 = phi*0.1, phi2 = phi*1.4,
  delta = 0.6, delta1 = delta*0.6,
  alpha.T1 = 0.5, alpha.E1 = 0.5, tau.T, tau.E,
  te.corr = 0.2, gen.event.time = "weibull",
  accrual, gen.enroll.time = "uniform",
  stopping.npts = size.cohort*n.cohort,
  stopping.prob.T = 0.95, stopping.prob.E = 0.99,
  estpt.method = "obs.prob", obd.method = "max.effprob",
  w1 = 0.33, w2 = 1.09,
  plow.ast = phi1, pupp.ast = phi2,
  qlow.ast = delta1/2, qupp.ast = delta,
  psi00 = 40, psi11 = 60,
  n.sim = 1000, seed.sim = 100)
```

Arguments

n.dose	Integer specifying the number of dose levels to investigate.
start.dose	Integer specifying the starting dose level (1 = lowest dose). Generally recommended to start at the lowest dose for safety.
size.cohort	Integer specifying the number of patients per cohort. Commonly 3 or 6 patients, with 3 being standard for early-phase trials.
n.cohort	Integer specifying the maximum number of cohorts. Total sample size = size.cohort*n.cohort.
toxprob	Numeric vector of length n.dose specifying the true toxicity probabilities for each dose level. Used for simulation scenarios.
effprob	Numeric vector of length n.dose specifying the true efficacy probabilities for each dose level. Used for simulation scenarios.
phi	Numeric value between 0 and 1 specifying the target toxicity probability. Represents the maximum acceptable toxicity rate. Default is 0.3 (30%).
phi1	Numeric value specifying the highest toxicity probability that is deemed sub-therapeutic such that dose-escalation should be pursued. Doses with toxicity <= phi1 are considered under-dosed. Default is phi*0.1.
phi2	Numeric value specifying the lowest toxicity probability that is deemed overly toxic such that dose de-escalation is needed. Doses with toxicity >= phi2 are considered over-dosed. Default is phi*1.4.
delta	Numeric value between 0 and 1 specifying the target efficacy probability. Represents the desired minimum efficacy rate. Default is 0.6 (60%).
delta1	Numeric value specifying the minimum probability deemed efficacious such that the dose levels with efficacy < delta1 are considered sub-therapeutic. Default is delta*0.6.
alpha.T1	Numeric value specifying the probability that a toxicity outcome occurs in the late half of the toxicity assessment window. Used for event time generation. Default is 0.5.

alpha.E1	Numeric value specifying the probability that an efficacy outcome occurs in the late half of the efficacy assessment window. Used for event time generation. Default is 0.5.
tau.T	Numeric value specifying the toxicity assessment window in days. All toxicity evaluations must be completed within this period.
tau.E	Numeric value specifying the efficacy assessment window in days. All efficacy evaluations must be completed within this period.
te.corr	Numeric value between -1 and 1 specifying the correlation between toxicity and efficacy, specified as Gaussian copula parameter. Default is 0.2 (weak positive correlation).
gen.event.time	Character string specifying the distribution for generating event times. Options are "weibull" (default) or "uniform". A bivariate Gaussian copula model is used to jointly generate the time to first toxicity and efficacy outcome, where the marginal distributions are set to Weibull distribution when gen.event.time="weibull", and uniform distribution when gen.event.time="uniform".
accrual	Numeric value specifying the accrual rate (days), which is the average number of days between patient enrollments. Lower values indicate faster accrual.
gen.enroll.time	Character string specifying the distribution for enrollment times. Options are "uniform" (default) or "exponential". Uniform distribution is used when gen.enroll.time="uniform", and exponential distribution is used when gen.enroll.time="exponential".
stopping.npts	Integer specifying the maximum number of patients per dose for early study termination. If the number of patients at the current dose reaches this criteria, the study stops the enrollment and is terminated. Default is size.cohort*n.cohort.
stopping.prob.T	Numeric value between 0 and 1 specifying the early study termination threshold for toxicity. If P(toxicity > phi) > stopping.prob.T, the dose levels are eliminated from the investigation. Default is 0.95.
stopping.prob.E	Numeric value between 0 and 1 specifying the early study termination threshold for efficacy. If P(efficacy < delta1) > stopping.prob.E, the dose levels are eliminated from the investigation. Default is 0.99.
estpt.method	Character string specifying the method for estimating efficacy probabilities. Options: "obs.prob" (observed efficacy probabilitiesrates), "fp.logistic" (fractional polynomial), or "multi.iso" (model averaging of multiple unimodal isotopic regression). Default is "obs.prob".
obd.method	Character string specifying the method for OBD selection. Options: "utility.weighted", "utility.truncated.linear", "utility.scoring", or "max.effprob" (default).
w1	Numeric value specifying the weight for toxicity-efficacy trade-off in "utility.weighted" method. Default is 0.33.
w2	Numeric value specifying the penalty weight for toxic doses in "utility.weighted" method. Default is 1.09.
plow.ast	Numeric value specifying the lower toxicity threshold for "utility.truncated.linear" method. Default is phi1.

pupp.ast	Numeric value specifying the upper toxicity threshold for "utility.truncated.linear" method. Default is phi2.
qlow.ast	Numeric value specifying the lower efficacy threshold for "utility.truncated.linear" method. Default is delta1/2.
qupp.ast	Numeric value specifying the upper efficacy threshold for "utility.truncated.linear" method. Default is delta.
psi00	Numeric value specifying the utility score for (toxicity=no, efficacy=no) in "utility.scoring" method. Default is 40.
psi11	Numeric value specifying the utility score for (toxicity=yes, efficacy=yes) in "utility.scoring" method. Default is 60.
n.sim	Integer specifying the number of simulated trials. Default is 1000. Higher values provide more stable operating characteristics.
seed.sim	Integer specifying the random seed for reproducible results. Default is 100.

Details

Design Philosophy and Context:

One of the main purposes of a phase I dose-finding trial in oncology is to identify an optimal dose (OD) that is both tolerable and has an indication of therapeutic benefit for subjects in subsequent phase II and III trials. Traditional dose-finding methods assume monotonic dose-toxicity and dose-efficacy relationships, but this assumption often fails for modern cancer therapies.

The BOIN-ET design is a **model-assisted** approach, not model-based, which makes it:

- **Robust**: No parametric assumptions about dose-response curves
- **Simple**: Pre-tabulated decision rules make implementation transparent
- **Flexible**: Accommodates various toxicity and efficacy scenarios
- **Clinically interpretable**: Decisions based on intuitive probability intervals

Key Design Features:

Dose Escalation/De-escalation Rules: The design uses pre-calculated boundaries (lambda1, lambda2, eta1) to make dosing decisions:

- **Escalate**: When toxicity <= lambda1 AND efficacy <= eta1
- **Stay**: When lambda1 < toxicity < lambda2 AND efficacy > eta1
- **De-escalate**: When toxicity >= lambda2
- **Efficacy-guided**: When lambda1 < toxicity < lambda2 AND efficacy <= eta1

Assessment Windows: Unlike time-to-event designs, standard BOIN-ET waits for complete outcome assessment within specified windows (tau.T for toxicity, tau.E for efficacy) before making dose decisions. This ensures data completeness but may slow accrual.

Early Stopping Rules: The design includes safety and futility stopping criteria based on posterior probabilities exceeding pre-specified thresholds (stopping.prob.T, stopping.prob.E).

OBD Selection Methods: Multiple utility-based approaches are available for final dose selection:

- **utility.weighted**: Balances efficacy and toxicity with user-defined weights

- **utility.truncated.linear**: Uses piecewise linear utility functions
- **utility.scoring**: Discrete scoring system for outcome combinations
- **max.effprob**: Maximizes efficacy among acceptably safe doses

When to Use BOIN-ET vs TITE-BOIN-ET:

Use BOIN-ET when:

- Outcomes occur relatively quickly (within assessment windows)
- Patient accrual is slow enough to wait for complete assessments
- You prefer simpler implementation without time-to-event modeling
- Historical precedent suggests minimal late-onset effects

Use TITE-BOIN-ET when:

- Late-onset toxicity or efficacy is expected
- Rapid patient accrual is anticipated
- Trial duration is a critical constraint

Simulation Output Interpretation:

The simulation results provide crucial operating characteristics:

- **prop.select**: Probability of correctly identifying each dose as OBD
- **n.patient**: Expected patient allocation across doses
- **prop.stop**: Probability of early termination without OBD selection
- **duration**: Expected trial duration

Value

A list object of class "boinet" containing the following components:

toxprob	True toxicity probabilities used in simulation.
effprob	True efficacy probabilities used in simulation.
phi	Target toxicity probability.
delta	Target efficacy probability.
lambda1	Lower toxicity decision boundary.
lambda2	Upper toxicity decision boundary.
eta1	Lower efficacy decision boundary.
tau.T	Toxicity assessment window (days).
tau.E	Efficacy assessment window (days).
accrual	Accrual rate (days).
estpt.method	Method used for efficacy probability estimation.
obd.method	Method used for optimal biological dose selection.
n.patient	Average number of patients treated at each dose level across simulations.
prop.select	Percentage of simulations selecting each dose level as OBD.
prop.stop	Percentage of simulations terminating early without OBD selection.
duration	Expected trial duration in days.

Note

- Design parameters must satisfy: phi1 < phi < phi2 and delta1 < delta
- Toxicity and efficacy probability vectors must have length n.dose
- The design waits for complete outcome assessment before dose decisions
- Consider TITE-BOIN-ET for trials with late-onset outcomes or rapid accrual
- Boundary values (lambda1, lambda2, eta1) are automatically optimized via grid search

References

- Takeda, K., Taguri, M., & Morita, S. (2018). BOIN-ET: Bayesian optimal interval design for dose finding based on both efficacy and toxicity outcomes. *Pharmaceutical Statistics*, 17(4), 383-395.
- Yamaguchi, Y., Takeda, K., Yoshida, S., & Maruo, K. (2024). Optimal biological dose selection in dose-finding trials with model-assisted designs based on efficacy and toxicity: a simulation study. *Journal of Biopharmaceutical Statistics*, 34(3), 379-393.

See Also

tite.boinet for the time-to-event version that handles late-onset outcomes, obd.select for optimal biological dose selection methods, utility.weighted, utility.truncated.linear, utility.scoring for utility functions, gridoptim for boundary optimization.

Examples

```
# Example 1: Basic BOIN-ET simulation for targeted therapy
# Scenario: Non-monotonic efficacy with moderate toxicity

n.dose      <- 5
start.dose  <- 1
size.cohort <- 3
n.cohort    <- 15  # Total: 45 patients

# Dose levels: 25mg, 50mg, 100mg, 200mg, 400mg
toxprob <- c(0.05, 0.10, 0.25, 0.40, 0.60)  # Monotonic toxicity
effprob <- c(0.20, 0.45, 0.70, 0.65, 0.55)  # Non-monotonic efficacy (plateau effect)

# Conservative targets for targeted therapy
phi   <- 0.25  # 25% maximum toxicity
delta <- 0.60  # 60% target efficacy

# Assessment windows
tau.T  <- 28  # 4 weeks for toxicity
tau.E  <- 84  # 12 weeks for efficacy
accrual <- 14  # 2 weeks between patients

# Use observed probabilities and maximum efficacy method
estpt.method <- "obs.prob"
obd.method   <- "max.effprob"
```

```r
# Run simulation (small n.sim for example)
results <- boinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  phi = phi, delta = delta,
  tau.T = tau.T, tau.E = tau.E, accrual = accrual,
  estpt.method = estpt.method, obd.method = obd.method,
  n.sim = 100
)

# Display key results
print(results$prop.select)  # OBD selection probabilities
print(results$n.patient)    # Patient allocation
print(results$duration)     # Expected trial duration

# Example 2: Immunotherapy with utility-weighted OBD selection
# Higher tolerance for toxicity if efficacy is present

n.dose      <- 4
size.cohort <- 6  # Larger cohorts for immunotherapy
n.cohort    <- 10

# Immunotherapy dose-response pattern
toxprob <- c(0.10, 0.20, 0.35, 0.50)
effprob <- c(0.15, 0.30, 0.50, 0.45)  # Slight plateau at highest dose

phi   <- 0.35  # Higher toxicity tolerance
delta <- 0.40  # Lower efficacy requirement

tau.T   <- 42  # 6 weeks for immune-related toxicity
tau.E   <- 112 # 16 weeks for immune response
accrual <- 7   # Weekly accrual

# Use utility-weighted method to balance toxicity-efficacy
results_utility <- boinet(
  n.dose = n.dose, start.dose = 1,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  phi = phi, delta = delta,
  tau.T = tau.T, tau.E = tau.E, accrual = accrual,
  estpt.method = "fp.logistic",  # Flexible dose-response modeling
  obd.method = "utility.weighted",
  w1 = 0.4,  # Moderate toxicity penalty
  w2 = 0.8,  # Additional penalty for high toxicity
  n.sim = 100
)

# Display key results
print(results_utility$prop.select)  # OBD selection probabilities
print(results_utility$n.patient)    # Patient allocation
print(results_utility$duration)     # Expected trial duration
```

create_boinet_tables   *Create Report-Ready Tables from BOIN-ET Results*

Description

Generates publication-ready gt tables from boinet simulation results. This is a convenience wrapper for format_boinet_results with gt_ready output.

Usage

```
create_boinet_tables(
  boinet_result,
  oc_title = "Operating Characteristics",
  design_title = "Design Parameters"
)
```

Arguments

boinet_result	Result object from any boinet function
oc_title	Title for operating characteristics table
design_title	Title for design parameters table

Value

List containing formatted gt tables and tidy data

Examples

```
## Not run:
result <- tite.boinet(...)
tables <- create_boinet_tables(result,
                               oc_title = "Trial Operating Characteristics",
                               design_title = "Design Specifications")

# Print tables
print(tables$oc_table)
print(tables$design_table)

# Save tables
tables$oc_table |> gt::gtsave("oc_results.html")

## End(Not run)
```

---

create_design_gt_table

Create GT Table for Design Parameters

Description

Creates a formatted gt table displaying design parameters.

Usage

```
create_design_gt_table(boinet_result, title = "Design Parameters")
```

Arguments

boinet_result   Result object from boinet functions

title   Optional table title

Value

A gt table object

---

create_oc_gt_table   *Create GT Table for Operating Characteristics*

Description

Creates a formatted gt table displaying operating characteristics from boinet simulation results.

Usage

```
create_oc_gt_table(boinet_result, title = "Operating Characteristics")
```

Arguments

boinet_result   Result object from boinet functions

title   Optional table title

Value

A gt table object

Examples

```
## Not run:
result <- tite.boinet(...)
table <- create_oc_gt_table(result, "My Trial Results")
print(table)

## End(Not run)
```

---

extract_boinet_data   *Extract Tidy Data from BOIN-ET Results*

Description

Convenience function to extract tidy data frames from boinet results. This is a wrapper for format_boinet_results with tidy output.

Usage

```
extract_boinet_data(boinet_result)
```

Arguments

boinet_result   Result object from any boinet function

Value

List containing tidy data frames

Examples

```
## Not run:
result <- gboinet(...)
tidy_data <- extract_boinet_data(result)

# Use tidy data for custom analysis
library(ggplot2)
tidy_data$operating_characteristics |>
  ggplot(aes(x = dose_level, y = selection_prob)) +
  geom_col()

## End(Not run)
```

---

extract_design_summary

Extract Design Parameters Summary

Description

Extracts design parameters from boinet results in a tidy format.

Usage

```
extract_design_summary(boinet_result)
```

Arguments

boinet_result   Result object from boinet functions (boinet, tite.boinet, gboinet, tite.gboinet)

Value

A tibble with parameter names and values

---

extract_operating_characteristics

Extract Operating Characteristics in Tidy Format

Description

Converts boinet simulation results into a tidy data frame suitable for analysis and reporting.

Usage

```
extract_operating_characteristics(boinet_result)
```

Arguments

boinet_result   Result object from boinet functions (tite.boinet, tite.gboinet, boinet, gboinet)

Value

A tibble with columns: dose_level, toxicity_prob, efficacy_prob, n_patients, selection_prob, selection_pct

Examples

```
## Not run:
# Run BOIN-ET simulation
result <- tite.boinet(
  n.dose = 4, start.dose = 1, size.cohort = 3, n.cohort = 10,
  toxprob = c(0.05, 0.15, 0.25, 0.40),
  effprob = c(0.15, 0.30, 0.45, 0.60),
  phi = 0.30, delta = 0.60, n.sim = 100
)

# Extract in tidy format
oc_data <- extract_operating_characteristics(result)
print(oc_data)

## End(Not run)
```

---

format_boinet_results *Format BOIN-ET Results for Reporting*

Description

Converts boinet simulation results into various formats optimized for analysis and reporting workflows. This is the MAIN new function.

Usage

```
format_boinet_results(
  boinet_result,
  output_format = c("list", "tidy", "gt_ready")
)
```

Arguments

boinet_result   Result object from any boinet function (boinet, tite.boinet, gboinet, tite.gboinet)

output_format   Character string specifying output format:

- "list" - Returns original result unchanged
- "tidy" - Returns tidy data frames for analysis
- "gt_ready" - Returns formatted gt tables for reporting

Value

Results in specified format

Examples

```
## Not run:
# Run simulation first
result <- tite.boinet(
  n.dose = 4, start.dose = 1, size.cohort = 3, n.cohort = 10,
  toxprob = c(0.05, 0.15, 0.25, 0.40),
  effprob = c(0.15, 0.30, 0.45, 0.60),
  phi = 0.30, delta = 0.60, n.sim = 100
)

# Format for different uses
original <- format_boinet_results(result, "list")
tidy_data <- format_boinet_results(result, "tidy")
report_tables <- format_boinet_results(result, "gt_ready")

# Use formatted results
print(tidy_data$operating_characteristics)
print(report_tables$oc_table)

## End(Not run)
```

---

fp.logit   *Fractional Polynomial Logistic Regression for Dose-Efficacy Model-*
                                  ing

Description

Performs fractional polynomial (FP) logistic regression with two degrees of freedom to estimate the
efficacy probabilities. This method provides a flexible alternative to standard polynomial regression
for capturing non-linear dose-efficacy relationships. Fractional polynomials have steadily gained
popularity as a tool for flexible parametric modeling of regression relationships and are particularly
useful when the relationship between dose and response may not follow simple linear or quadratic
patterns.

Usage

```
fp.logit(obs, n, dose)
```

Arguments

obs	Numeric vector of the number of patients experiencing the event of interest at each dose level. Must be non-negative integers.
n	Numeric vector of the total number of patients treated at each dose level. Must be positive integers. Length must match obs and dose.
dose	Numeric vector of dose levels investigated. Should be positive values representing actual doses (not dose level indices). Length must match obs and n.

Details

Fractional polynomials extend conventional polynomials by allowing non-integer powers from a predefined set. All commonly used transformations such as the logarithmic, square, cubic, or reciprocal are embedded in the FP method. This approach is especially valuable in dose-finding studies where the true shape of the dose-response curve is unknown and may exhibit complex non-linear behavior.

Mathematical Framework: The fractional polynomial of degree m for a positive variable x takes the form:

$$FP_m(x) = \beta_0 + \sum_{j=1}^{m} \beta_j H_j(x)$$

where $H_j(x)$ represents transformed versions of x using powers from the set {-2, -1, -0.5, 0, 0.5, 1, 2, 3}, with 0 representing the natural logarithm.

Implementation Details: This function implements FP logistic regression with 2 degrees of freedom, using very liberal selection criteria (select=0.99999, alpha=0.99999) to avoid the closed testing procedure and focus on finding the best-fitting model.

Value

A numeric vector of estimated efficacy probabilities for each dose level, corresponding to the input dose vector. Values are bounded between 0 and 1. The estimates are derived from the best-fitting fractional polynomial model selected based on deviance criteria.

References

- Royston, P., & Altman, D. G. (1994). Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 43(3), 429-467.

See Also

mfp for the underlying fractional polynomial fitting algorithm, glm for standard logistic regression, obd.select for optimal biological dose selection using the output from this function.

Examples

```
# Modeling efficacy probabilities in a dose-escalation study
dose_levels <- c(25, 50, 100, 200, 400)  # mg doses
efficacy_responses <- c(1, 3, 8, 12, 10)  # patients with efficacy
total_patients <- c(6, 6, 12, 15, 12)     # total patients per dose

# Fit fractional polynomial model
efficacy_probs <- fp.logit(obs = efficacy_responses,
                           n = total_patients,
                           dose = dose_levels)

# Display results
results <- data.frame(
  Dose = dose_levels,
```

```
    Observed_Rate = efficacy_responses / total_patients,
    FP_Predicted = round(efficacy_probs, 3)
)
print(results)
```

---

gboinet   *gBOIN-ET: Generalized Bayesian Optimal Interval Design for Ordi-*
                                 nal Graded Outcomes

Description

Conducts simulation studies of the gBOIN-ET (generalized Bayesian Optimal Interval design for optimal dose-finding accounting for ordinal graded Efficacy and Toxicity) design. This extension of BOIN-ET utilizes ordinal (graded) outcome information rather than binary endpoints, providing more nuanced dose-finding decisions by incorporating the full spectrum of toxicity severity and efficacy response levels.

Unlike traditional binary approaches that classify outcomes as simply "toxic/non-toxic" or "effective/ineffective," gBOIN-ET recognizes that clinical outcomes exist on a continuum. This design is particularly valuable when the degree of toxicity or efficacy response significantly impacts clinical decision-making and patient outcomes.

Usage

```
gboinet(
  n.dose, start.dose, size.cohort, n.cohort,
  toxprob, effprob, sev.weight, res.weight,
  phi, phi1 = phi*0.1, phi2 = phi*1.4,
  delta, delta1 = delta*0.6,
  alpha.T1 = 0.5, alpha.E1 = 0.5, tau.T, tau.E,
  te.corr = 0.2, gen.event.time = "weibull",
  accrual, gen.enroll.time = "uniform",
  stopping.npts = size.cohort*n.cohort,
  stopping.prob.T = 0.95, stopping.prob.E = 0.99,
  estpt.method = "obs.prob", obd.method = "max.effprob",
  w1 = 0.33, w2 = 1.09,
  plow.ast = phi1, pupp.ast = phi2,
  qlow.ast = delta1/2, qupp.ast = delta,
  psi00 = 40, psi11 = 60,
  n.sim = 1000, seed.sim = 100)
```

Arguments

n.dose   Integer specifying the number of dose levels to investigate.

start.dose   Integer specifying the starting dose level (1 = lowest dose). Generally recom-
             mended to start at the lowest dose for safety.

size.cohort	Integer specifying the number of patients per cohort. Commonly 3 or 6 patients, with 3 being standard for early-phase trials.
n.cohort	Integer specifying the maximum number of cohorts. Total sample size = size.cohort*n.cohort.
toxprob	Matrix (nrow = toxicity categories, ncol = n.dose) specifying true toxicity probabilities. Each column must sum to 1.0. Rows represent ordered toxicity levels from none to most severe.
effprob	Matrix (nrow = efficacy categories, ncol = n.dose) specifying true efficacy probabilities. Each column must sum to 1.0. Rows represent ordered response levels from none to best response.
sev.weight	Numeric vector of toxicity severity weights. Length must equal nrow(toxprob). Should be non-decreasing and reflect clinical impact. First element typically 0 (no toxicity).
res.weight	Numeric vector of efficacy response weights. Length must equal nrow(effprob). Should be non-decreasing and reflect clinical benefit. First element typically 0 (no response).
phi	Numeric target for normalized equivalent toxicity score (nETS). Should be calibrated for weighted scores, not binary probabilities.
phi1	Numeric lower boundary for nETS. Doses with nETS <= phi1 considered underdosed for toxicity. Default phi*0.1.
phi2	Numeric upper boundary for nETS. Doses with nETS >= phi2 trigger de-escalation. Default phi*1.4.
delta	Numeric target for normalized equivalent efficacy score (nEES). Should reflect desired level of clinical benefit.
delta1	Numeric minimum threshold for nEES. Doses below this considered sub-therapeutic. Default delta*0.6.
alpha.T1	Numeric value specifying the probability that a toxicity outcome occurs in the late half of the toxicity assessment window. Used for event time generation. Default is 0.5.
alpha.E1	Numeric value specifying the probability that an efficacy outcome occurs in the late half of the efficacy assessment window. Used for event time generation. Default is 0.5.
tau.T	Numeric value specifying the toxicity assessment window in days. All toxicity evaluations must be completed within this period.
tau.E	Numeric value specifying the efficacy assessment window in days. All efficacy evaluations must be completed within this period.
te.corr	Numeric value between -1 and 1 specifying the correlation between toxicity and efficacy, specified as Gaussian copula parameter. Default is 0.2 (weak positive correlation).
gen.event.time	Character string specifying the distribution for generating event times. Options are "weibull" (default) or "uniform". A bivariate Gaussian copula model is used to jointly generate the time to first ordinal toxicity and efficacy outcome, where the marginal distributions are set to Weibull distribution when gen.event.time="weibull", and uniform distribution when gen.event.time="uniform".

accrual	Numeric value specifying the accrual rate (days), which is the average number of days between patient enrollments. Lower values indicate faster accrual.
gen.enroll.time	
	Character string specifying the distribution for enrollment times. Options are "uniform" (default) or "exponential". Uniform distribution is used when gen.enroll.time="uniform", and exponential distribution is used when gen.enroll.time="exponential".
stopping.npts	Integer specifying the maximum number of patients per dose for early study termination. If the number of patients at the current dose reaches this criteria, the study stops the enrollment and is terminated. Default is size.cohort*n.cohort.
stopping.prob.T	
	Numeric value between 0 and 1 specifying the early study termination threshold for toxicity. If P(nETS > phi) > stopping.prob.T, the dose levels are eliminated from the investigation. Default is 0.95.
stopping.prob.E	
	Numeric value between 0 and 1 specifying the early study termination threshold for efficacy. If P(nEES < delta1) > stopping.prob.E, the dose levels are eliminated from the investigation. Default is 0.99.
estpt.method	Character string specifying the method for estimating efficacy probabilities. Options: "obs.prob" (observed efficacy probabilitiesrates), "fp.logistic" (fractional polynomial), or "multi.iso" (model averaging of multiple unimodal isotopic regression). Default is "obs.prob".
obd.method	Character string specifying the method for OBD selection. Options: "utility.weighted", "utility.truncated.linear", "utility.scoring", or "max.effprob" (default).
w1	Numeric value specifying the weight for toxicity-efficacy trade-off in "utility.weighted" method. Default is 0.33.
w2	Numeric value specifying the penalty weight for toxic doses in "utility.weighted" method. Default is 1.09.
plow.ast	Numeric value specifying the lower toxicity threshold for "utility.truncated.linear" method. Default is phi1.
pupp.ast	Numeric value specifying the upper toxicity threshold for "utility.truncated.linear" method. Default is phi2.
qlow.ast	Numeric value specifying the lower efficacy threshold for "utility.truncated.linear" method. Default is delta1/2.
qupp.ast	Numeric value specifying the upper efficacy threshold for "utility.truncated.linear" method. Default is delta.
psi00	Numeric value specifying the utility score for (toxicity=no, efficacy=no) in "utility.scoring" method. Default is 40.
psi11	Numeric value specifying the utility score for (toxicity=yes, efficacy=yes) in "utility.scoring" method. Default is 60.
n.sim	Integer specifying the number of simulated trials. Default is 1000. Higher values provide more stable operating characteristics.
seed.sim	Integer specifying the random seed for reproducible results. Default is 100.

Details

Conceptual Foundation:

Binary vs Ordinal Paradigm: Traditional designs lose valuable information by dichotomizing outcomes:

- **Binary approach**: Grade 3 or Grade 4 toxicity treated identically as "toxic"
- **Ordinal approach**: Preserves clinically meaningful distinctions between severity levels
- **Information gain**: More efficient use of patient data for dose-finding decisions

Equivalent Toxicity/Efficacy Score Framework: The design converts ordinal categories into continuous scores:

- **ETS (Equivalent Toxicity Score)**: Relative severity of different toxicity grades
- **EES (Equivalent Efficacy Score)**: Relative effectiveness of different efficacy outcomes
- **Normalized scores (nETS, nEES)**: Standardized to a scale ranging from 0 to 1 (quasi-Bernoulli probability)

Decision Algorithm: Uses the same boundary-based logic as BOIN-ET but applied to normalized scores:

- **Escalate**: When nETS <= lambda1 AND nEES <= eta1
- **Stay**: When lambda1 < nETS < lambda2 AND nEES > eta1
- **De-escalate**: When nETS >= lambda2
- **Efficacy-guided**: When lambda1 < nETS < lambda2 AND nEES <= eta1

Key Advantages:

1. Enhanced Discrimination:

- Better differentiation between dose levels with similar binary rates
- Captures clinically meaningful severity gradations

2. Clinical Relevance:

- Aligns with clinical practice where severity matters
- Better reflection of risk-benefit trade-offs

3. Regulatory Appeal:

- Utilizes standard grading systems (CTCAE, RECIST)
- Transparent scoring methodology
- Maintains model-assisted design simplicity

Weight Selection:

Example of Toxicity Weights (sev.weight): Should reflect clinical impact and patient burden:

- **Grade 0 and 1**: 0.0
- **Grade 2**: 0.5
- **Grade 3**: 1.0

- **Grade 4**: 1.5

Example of Efficacy Weights (res.weight): Should reflect clinical benefit and durability:

- **PD**: 0.0
- **SD**: 0.25
- **PR**: 1.0
- **CR**: 3.0

When to Use gBOIN-ET vs TITE-gBOIN-ET:

Choose gBOIN-ET when:

- Outcomes occur within reasonable assessment windows
- Patient accrual allows waiting for complete outcome assessment
- Preference for simpler, well-established approaches

Choose TITE-gBOIN-ET when:

- Late-onset outcomes are expected
- Rapid accrual necessitates continuous enrollment
- Trial duration constraints are critical

Value

A list object of class "gboinet" containing the following components:

toxprob	True toxicity probability matrix used in simulation.
effprob	True efficacy probability matrix used in simulation.
nETS	True normalized equivalent toxicity scores by dose level.
nEES	True normalized equivalent efficacy scores by dose level.
phi	Target normalized equivalent toxicity scores.
delta	Target normalized equivalent efficacy scores.
lambda1	Lower toxicity decision boundary.
lambda2	Upper toxicity decision boundary.
eta1	Lower efficacy decision boundary.
tau.T	Toxicity assessment window (days).
tau.E	Efficacy assessment window (days).
accrual	Accrual rate (days).
ncat.T	Number of ordinal toxicity outcome categories.
ncat.E	Number of ordinal efficacy outcome categories.
estpt.method	Method used for efficacy probability estimation.
obd.method	Method used for optimal biological dose selection.
n.patient	Average number of patients treated at each dose level across simulations.
prop.select	Percentage of simulations selecting each dose level as OBD.
prop.stop	Percentage of simulations terminating early without OBD selection.
duration	Expected trial duration in days.

Note

- Matrix inputs require careful validation - rows must sum to 1.0
- Weight selection should involve clinical stakeholders and reflect patient preferences
- Normalized equivalent scores may not directly correspond to familiar probability scales

References

- Takeda, K., Morita, S., & Taguri, M. (2022). gBOIN-ET: The generalized Bayesian optimal interval design for optimal dose-finding accounting for ordinal graded efficacy and toxicity in early clinical trials. *Biometrical Journal*, 64(7), 1178-1191.
- Yamaguchi, Y., Takeda, K., Yoshida, S., & Maruo, K. (2024). Optimal biological dose selection in dose-finding trials with model-assisted designs based on efficacy and toxicity: a simulation study. *Journal of Biopharmaceutical Statistics*, 34(3), 379-393.

See Also

tite.gboinet for time-to-event version with ordinal outcomes, boinet for binary outcome version, obd.select for dose selection methods, utility.weighted, utility.truncated.linear, utility.scoring for utility functions.

Examples

```
# Example 1: Targeted therapy with hepatotoxicity grading
# Scenario: Kinase inhibitor with dose-dependent liver toxicity

n.dose      <- 5
start.dose  <- 1
size.cohort <- 4  # Slightly larger for ordinal information
n.cohort    <- 12

# Hepatotoxicity categories: Normal, Grade 1, Grade 2, Grade 3+
# Progressive increase in severe hepatotoxicity with dose
toxprob <- rbind(
  c(0.85, 0.70, 0.50, 0.35, 0.20),  # Normal LFTs
  c(0.12, 0.20, 0.25, 0.25, 0.20),  # Grade 1 elevation
  c(0.02, 0.08, 0.20, 0.30, 0.40),  # Grade 2 elevation
  c(0.01, 0.02, 0.05, 0.10, 0.20)   # Grade 3+ hepatotoxicity
)

# Response categories: PD, SD, PR, CR
# Plateau in efficacy at higher doses
effprob <- rbind(
  c(0.70, 0.50, 0.30, 0.25, 0.30),  # Progressive disease
  c(0.25, 0.35, 0.40, 0.35, 0.35),  # Stable disease
  c(0.04, 0.12, 0.25, 0.30, 0.25),  # Partial response
  c(0.01, 0.03, 0.05, 0.10, 0.10)   # Complete response
)

# Hepatotoxicity severity weights (clinical practice-based)
sev.weight <- c(0.0, 0.3, 1.0, 3.0)  # Strong penalty for Grade 3+
```

```
res.weight <- c(0.0, 0.2, 1.5, 3.5)  # Preference for objective responses

# Moderate toxicity tolerance for targeted therapy
phi   <- 0.60  # Accept moderate weighted hepatotoxicity
delta <- 0.80  # Target meaningful weighted efficacy

# Standard assessment windows for targeted therapy
tau.T   <- 42   # 6 weeks for LFT monitoring
tau.E   <- 56   # 8 weeks for response assessment
accrual <- 7    # Weekly enrollment

results_tki <- gboinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  sev.weight = sev.weight, res.weight = res.weight,
  phi = phi, delta = delta,
  tau.T = tau.T, tau.E = tau.E, accrual = accrual,
  estpt.method = "obs.prob",
  obd.method = "utility.weighted",
  w1 = 0.4, w2 = 1.2,
  n.sim = 100
)

# Display normalized equivalent scores (true values)
cat("True Normalized Equivalent Scores:\\n")
cat("nETS (Toxicity):", round(results_tki$nETS, 2), "\\n")
cat("nEES (Efficacy):", round(results_tki$nEES, 2), "\\n")

# Example 2: Chemotherapy with neuropathy grading
# Scenario: Taxane with cumulative peripheral neuropathy

n.dose      <- 4
size.cohort <- 6  # Larger cohorts for safety
n.cohort    <- 8

# Neuropathy categories: None, Mild, Moderate, Severe
# Cumulative dose-dependent neuropathy
toxprob <- rbind(
  c(0.75, 0.55, 0.35, 0.20),  # No neuropathy
  c(0.20, 0.30, 0.35, 0.30),  # Mild neuropathy
  c(0.04, 0.12, 0.25, 0.35),  # Moderate neuropathy
  c(0.01, 0.03, 0.05, 0.15)   # Severe neuropathy
)

# Response categories: No response, Minor, Major, Complete
effprob <- rbind(
  c(0.60, 0.40, 0.25, 0.20),  # No response
  c(0.30, 0.35, 0.35, 0.30),  # Minor response
  c(0.08, 0.20, 0.30, 0.35),  # Major response
  c(0.02, 0.05, 0.10, 0.15)   # Complete response
)
```

```
# Neuropathy-specific weights (functional impact)
sev.weight <- c(0.0, 0.4, 1.2, 2.8)  # Severe neuropathy major QoL impact
res.weight <- c(0.0, 0.3, 1.8, 3.2)  # Complete response highly valued

phi   <- 0.50  # Moderate neuropathy tolerance
delta <- 0.80  # Target substantial response

tau.T   <- 84   # 12 weeks for neuropathy development
tau.E   <- 56   # 8 weeks for response assessment
accrual <- 14   # Bi-weekly enrollment

results_chemo <- gboinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  sev.weight = sev.weight, res.weight = res.weight,
  phi = phi, delta = delta,
  tau.T = tau.T, tau.E = tau.E, accrual = accrual,
  estpt.method = "obs.prob",
  obd.method = "utility.truncated.linear",
  n.sim = 100
)

# Compare with binary approximation
binary_tox <- 1 - toxprob[1,]  # Any neuropathy
binary_eff <- effprob[3,] + effprob[4,]  # Major + Complete response

cat("Ordinal vs Binary Information:\\n")
cat("Binary toxicity rates:", round(binary_tox, 2), "\\n")
cat("Ordinal nETS scores:", round(results_chemo$nETS, 2), "\\n")
cat("Binary efficacy rates:", round(binary_eff, 2), "\\n")
cat("Ordinal nEES scores:", round(results_chemo$nEES, 2), "\\n")
```

---

gridoptim	*Grid Search Optimization for BOIN-ET Design Boundaries*

Description

Performs grid search optimization to determine optimal threshold values for toxicity and efficacy boundaries used in BOIN-ET dose-escalation and de-escalation decisions. This function implements the foundational calibration step for Bayesian Optimal Interval designs, ensuring that the decision boundaries (lambda1, lambda2, eta1) are optimally chosen to minimize incorrect dose selection decisions across various clinical scenarios.

The optimization process balances the competing objectives of avoiding under-dosing (missing therapeutic opportunities), over-dosing (exposing patients to excessive toxicity), and selecting ineffective doses (futility), making it critical for the overall performance of BOIN-ET designs.

Usage

```
gridoptim(pi = rep(1/6, 6), phi, phi1, phi2, delta, delta1, n = 100)
```

Arguments

pi	Numeric vector of length 6 specifying prior probabilities for the six dose-response hypotheses. Must sum to 1. Default is uniform priors (rep(1/6, 6)). Order corresponds to: (1) under-dosed both, (2) under-dosed toxicity only, (3) adequate both, (4) adequate toxicity only, (5) over-dosed toxicity/adequate efficacy, (6) over-dosed toxicity/under-dosed efficacy.
phi	Numeric value specifying the target toxicity probability. Must satisfy phi1 < phi < phi2.
phi1	Numeric value specifying the lower bound of acceptable toxicity probability range. Doses with toxicity <= phi1 are considered under-dosed for toxicity. Must be less than phi.
phi2	Numeric value specifying the upper bound of acceptable toxicity probability range. Doses with toxicity >= phi2 are considered over-dosed. Must be greater than phi.
delta	Numeric value specifying the target efficacy probability. This represents the desired minimum efficacy rate. Must be greater than delta1.
delta1	Numeric value specifying the lower bound of efficacy probability range. Doses with efficacy < delta1 are considered sub-therapeutic. Must be less than delta.
n	Numeric value specifying the reference sample size for boundary optimization. Default is 100. This affects the granularity of decision probabilities but optimal boundaries are relatively robust to this choice within typical phase I sample size ranges.

Details

The grid search optimization addresses the fundamental problem in dose-finding: determining decision boundaries that minimize the probability of incorrect dosing decisions. The method considers six possible true dose-response scenarios:

Grid Search Methodology:

Search Space Construction:

- **lambda1 range**: (phi1,phi) in 0.01 increments
- **lambda2 range**: (phi,phi2) in 0.01 increments
- **eta1 range**: (delta1,delta) in 0.01 increments
- **Total combinations**: Typically 50-200 combinations depending on ranges

Evaluation Criteria: For each boundary combination, the function calculates the probability of incorrect selection by:

1. Computing decision probabilities under each hypothesis
2. Weighting by prior hypothesis probabilities
3. Summing across all incorrect decision scenarios
4. Selecting boundaries that minimize total probability of incorrect decision

Value

A data frame containing the optimal boundary values:

lambda1   Optimal lower toxicity boundary for dose escalation decisions.

lambda2   Optimal upper toxicity boundary for dose de-escalation decisions.

eta1   Optimal lower efficacy boundary for dose selection decisions.

These boundaries should satisfy: phi1 <= lambda1 <= phi <= lambda2 <= phi2 and delta1 <= eta1 <= delta.

Note

- Parameter constraints must be satisfied: phi1 < phi < phi2 and delta1 < delta
- Prior probabilities in pi must sum to 1.0
- Grid search resolution is 0.01, providing good balance of precision and speed
- Results are relatively robust to moderate changes in reference sample size (n)

References

- Takeda, K., Taguri, M., & Morita, S. (2018). BOIN-ET: Bayesian optimal interval design for dose finding based on both efficacy and toxicity outcomes. *Pharmaceutical Statistics*, 17(4), 383-395.

See Also

boinet and tite.boinet which use optimized boundaries, obd.select for dose selection using the optimized boundaries.

Examples

```
phi <- 0.30      # 30% target toxicity
phi1 <- 0.05     # 5% lower toxicity bound
phi2 <- 0.45     # 45% upper toxicity bound
delta <- 0.60    # 60% target efficacy
delta1 <- 0.35   # 35% minimum efficacy threshold

optimal_boundaries <- gridoptim(
  phi = phi, phi1 = phi1, phi2 = phi2,
  delta = delta, delta1 = delta1
)

print(optimal_boundaries)

# Verify boundary relationships
cat("\\nBoundary Verification:\\n")
cat("phi1 <= lambda1 <= phi:", phi1, "<=", optimal_boundaries$lambda1, "<=", phi,
    "->", phi1 <= optimal_boundaries$lambda1 && optimal_boundaries$lambda1 <= phi, "\\n")
cat("phi <= lambda2 <= phi2:", phi, "<=", optimal_boundaries$lambda2, "<=", phi2,
    "->", phi <= optimal_boundaries$lambda2 && optimal_boundaries$lambda2 <= phi2, "\\n")
cat("delta1 <= eta1 <= delta:", delta1, "<=", optimal_boundaries$eta1, "<=", delta,
```

```
    "->", delta1 <= optimal_boundaries$eta1 && optimal_boundaries$eta1 <= delta, "\\n")
```

---

multi.iso	*Model Averaging of Multiple Unimodal Isotonic Regression*

Description

Performs model averaging across multiple unimodal isotonic regression models to estimate efficacy probabilities. This approach addresses the uncertainty in the location of the optimal dose by considering all possible modes (peak response locations).

The method is particularly valuable when the dose-response relationship is expected to be unimodal (single peak) but the location of maximum efficacy is unknown. This commonly occurs with targeted therapies, immunotherapies, and biologics where efficacy may plateau or even decrease at very high doses due to off-target effects or immune suppression.

Usage

```
multi.iso(obs, n)
```

Arguments

obs	Numeric vector specifying the number of patients experiencing the event of interest at each dose level. Must be non-negative integers.
n	Numeric vector specifying the total number of patients treated at each dose level. Must be positive integers with the same length as obs.

Details

Unimodal Isotonic Regression: Unlike standard isotonic regression which assumes monotonic relationships, unimodal isotonic regression allows for:

- **Increasing phase**: Response increases up to a peak
- **Peak response**: Maximum efficacy at the mode
- **Decreasing phase**: Response decreases beyond the peak
- **Constraint**: Non-decreasing before mode, non-increasing after mode

Model Ensemble Approach: The function fits J separate unimodal models (where J = number of doses), each assuming the mode is at a different dose level:

- **Model 1**: Mode at dose 1 (monotonically decreasing)
- **Model 2**: Mode at dose 2 (increase then decrease)
- **...**
- **Model J**: Mode at dose J (monotonically increasing)

Model Averaging: Given the location of the mode to be at dose level k, a frequentist model averaging approach is used for estimating the efficacy probability at each dose level, where the weights are calculated based on a pseudo-likelihood on the unimodal isotonic regression.

Value

The `multi.iso` returns a vector of estimated probabilities for each dose level.

Numeric vector of model-averaged estimated efficacy probabilities for each dose level. The length matches the input vectors, and values are bounded between 0 and 1. The estimates incorporate uncertainty about the mode location through AIC-weighted averaging across all possible unimodal models.

References

- Turner, T. R., & Wollan, P. C. (1997). Locating a maximum using isotonic regression. *Computational Statistics and Data Analysis*, 25, 305-320.

- Tjort, N. L, & Claeskens, G. (2003). Frequentist model average estimators. *Journal of the American Statistical Association*, 98, 879-899.

See Also

`ufit` for underlying unimodal isotonic regression, `pava` for standard isotonic regression, `fp.logit` for alternative dose-response modeling approach, `obd.select` for dose selection using estimated probabilities.

Examples

```
# Basic unimodal model averaging
# Scenario: Targeted therapy with efficacy peak at intermediate dose

# Dose-response data showing unimodal pattern
observed_responses <- c(2, 6, 12, 8, 4)  # Peak at dose 3
total_patients <- c(8, 10, 15, 12, 9)

# Apply multi-unimodal isotonic regression
averaged_probs <- multi.iso(obs = observed_responses, n = total_patients)

# Compare with simple observed probabilities
simple_probs <- observed_responses / total_patients

# Display comparison
results <- data.frame(
  Dose = 1:5,
  Observed_Events = observed_responses,
  Total_Patients = total_patients,
  Simple_Probability = round(simple_probs, 3),
  MultiIso_Probability = round(averaged_probs, 3),
  Difference = round(averaged_probs - simple_probs, 3)
)

cat("Unimodal Model Averaging Results:\\n")
print(results)
```

---

obd.select   *Optimal Biological Dose Selection*

Description

Selects the optimal biological dose (OBD) from a set of candidate doses by balancing toxicity and efficacy considerations using various utility-based methods. The OBD is the dose that optimizes the risk-benefit trade-off, typically used in oncology trials where both safety and therapeutic effect must be considered simultaneously.

Usage

```
obd.select(
  probt, probe, method,
  phi, phi1, phi2,
  delta, delta1,
  tterm, eterm, stopT, stopE,
  w1, w2,
  plow.ast, pupp.ast,
  qlow.ast, qupp.ast,
  psi00, psi11)
```

Arguments

probt	Numeric vector of estimated toxicity probabilities for each dose level. Values should be between 0 and 1.
probe	Numeric vector of estimated efficacy probabilities for each dose level. Values should be between 0 and 1. Must have same length as `probt`.
method	Character string specifying the method for OBD selection. Must be one of: "utility.weighted", "utility.truncated.linear", "utility.scoring", or "max.effprob".
phi	Target toxicity probability (used in "max.effprob" method).
phi1	Lower bound of acceptable toxicity probability range. Used for defining toxicity intervals in some methods.
phi2	Upper bound of acceptable toxicity probability range. Used as toxicity threshold in "utility.weighted" method.
delta	Target efficacy probability.
delta1	Lower bound of acceptable efficacy probability range.
tterm	Numeric vector of probabilities of meeting toxicity stopping criteria for each dose. Same length as `probt`.
eterm	Numeric vector of probabilities of meeting efficacy stopping criteria for each dose. Same length as `probe`.
stopT	Toxicity stopping threshold. Doses are excluded if `tterm < (1 - stopT)`.
stopE	Efficacy stopping threshold. Doses are excluded if `eterm < (1 - stopE)`.

w1	Weight parameter for toxicity-efficacy trade-off in "utility.weighted" method. Higher values emphasize efficacy more heavily.
w2	Weight parameter for penalty imposed on toxic doses in "utility.weighted" method. Higher values penalize toxicity more heavily.
plow.ast	Lower threshold of toxicity for the linear truncated utility function (used in "utility.truncated.linear" method).
pupp.ast	Upper threshold of toxicity for the linear truncated utility function (used in "utility.truncated.linear" method).
qlow.ast	Lower threshold of efficacy for the linear truncated utility function (used in "utility.truncated.linear" method).
qupp.ast	Upper threshold of efficacy for the linear truncated utility function (used in "utility.truncated.linear" method).
psi00	Utility score for the outcome (toxicity=FALSE, efficacy=FALSE) in "utility.scoring" method.
psi11	Utility score for the outcome (toxicity=TRUE, efficacy=TRUE) in "utility.scoring" method.

Details

The function implements four different methods for OBD selection:

Method 1: "utility.weighted" Uses a weighted utility function that combines toxicity and efficacy probabilities with user-specified weights. Higher weight on w1 emphasizes efficacy, while higher weight on w2 penalizes toxicity more heavily.

Method 2: "utility.truncated.linear" Employs piecewise linear utility functions with truncation points for both toxicity and efficacy. This method allows for more flexible modeling of the utility surface with different slopes in different probability ranges.

Method 3: "utility.scoring" Uses a discrete scoring system based on four possible outcomes: (no toxicity, no efficacy), (no toxicity, efficacy), (toxicity, no efficacy), and (toxicity, efficacy).

Method 4: "max.effprob" Maximizes efficacy probability among doses with acceptable toxicity profiles. First identifies the maximum tolerated dose (MTD), then selects the dose with highest efficacy probability among acceptable doses.

The function only considers doses that meet both toxicity and efficacy stopping criteria (controlled by stopT and stopE parameters).

Value

Integer value representing the index of the selected optimal biological dose. The returned value corresponds to the position in the input vectors probt and probe. If no dose meets the stopping criteria, the function behavior depends on the implementation of the underlying utility functions.

Note

- All probability vectors (probt, probe, tterm, eterm) must have the same length.
- Only doses that satisfy both tterm >= (1-stopT) and eterm >= (1-stopE) are considered as candidates.

- If multiple doses have the same maximum utility, the function returns the lowest dose index among them.
- Required parameters vary by method - ensure appropriate parameters are provided for the selected method.

References

- Thall, P. F., & Cook, J. D. (2004). Dose-finding based on efficacy-toxicity trade-offs. *Biometrics*, 60(3), 684-693.
- Yin, G., Li, Y., & Ji, Y. (2006). Bayesian dose-finding in phase I/II clinical trials using toxicity and efficacy odds ratios. *Biometrics*, 62(3), 777-784.
- Houede, N., Thall, P. F., Nguyen, H., Paoletti, X., & Kramar, A. (2010). Utility-based optimization of combination therapy using ordinal toxicity and efficacy in phase I/II trials. *Biometrics*, 66(2), 532-540.
- Zhou, H., Murray, T. A., Pan, H., & Yuan, Y. (2018). Comparative review of novel model-assisted designs for phase I clinical trials. *Statistics in Medicine*, 37(20), 2892-2922.

See Also

utility.weighted, utility.truncated.linear, utility.scoring for the underlying utility functions.

Examples

```
# Example 1: Using utility.weighted method
probt <- c(0.05, 0.15, 0.30, 0.50, 0.65)
probe <- c(0.10, 0.25, 0.45, 0.60, 0.55)
tterm <- c(0.95, 0.90, 0.85, 0.75, 0.60)
eterm <- c(0.90, 0.85, 0.80, 0.85, 0.70)

obd_weighted <- obd.select(
  probt = probt, probe = probe,
  method = "utility.weighted",
  phi2 = 0.35, w1 = 1.0, w2 = 0.5,
  tterm = tterm, eterm = eterm,
  stopT = 0.10, stopE = 0.10
)

# Example 2: Using max.effprob method
obd_maxeff <- obd.select(
  probt = probt, probe = probe,
  method = "max.effprob",
  phi = 0.30,
  tterm = tterm, eterm = eterm,
  stopT = 0.10, stopE = 0.10
)

# Example 3: Using utility.scoring method
obd_scoring <- obd.select(
  probt = probt, probe = probe,
```

```
  method = "utility.scoring",
  psi00 = 0, psi11 = 60,
  tterm = tterm, eterm = eterm,
  stopT = 0.10, stopE = 0.10
)
```

---

print.boinet   *Print boinet*

Description

Display key summary results from boinet.

Usage

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

Arguments

x	Object from boinet.
...	More options to pass to print.

Value

No return values. Key summary results from boinet are displayed with trial design settings.

See Also

[boinet](#)

---

print.gboinet   *Print gboinet*

Description

Display key summary results from gboinet.

Usage

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

Arguments

x   Object from gboinet.

...   More options to pass to print.

Value

No return values. Key summary results from gboinet are displayed with trial design settings.

See Also

gboinet

---

print.tite.boinet   *Print tite.boinet*

Description

Display key summary results from tite.boinet.

Usage

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

Arguments

x   Object from tite.boinet.

...   More options to pass to print.

Value

No return values. Key summary results from tite.boinet are displayed with trial design settings.

See Also

tite.boinet

print.tite.gboinet	*Print tite.gboinet*

Description

Display key summary results from `tite.gboinet`.

Usage

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

Arguments

x	Object from `tite.gboinet`.
...	More options to pass to print.

Value

No return values. Key summary results from `tite.gboinet` are displayed with trial design settings.

See Also

[tite.gboinet](#)

summary.boinet	*Summary method for boinet objects*

Description

Provides a formatted summary of boinet simulation results.

Usage

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

Arguments

object	A boinet result object
...	Additional arguments (currently unused)

Value

Formatted summary output (invisible)

---

summary.gboinet *Summary method for gboinet objects*

Description

Summary method for gboinet objects

Usage

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

Arguments

object	A gboinet result object
...	Additional arguments (currently unused)

Value

Formatted summary output (invisible)

---

summary.tite.boinet *Summary method for tite.boinet objects*

Description

Summary method for tite.boinet objects

Usage

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

Arguments

object	A tite.boinet result object
...	Additional arguments (currently unused)

Value

Formatted summary output (invisible)

---

summary.tite.gboinet   *Summary method for tite.gboinet objects*

Description

Summary method for tite.gboinet objects

Usage

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

Arguments

object	A tite.gboinet result object
...	Additional arguments (currently unused)

Value

Formatted summary output (invisible)

---

tite.boinet   *TITE-BOIN-ET: Time-to-Event Bayesian Optimal Interval Design*

Description

Conducts simulation studies of the TITE-BOIN-ET (Time-to-Event Bayesian Optimal Interval design to accelerate dose-finding based on both Efficacy and Toxicity outcomes) design. This advanced extension of BOIN-ET addresses the practical challenges of late-onset outcomes and rapid patient accrual in modern oncology trials by incorporating time-to-event information and allowing continuous enrollment without waiting for complete outcome assessment.

The TITE-BOIN-ET design is particularly valuable for immunotherapy, targeted therapy, and other novel agents where Late-onset toxicity is common and causes major logistic difficulty for existing adaptive phase I trial designs, which require the observance of toxicity early enough to apply dose-escalation rules for new patients.

Usage

```
tite.boinet(
  n.dose, start.dose, size.cohort, n.cohort,
  toxprob, effprob,
  phi = 0.3, phi1 = phi*0.1, phi2 = phi*1.4,
  delta = 0.6, delta1 = delta*0.6,
  alpha.T1 = 0.5, alpha.E1 = 0.5, tau.T, tau.E,
  te.corr = 0.2, gen.event.time = "weibull",
```

```
accrual, gen.enroll.time = "uniform",
stopping.npts = size.cohort*n.cohort,
stopping.prob.T = 0.95, stopping.prob.E = 0.99,
estpt.method = "obs.prob", obd.method = "max.effprob",
w1 = 0.33, w2 = 1.09,
plow.ast = phi1, pupp.ast = phi2,
qlow.ast = delta1/2, qupp.ast = delta,
psi00 = 40, psi11 = 60,
n.sim = 1000, seed.sim = 100)
```

Arguments

n.dose	Integer specifying the number of dose levels to investigate.
start.dose	Integer specifying the starting dose level (1 = lowest dose). Generally recommended to start at the lowest dose for safety.
size.cohort	Integer specifying the number of patients per cohort. Commonly 3 or 6 patients, with 3 being standard for early-phase trials.
n.cohort	Integer specifying the maximum number of cohorts. Total sample size = size.cohort*n.cohort.
toxprob	Numeric vector of length n.dose specifying the true toxicity probabilities for each dose level. Used for simulation scenarios. Should reflect cumulative toxicity over tau.T period.
effprob	Numeric vector of length n.dose specifying the true efficacy probabilities for each dose level. Used for simulation scenarios. Should reflect cumulative efficacy over tau.E period.
phi	Numeric value between 0 and 1 specifying the target toxicity probability. Represents the maximum acceptable toxicity rate. Default is 0.3 (30%).
phi1	Numeric value specifying the highest toxicity probability that is deemed subtherapeutic such that dose-escalation should be pursued. Doses with toxicity <= phi1 are considered under-dosed. Default is phi*0.1.
phi2	Numeric value specifying the lowest toxicity probability that is deemed overly toxic such that dose de-escalation is needed. Doses with toxicity >= phi2 are considered over-dosed. Default is phi*1.4.
delta	Numeric value between 0 and 1 specifying the target efficacy probability. Represents the desired minimum efficacy rate. Default is 0.6 (60%).
delta1	Numeric value specifying the minimum probability deemed efficacious such that the dose levels with efficacy < delta1 are considered sub-therapeutic. Default is delta*0.6.
alpha.T1	Numeric value specifying the probability that a toxicity outcome occurs in the late half of the toxicity assessment window. Used for event time generation. Default is 0.5.
alpha.E1	Numeric value specifying the probability that an efficacy outcome occurs in the late half of the efficacy assessment window. Used for event time generation. Default is 0.5.
tau.T	Numeric value specifying the toxicity assessment window in days. Should reflect the expected time course of relevant toxicities.

tau.E	Numeric value specifying the efficacy assessment window in days.
te.corr	Numeric value between -1 and 1 specifying the correlation between toxicity and efficacy, specified as Gaussian copula parameter. Default is 0.2 (weak positive correlation).
gen.event.time	Character string specifying the distribution for generating event times. Options are "weibull" (default) or "uniform". A bivariate Gaussian copula model is used to jointly generate the time to first toxicity and efficacy outcome, where the marginal distributions are set to Weibull distribution when gen.event.time="weibull", and uniform distribution when gen.event.time="uniform".
accrual	Numeric value specifying the accrual rate (days), which is the average number of days between patient enrollments. Lower values indicate faster accrual.
gen.enroll.time	
	Character string specifying the distribution for enrollment times. Options are "uniform" (default) or "exponential". Uniform distribution is used when gen.enroll.time="uniform", and exponential distribution is used when gen.enroll.time="exponential".
stopping.npts	Integer specifying the maximum number of patients per dose for early study termination. If the number of patients at the current dose reaches this criteria, the study stops the enrollment and is terminated. Default is size.cohort*n.cohort.
stopping.prob.T	
	Numeric value between 0 and 1 specifying the early study termination threshold for toxicity. If P(toxicity > phi) > stopping.prob.T, the dose levels are eliminated from the investigation. Default is 0.95.
stopping.prob.E	
	Numeric value between 0 and 1 specifying the early study termination threshold for efficacy. If P(efficacy < delta1) > stopping.prob.E, the dose levels are eliminated from the investigation. Default is 0.99.
estpt.method	Character string specifying the method for estimating efficacy probabilities. Options: "obs.prob" (observed efficacy probabilitiesrates), "fp.logistic" (fractional polynomial), or "multi.iso" (model averaging of multiple unimodal isotopic regression). Default is "obs.prob".
obd.method	Character string specifying the method for OBD selection. Options: "utility.weighted", "utility.truncated.linear", "utility.scoring", or "max.effprob" (default).
w1	Numeric value specifying the weight for toxicity-efficacy trade-off in "utility.weighted" method. Default is 0.33.
w2	Numeric value specifying the penalty weight for toxic doses in "utility.weighted" method. Default is 1.09.
plow.ast	Numeric value specifying the lower toxicity threshold for "utility.truncated.linear" method. Default is phi1.
pupp.ast	Numeric value specifying the upper toxicity threshold for "utility.truncated.linear" method. Default is phi2.
qlow.ast	Numeric value specifying the lower efficacy threshold for "utility.truncated.linear" method. Default is delta1/2.
qupp.ast	Numeric value specifying the upper efficacy threshold for "utility.truncated.linear" method. Default is delta.

psi00	Numeric value specifying the utility score for (toxicity=no, efficacy=no) in "utility.scoring" method. Default is 40.
psi11	Numeric value specifying the utility score for (toxicity=yes, efficacy=yes) in "utility.scoring" method. Default is 60.
n.sim	Integer specifying the number of simulated trials. Default is 1000. Higher values provide more stable operating characteristics.
seed.sim	Integer specifying the random seed for reproducible results. Default is 100.

Details

Key Advantages:

1. Continuous Accrual: Unlike standard BOIN-ET which waits for complete outcome assessment, TITE-BOIN-ET allows continuous patient enrollment by utilizing both complete and pending (censored) outcome data. This can significantly reduce trial duration.

2. Late-Onset Outcome Handling: The design explicitly models time-to-event outcomes, making it suitable for:

- Immune-related adverse events that may occur months after treatment
- Delayed efficacy responses common in immunotherapy
- Targeted agents with cumulative toxicity effects

3. Flexible Assessment Windows: Different assessment periods for toxicity (tau.T) and efficacy (tau.E) accommodate the reality that safety and efficacy endpoints often have different time courses.

4. Correlated Outcomes: The design can model correlation between toxicity and efficacy through copula functions, reflecting the biological relationship between these endpoints.

Statistical Methodology:

Time-to-Event Integration: The design uses a weighted likelihood approach where:

- Complete observations receive full weight
- Pending observations receive fractional weight based on follow-up time
- Weight = (observation time) / (assessment window)

Decision Algorithm: At each interim analysis, the design:

1. Updates outcome estimates using complete and pending data
2. Applies the same decision boundaries as BOIN-ET (lambda1, lambda2, eta1)
3. Makes dose escalation/de-escalation decisions
4. Continues enrollment while maintaining safety monitoring

When to Choose TITE-BOIN-ET:

- Expected late-onset toxicity
- Delayed efficacy assessment
- Rapid accrual
- Trial duration is a critical constraint

Consider Standard BOIN-ET When:

- Outcomes occur within 2-4 weeks
- Slow accrual allows waiting for complete data
- Preference for simpler designs

Value

A list object of class "tite.boinet" containing:

toxprob	True toxicity probabilities used in simulation.
effprob	True efficacy probabilities used in simulation.
phi	Target toxicity probability.
delta	Target efficacy probability.
lambda1	Lower toxicity decision boundary.
lambda2	Upper toxicity decision boundary.
eta1	Lower efficacy decision boundary.
tau.T	Toxicity assessment window (days).
tau.E	Efficacy assessment window (days).
accrual	Accrual rate (days).
estpt.method	Method used for efficacy probability estimation.
obd.method	Method used for optimal biological dose selection.
n.patient	Average number of patients treated at each dose level across simulations.
prop.select	Percentage of simulations selecting each dose level as OBD.
prop.stop	Percentage of simulations terminating early without OBD selection.
duration	Expected trial duration in days.

Note

- Accrual rate significantly impacts design performance and trial duration
- Early stopping rules are critical for patient safety in TITE designs

References

- Takeda, K., Morita, S., & Taguri, M. (2020). TITE-BOIN-ET: Time-to-event Bayesian optimal interval design to accelerate dose-finding based on both efficacy and toxicity outcomes. *Pharmaceutical Statistics*, 19(3), 335-349.
- Yamaguchi, Y., Takeda, K., Yoshida, S., & Maruo, K. (2024). Optimal biological dose selection in dose-finding trials with model-assisted designs based on efficacy and toxicity: a simulation study. *Journal of Biopharmaceutical Statistics*, 34(3), 379-393.

See Also

boinet for the standard version without time-to-event modeling, tite.gboinet for the generalized version with ordinal outcomes, obd.select for optimal biological dose selection methods, utility.weighted, utility.truncated.linear, utility.scoring for utility functions.

Examples

```
# Example 1: Immunotherapy trial with delayed immune-related toxicity
# Scenario: CAR-T therapy with cytokine release syndrome and delayed efficacy

n.dose      <- 4  # Four dose levels
start.dose  <- 1
size.cohort <- 6  # Larger cohorts for immunotherapy
n.cohort    <- 8  # Total: 48 patients

# CAR-T dose levels with delayed toxicity pattern
toxprob <- c(0.10, 0.25, 0.40, 0.55)  # Including delayed immune toxicity
effprob <- c(0.20, 0.50, 0.70, 0.75)  # Strong efficacy at higher doses

# Immunotherapy-appropriate targets
phi   <- 0.35  # Higher toxicity tolerance
delta <- 0.60  # Target response rate

# Extended assessment windows for immune effects
tau.T   <- 84   # 12 weeks for immune-related AEs
tau.E   <- 112  # 16 weeks for response assessment
accrual <- 7    # Weekly enrollment

# Delayed toxicity/efficacy parameters
alpha.T1 <- 0.6  # Most toxicity in later period
alpha.E1 <- 0.7  # Most responses delayed
te.corr  <- 0.3  # Moderate positive correlation

results_cart <- tite.boinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  phi = phi, delta = delta,
  alpha.T1 = alpha.T1, alpha.E1 = alpha.E1,
  tau.T = tau.T, tau.E = tau.E,
  te.corr = te.corr, accrual = accrual,
  estpt.method = "obs.prob",  # Conservative for small sample
  obd.method = "utility.weighted",
  w1 = 0.4, w2 = 1.2,  # Balanced approach with toxicity penalty
  n.sim = 40
)

cat("Expected trial duration:", results_cart$duration, "days\\n")
cat("OBD selection probabilities:\\n")
print(results_cart$prop.select)

# Example 2: Targeted therapy with rapid accrual
# Scenario: Tyrosine kinase inhibitor with fast enrollment

n.dose      <- 5
size.cohort <- 3
n.cohort    <- 15  # 45 patients total
```

```
# Targeted therapy dose-response
toxprob <- c(0.05, 0.12, 0.22, 0.35, 0.52)
effprob <- c(0.15, 0.35, 0.55, 0.65, 0.60)  # Plateau effect

phi   <- 0.30
delta <- 0.50

# Shorter windows for targeted therapy
tau.T   <- 28   # 4 weeks for acute toxicity
tau.E   <- 56   # 8 weeks for response
accrual <- 3    # Very rapid accrual (every 3 days)

# More uniform timing
alpha.T1 <- 0.5
alpha.E1 <- 0.5
te.corr  <- 0.1  # Weak correlation

results_tki <- tite.boinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  phi = phi, delta = delta,
  alpha.T1 = alpha.T1, alpha.E1 = alpha.E1,
  tau.T = tau.T, tau.E = tau.E,
  te.corr = te.corr, accrual = accrual,
  gen.event.time = "weibull",
  gen.enroll.time = "exponential",  # Variable enrollment
  estpt.method = "fp.logistic",  # Smooth modeling
  obd.method = "max.effprob",
  n.sim = 40
)

# Compare duration to standard BOIN-ET (hypothetical)
standard_duration <- tau.E + (n.cohort * size.cohort * accrual)
cat("TITE duration:", results_tki$duration, "days\\n")
cat("Standard BOIN-ET would take ~", standard_duration, "days\\n")
cat("Time savings:", standard_duration - results_tki$duration, "days\\n")
```

---

tite.gboinet	*TITE-gBOIN-ET: Time-to-Event Generalized BOIN Design for Ordinal Graded Outcomes*

Description

Conducts simulation studies of the TITE-gBOIN-ET (Time-to-Event generalized Bayesian Optimal Interval design to accelerate dose-finding accounting for ordinal graded Efficacy and Toxicity outcomes) design. This advanced extension incorporates both time-to-event modeling and ordinal (graded) outcome assessment, making it suitable for modern oncology trials where both the severity of toxicity and the degree of efficacy response are clinically meaningful.

The design addresses the reality that clinical outcomes are rarely binary. For example, toxicity may range from mild (Grade 1) to life-threatening (Grade 4), while efficacy can span from no response to complete response. By utilizing this additional information, TITE-gBOIN-ET can make more informed dose selection decisions while maintaining the advantages of time-to-event modeling for delayed outcomes.

Usage

```
tite.gboinet(
  n.dose, start.dose, size.cohort, n.cohort,
  toxprob, effprob, sev.weight, res.weight,
  phi, phi1 = phi*0.1, phi2 = phi*1.4,
  delta, delta1 = delta*0.6,
  alpha.T1 = 0.5, alpha.E1 = 0.5, tau.T, tau.E,
  te.corr = 0.2, gen.event.time = "weibull",
  accrual, gen.enroll.time = "uniform",
  stopping.npts = size.cohort*n.cohort,
  stopping.prob.T = 0.95, stopping.prob.E = 0.99,
  estpt.method = "obs.prob", obd.method = "max.effprob",
  w1 = 0.33, w2 = 1.09,
  plow.ast = phi1, pupp.ast = phi2,
  qlow.ast = delta1/2, qupp.ast = delta,
  psi00 = 40, psi11 = 60,
  n.sim = 1000, seed.sim = 100)
```

Arguments

n.dose	Integer specifying the number of dose levels to investigate.
start.dose	Integer specifying the starting dose level (1 = lowest dose). Generally recommended to start at the lowest dose for safety.
size.cohort	Integer specifying the number of patients per cohort. Commonly 3 or 6 patients, with 3 being standard for early-phase trials.
n.cohort	Integer specifying the maximum number of cohorts. Total sample size = size.cohort*n.cohort.
toxprob	Matrix (nrow = toxicity categories, ncol = n.dose) specifying true toxicity probabilities. Each column must sum to 1.0. Rows represent ordered toxicity levels from none to most severe.
effprob	Matrix (nrow = efficacy categories, ncol = n.dose) specifying true efficacy probabilities. Each column must sum to 1.0. Rows represent ordered response levels from none to best response.
sev.weight	Numeric vector of toxicity severity weights. Length must equal nrow(toxprob). Should be non-decreasing and reflect clinical impact. First element typically 0 (no toxicity). Example: c(0, 0.5, 1.0, 1.5) for Grade 0 and 1, Grade 2, Grade 3, Grade 4.
res.weight	Numeric vector of efficacy response weights. Length must equal nrow(effprob). Should be non-decreasing and reflect clinical benefit. First element typically 0 (no response). Example: c(0, 0.25, 1.0, 3.0) for PD, SD, PR, CR.

phi	Numeric target for normalized equivalent toxicity score (nETS). Should be calibrated for weighted scores, not binary probabilities.
phi1	Numeric lower boundary for nETS. Doses with nETS <= phi1 considered under-dosed for toxicity. Default phi*0.1.
phi2	Numeric upper boundary for nETS. Doses with nETS >= phi2 trigger de-escalation. Default phi*1.4.
delta	Numeric target for normalized equivalent efficacy score (nEES). Should reflect desired level of clinical benefit.
delta1	Numeric minimum threshold for nEES. Doses below this considered sub-therapeutic. Default delta*0.6.
alpha.T1	Numeric value specifying the probability that a toxicity outcome occurs in the late half of the toxicity assessment window. Used for event time generation. Default is 0.5.
alpha.E1	Numeric value specifying the probability that an efficacy outcome occurs in the late half of the efficacy assessment window. Used for event time generation. Default is 0.5.
tau.T	Numeric value specifying the toxicity assessment window in days.
tau.E	Numeric value specifying the efficacy assessment window in days.
te.corr	Numeric value between -1 and 1 specifying the correlation between toxicity and efficacy, specified as Gaussian copula parameter. Default is 0.2 (weak positive correlation).
gen.event.time	Character string specifying the distribution for generating event times. Options are "weibull" (default) or "uniform". A bivariate Gaussian copula model is used to jointly generate the time to first ordinal toxicity and efficacy outcome, where the marginal distributions are set to Weibull distribution when gen.event.time="weibull", and uniform distribution when gen.event.time="uniform".
accrual	Numeric value specifying the accrual rate (days), which is the average number of days between patient enrollments. Lower values indicate faster accrual.
gen.enroll.time	
	Character string specifying the distribution for enrollment times. Options are "uniform" (default) or "exponential". Uniform distribution is used when gen.enroll.time="uniform", and exponential distribution is used when gen.enroll.time="exponential".
stopping.npts	Integer specifying the maximum number of patients per dose for early study termination. If the number of patients at the current dose reaches this criteria, the study stops the enrollment and is terminated. Default is size.cohort*n.cohort.
stopping.prob.T	
	Numeric value between 0 and 1 specifying the early study termination threshold for toxicity. If P(nETS > phi) > stopping.prob.T, the dose levels are eliminated from the investigation. Default is 0.95.
stopping.prob.E	
	Numeric value between 0 and 1 specifying the early study termination threshold for efficacy. If P(nEES < delta1) > stopping.prob.E, the dose levels are eliminated from the investigation. Default is 0.99.

estpt.method	Character string specifying the method for estimating efficacy probabilities. Options: "obs.prob" (observed efficacy probabilitiesrates), "fp.logistic" (fractional polynomial), or "multi.iso" (model averaging of multiple unimodal isotopic regression). Default is "obs.prob".
obd.method	Character string specifying the method for OBD selection. Options: "utility.weighted", "utility.truncated.linear", "utility.scoring", or "max.effprob" (default).
w1	Numeric value specifying the weight for toxicity-efficacy trade-off in "utility.weighted" method. Default is 0.33.
w2	Numeric value specifying the penalty weight for toxic doses in "utility.weighted" method. Default is 1.09.
plow.ast	Numeric value specifying the lower toxicity threshold for "utility.truncated.linear" method. Default is phi1.
pupp.ast	Numeric value specifying the upper toxicity threshold for "utility.truncated.linear" method. Default is phi2.
qlow.ast	Numeric value specifying the lower efficacy threshold for "utility.truncated.linear" method. Default is delta1/2.
qupp.ast	Numeric value specifying the upper efficacy threshold for "utility.truncated.linear" method. Default is delta.
psi00	Numeric value specifying the utility score for (toxicity=no, efficacy=no) in "utility.scoring" method. Default is 40.
psi11	Numeric value specifying the utility score for (toxicity=yes, efficacy=yes) in "utility.scoring" method. Default is 60.
n.sim	Integer specifying the number of simulated trials. Default is 1000. Higher values provide more stable operating characteristics.
seed.sim	Integer specifying the random seed for reproducible results. Default is 100.

Details

Key Advantages:

1. Ordinal Outcome Modeling: Instead of binary toxicity/efficacy, the design uses:

- **Toxicity categories**: e.g., None, Mild, Moderate, Severe
- **Efficacy categories**: e.g., No Response, Partial Response, Complete Response
- **Weighted scoring**: Different severity levels receive different weights
- **Normalized equivalent scores**: nETS (toxicity) and nEES (efficacy)

2. Enhanced Information Utilization: The design captures more granular clinical information by considering:

- Grade 2 toxicity as different from Grade 4 toxicity
- Partial response as different from complete response
- Clinically meaningful gradations within traditional binary endpoints

3. Flexible Weighting Schemes: Users can specify custom weights reflecting clinical importance:

- **sev.weight**: Toxicity severity weights (e.g., 0, 0.5, 1.0, 1.5)
- **res.weight**: Efficacy response weights (e.g., 0, 0.25, 1.0, 3.0)

Statistical Methodology:

Equivalent Toxicity/Efficacy Scores:

The design converts ordinal outcomes to continuous scores:

- **ETS (Equivalent Toxicity Score)**: Weighted sum of toxicity categories
- **EES (Equivalent Efficacy Score)**: Weighted sum of efficacy categories
- **Normalization**: Scores divided by maximum possible weight (nETS, nEES)
- **Decision making**: Uses normalized scores with same boundaries as binary BOIN-ET

Time-to-Event Integration: Combines ordinal scoring with time-to-event methodology:

- Events can occur at different times within assessment windows
- Partial information from censored observations

Matrix Input Structure: Probability matrices define outcome distributions:

- **toxprob**: Matrix where rows = toxicity categories, columns = doses
- **effprob**: Matrix where rows = efficacy categories, columns = doses
- Row sums should equal 1 (probability distributions)
- Allows complex, realistic outcome scenarios

Value

A list object of class "tite.gboinet" containing the following components:

toxprob	True toxicity probability matrix used in simulation.
effprob	True efficacy probability matrix used in simulation.
nETS	True normalized equivalent toxicity scores by dose level.
nEES	True normalized equivalent efficacy scores by dose level.
phi	Target normalized equivalent toxicity scores.
delta	Target normalized equivalent efficacy scores.
lambda1	Lower toxicity decision boundary.
lambda2	Upper toxicity decision boundary.
eta1	Lower efficacy decision boundary.
tau.T	Toxicity assessment window (days).
tau.E	Efficacy assessment window (days).
accrual	Accrual rate (days).
ncat.T	Number of ordinal toxicity outcome categories.
ncat.E	Number of ordinal efficacy outcome categories.
estpt.method	Method used for efficacy probability estimation.

obd.method	Method used for optimal biological dose selection.
n.patient	Average number of patients treated at each dose level across simulations.
prop.select	Percentage of simulations selecting each dose level as OBD.
prop.stop	Percentage of simulations terminating early without OBD selection.
duration	Expected trial duration in days.

Note

- Probability matrices must have rows summing to 1.0 for each dose
- Weight vectors must have same length as corresponding outcome categories
- Normalized scores may require different target values than binary probabilities

References

- Takeda, K., Yamaguchi, Y., Taguri, M., & Morita, S. (2023). TITE-gBOIN-ET: Time-to-event generalized Bayesian optimal interval design to accelerate dose-finding accounting for ordinal graded efficacy and toxicity outcomes. *Biometrical Journal*, 65(7), e2200265.
- Yamaguchi, Y., Takeda, K., Yoshida, S., & Maruo, K. (2024). Optimal biological dose selection in dose-finding trials with model-assisted designs based on efficacy and toxicity: a simulation study. *Journal of Biopharmaceutical Statistics*, 34(3), 379-393.

See Also

`tite.boinet` for binary outcome version, `gboinet` for non-time-to-event ordinal version, `obd.select` for dose selection methods, `utility.weighted`, `utility.truncated.linear`, `utility.scoring` for utility functions.

Examples

```
# Example 1: CAR-T therapy with graded CRS and response levels
# Scenario: 4 dose levels with detailed toxicity/efficacy grading

n.dose      <- 4
start.dose  <- 1
size.cohort <- 6  # Larger cohorts for complex outcomes
n.cohort    <- 8

# Toxicity categories: None, Mild CRS, Moderate CRS, Severe CRS
# Higher doses increase severe CRS probability
toxprob <- rbind(
  c(0.85, 0.70, 0.50, 0.30),  # No CRS
  c(0.10, 0.20, 0.25, 0.25),  # Mild CRS
  c(0.04, 0.08, 0.20, 0.30),  # Moderate CRS
  c(0.01, 0.02, 0.05, 0.15)   # Severe CRS
)

# Efficacy categories: No response, Partial remission, Complete remission, MRD-negative
# Strong dose-response relationship
effprob <- rbind(
```

```
    c(0.70, 0.45, 0.25, 0.15),  # No response
    c(0.25, 0.35, 0.35, 0.25),  # Partial remission
    c(0.04, 0.15, 0.30, 0.40),  # Complete remission
    c(0.01, 0.05, 0.10, 0.20)   # MRD-negative CR
)

# Clinical severity weights
sev.weight <- c(0.0, 0.3, 1.0, 2.5)    # Severe CRS heavily weighted
res.weight <- c(0.0, 0.5, 2.0, 4.0)    # Strong preference for deep responses

# CAR-T appropriate parameters
phi   <- 0.40  # Accept moderate weighted toxicity
delta <- 0.80  # Target substantial weighted efficacy

# Extended assessment for immune effects
tau.T   <- 84   # 12 weeks for CRS resolution
tau.E   <- 168  # 24 weeks for response assessment
accrual <- 14   # Bi-weekly enrollment

# Delayed and correlated outcomes
alpha.T1 <- 0.4  # Earlier CRS onset
alpha.E1 <- 0.8  # Much delayed responses
te.corr  <- 0.4  # Moderate positive correlation

results_cart <- tite.gboinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  sev.weight = sev.weight, res.weight = res.weight,
  phi = phi, delta = delta,
  alpha.T1 = alpha.T1, alpha.E1 = alpha.E1,
  tau.T = tau.T, tau.E = tau.E,
  te.corr = te.corr, accrual = accrual,
  estpt.method = "obs.prob",
  obd.method = "utility.weighted",
  w1 = 0.5, w2 = 1.5,  # Balance with strong toxicity penalty
  n.sim = 40
)

# Display normalized equivalent scores
cat("Normalized Equivalent Toxicity Scores (nETS):\\n")
print(results_cart$nETS)
cat("Normalized Equivalent Efficacy Scores (nEES):\\n")
print(results_cart$nEES)

cat("OBD Selection Probabilities:\\n")
print(results_cart$prop.select)

# Example 2: Immunotherapy with immune-related adverse events
# Scenario: Detailed immune toxicity and response grading

n.dose      <- 5
size.cohort <- 3
```

```
n.cohort   <- 15

# irAE categories: None, Grade 1, Grade 2, Grade 3+
toxprob <- rbind(
  c(0.80, 0.65, 0.45, 0.30, 0.20),  # No irAE
  c(0.15, 0.25, 0.35, 0.35, 0.30),  # Grade 1 irAE
  c(0.04, 0.08, 0.15, 0.25, 0.35),  # Grade 2 irAE
  c(0.01, 0.02, 0.05, 0.10, 0.15)   # Grade 3+ irAE
)

# Response categories: PD, SD, PR, CR
effprob <- rbind(
  c(0.60, 0.45, 0.30, 0.25, 0.30),  # Progressive disease
  c(0.30, 0.35, 0.35, 0.30, 0.25),  # Stable disease
  c(0.08, 0.15, 0.25, 0.30, 0.30),  # Partial response
  c(0.02, 0.05, 0.10, 0.15, 0.15)   # Complete response
)

# Immunotherapy-appropriate weights
sev.weight <- c(0.0, 0.2, 0.8, 2.0)  # Strong penalty for Grade 3+ irAE
res.weight <- c(0.0, 0.1, 1.0, 2.5)  # Prefer objective responses

phi   <- 0.50  # Accept moderate irAE burden
delta <- 0.80  # Target meaningful response

tau.T   <- 112  # 16 weeks for irAE development
tau.E   <- 84   # 12 weeks for response assessment
accrual <- 10   # Weekly enrollment

results_immuno <- tite.gboinet(
  n.dose = n.dose, start.dose = start.dose,
  size.cohort = size.cohort, n.cohort = n.cohort,
  toxprob = toxprob, effprob = effprob,
  sev.weight = sev.weight, res.weight = res.weight,
  phi = phi, delta = delta,
  tau.T = tau.T, tau.E = tau.E, accrual = accrual,
  estpt.method = "fp.logistic",  # Smooth dose-response
  obd.method = "utility.truncated.linear",
  n.sim = 40
)

# Display normalized equivalent scores
cat("Normalized Equivalent Toxicity Scores (nETS):\\n")
print(results_immuno$nETS)
cat("Normalized Equivalent Efficacy Scores (nEES):\\n")
print(results_immuno$nEES)

cat("OBD Selection Probabilities:\\n")
print(results_immuno$prop.select)
```

utility.scoring	*Discrete Scoring Utility Function for Toxicity-Efficacy Trade-offs*

Description

Calculates utility scores using a discrete scoring system based on the four possible combinations of binary toxicity and efficacy outcomes. This approach assigns specific utility values to each outcome combination, providing an intuitive and clinically interpretable method for dose selection that directly reflects clinical preferences for different risk-benefit scenarios.

The scoring utility function is particularly valuable when clinical stakeholders can easily specify their preferences for discrete outcome scenarios rather than dealing with continuous probability trade-offs. It naturally handles the clinical reality that the combination of toxicity and efficacy outcomes may have non-additive effects on clinical utility.

Usage

```
utility.scoring(probt, probe, psi00, psi11)
```

Arguments

probt	Numeric vector of estimated toxicity probabilities for each dose. Values should be between 0 and 1.
probe	Numeric vector of estimated efficacy probabilities for each dose. Values should be between 0 and 1. Must have same length as probt.
psi00	Numeric value specifying the utility score for the outcome combination (Toxicity=No, Efficacy=No). Typically ranges from 30-70, representing the clinical value of avoiding harm while missing therapeutic benefit.
psi11	Numeric value specifying the utility score for the outcome combination (Toxicity=Yes, Efficacy=Yes). Typically ranges from 40-80, representing the net clinical value when efficacy is achieved despite toxicity.

Details

Mathematical Formulation:

The utility function is based on expected utility across four discrete outcomes:

$$U(p_T, p_E) = \sum_{t \in \{0,1\}} \sum_{e \in \{0,1\}} P(T = t, E = e) \cdot \psi_{te}$$

Assuming independence between toxicity and efficacy:

$$U(p_T, p_E) = \psi_{00}(1 - p_T)(1 - p_E) + \psi_{01}(1 - p_T)p_E + \psi_{10}p_T(1 - p_E) + \psi_{11}p_Tp_E$$

Where:

- ψ_{00}: Utility score for (Toxicity=No, Efficacy=No)

- ψ_{01}: Utility score for (Toxicity=No, Efficacy=Yes)
- ψ_{10}: Utility score for (Toxicity=Yes, Efficacy=No)
- ψ_{11}: Utility score for (Toxicity=Yes, Efficacy=Yes)

Default Implementation: This function uses a simplified 2-parameter version with fixed values:

- $\psi_{01} = 100$: Best outcome (efficacy without toxicity)
- $\psi_{10} = 0$: Worst outcome (toxicity without efficacy)
- ψ_{00}: User-specified (typically 30-60)
- ψ_{11}: User-specified (typically 40-80)

Comparison with Other Utility Functions:

vs. Weighted Utility:

- More intuitive for clinical stakeholders
- Better handling of outcome interactions
- Less flexible for continuous trade-offs
- More suitable for committee-based decisions

vs. Truncated Linear:

- Simpler parameter specification
- More appropriate for binary thinking
- Less sophisticated modeling of probability ranges
- Better for stakeholder engagement

Value

Numeric vector of expected utility values for each dose, with the same length as the input probability vectors. Values typically range from 0 to 100, where higher values indicate more desirable risk-benefit profiles. The maximum possible utility is 100 (efficacy without toxicity) and minimum is 0 (toxicity without efficacy).

Note

- The function assumes independence between toxicity and efficacy outcomes
- Fixed utility scores: 100 for (safe, effective) and 0 for (toxic, ineffective)
- User-specified scores (psi00, psi11) should reflect clinical context and stakeholder values

References

- Thall, P. F., & Cook, J. D. (2004). Dose-finding based on efficacy-toxicity trade-offs. *Biometrics*, 60(3), 684-693.
- Houede, N., Thall, P. F., Nguyen, H., Paoletti, X., & Kramar, A. (2010). Utility-based optimization of combination therapy using ordinal toxicity and efficacy in phase I/II trials. *Biometrics*, 66(2), 532-540.

See Also

utility.weighted for continuous trade-off approach, utility.truncated.linear for piecewise linear utility, obd.select for optimal biological dose selection using utilities.

Examples

```
# Example 1: Basic scoring utility calculation
# Scenario: Acute leukemia treatment with curative intent

# Dose-response probabilities
toxicity_probs <- c(0.10, 0.25, 0.40, 0.55, 0.70)
efficacy_probs <- c(0.20, 0.45, 0.65, 0.80, 0.85)

# Curative intent scoring: ineffectiveness penalized, mixed outcome acceptable
psi_safe_ineffective <- 35  # Low score for missing cure opportunity
psi_toxic_effective <- 75   # High score for achieving cure despite toxicity

utilities <- utility.scoring(
  probt = toxicity_probs, probe = efficacy_probs,
  psi00 = psi_safe_ineffective, psi11 = psi_toxic_effective
)

# Display outcome probability breakdown
results <- data.frame(
  Dose = 1:5,
  P_Tox = toxicity_probs,
  P_Eff = efficacy_probs,
  P_Safe_Ineffective = round((1-toxicity_probs) * (1-efficacy_probs), 3),
  P_Safe_Effective = round((1-toxicity_probs) * efficacy_probs, 3),
  P_Toxic_Ineffective = round(toxicity_probs * (1-efficacy_probs), 3),
  P_Toxic_Effective = round(toxicity_probs * efficacy_probs, 3),
  Expected_Utility = round(utilities, 1)
)
print(results)

# Optimal dose selection
optimal_dose <- which.max(utilities)
cat("\\nOptimal dose for curative intent:", optimal_dose,
    "with utility:", round(max(utilities), 1), "\\n")
```

---

utility.truncated.linear

Truncated Linear Utility Function for Toxicity-Efficacy Trade-offs

---

Description

Calculates utility scores using truncated linear functions that provide flexible, piecewise-linear relationships between toxicity/efficacy probabilities and their respective utility contributions. This

approach allows for different slopes in different probability ranges, making it suitable for scenarios where the clinical importance of toxicity or efficacy changes non-linearly across the probability spectrum.

The truncated linear utility function is particularly valuable when clinical preferences exhibit threshold effects, where small changes in probability may have dramatically different clinical implications depending on the probability range.

Usage

```
utility.truncated.linear(probt, probe, tlow, tupp, elow, eupp)
```

Arguments

probt	Numeric vector of estimated toxicity probabilities for each dose. Values should be between 0 and 1.
probe	Numeric vector of estimated efficacy probabilities for each dose. Values should be between 0 and 1. Must have same length as probt.
tlow	Numeric value specifying the lower toxicity threshold. Toxicity probabilities at or below this level receive no penalty (utility component = 1). Should be less than tupp.
tupp	Numeric value specifying the upper toxicity threshold. Toxicity probabilities at or above this level receive maximum penalty (utility component = 0). Should be greater than tlow.
elow	Numeric value specifying the lower efficacy threshold. Efficacy probabilities at or below this level provide no benefit (utility component = 0). Should be less than eupp.
eupp	Numeric value specifying the upper efficacy threshold. Efficacy probabilities at or above this level provide maximum benefit (utility component = 1). Should be greater than elow.

Details

Mathematical Formulation:

The utility function combines separate piecewise linear transformations for toxicity and efficacy:

$$U(p_T, p_E) = f_E(p_E) \times f_T(p_T)$$

Where the efficacy transformation $f_E(p_E)$ is:

$$f_E(p_E) = \begin{cases} 0 & \text{if } p_E <= e_{low} \\ \frac{p_E - e_{low}}{e_{upp} - e_{low}} & \text{if } e_{low} < p_E < e_{upp} \\ 1 & \text{if } p_E >= e_{upp} \end{cases}$$

And the toxicity transformation $f_T(p_T)$ is:

$$f_T(p_T) = \begin{cases} 1 & \text{if } p_T <= t_{low} \\ 1 - \frac{p_T - t_{low}}{t_{upp} - t_{low}} & \text{if } t_{low} < p_T < t_{upp} \\ 0 & \text{if } p_T >= t_{upp} \end{cases}$$

Component Interpretations:

Efficacy Function $f_E(p_E)$**:**

- **Below** e_{low}: No clinical benefit (utility = 0)
- **Between thresholds**: Linear increase in benefit
- **Above** e_{upp}: Maximum benefit achieved (utility = 1)
- **Slope**: $1/(e_{upp} - e_{low})$ represents rate of benefit increase

Toxicity Function $f_T(p_T)$**:**

- **Below** t_{low}: Minimal toxicity concern (utility = 1)
- **Between thresholds**: Linear decrease in acceptability
- **Above** t_{upp}: Unacceptable toxicity (utility = 0)
- **Slope**: $-1/(t_{upp} - t_{low})$ represents rate of penalty increase

Comparison with Other Utility Methods:

vs. Weighted Utility:

- More flexible functional form
- Handles threshold effects better
- Requires more parameter tuning
- Less intuitive for simple trade-offs

vs. Scoring Utility:

- Continuous rather than discrete
- Better for probability-based decisions
- More complex parameter specification
- Can handle intermediate probability values

Value

Numeric vector of utility values for each dose, with the same length as the input probability vectors. Values range between 0 and 1, where higher values indicate more desirable doses. The utility is the product of transformed efficacy and toxicity components.

Note

- Threshold parameters must satisfy: tlow < tupp and elow < eupp
- The function returns the product of efficacy and toxicity transformations
- Utility values are bounded between 0 and 1
- Threshold selection significantly impacts dose selection behavior

References

- Houede, N., Thall, P. F., Nguyen, H., Paoletti, X., & Kramar, A. (2010). Utility-based optimization of combination therapy using ordinal toxicity and efficacy in phase I/II trials. *Biometrics*, 66(2), 532-540.
- Thall, P. F., & Cook, J. D. (2004). Dose-finding based on efficacy-toxicity trade-offs. *Biometrics*, 60(3), 684-693.

See Also

[utility.weighted](#) for simpler weighted trade-off approach, [utility.scoring](#) for discrete outcome scoring method, [obd.select](#) for optimal biological dose selection using utilities.

Examples

```
toxicity_probs <- c(0.05, 0.15, 0.25, 0.40, 0.55)
efficacy_probs <- c(0.10, 0.25, 0.45, 0.60, 0.65)

# Clinical thresholds based on disease context
tox_low <- 0.10   # Below 10% toxicity: minimal concern
tox_upp <- 0.45   # Above 45% toxicity: major concern
eff_low <- 0.15   # Below 15% efficacy: clinically negligible
eff_upp <- 0.55   # Above 55% efficacy: highly satisfactory

utilities <- utility.truncated.linear(
  probt = toxicity_probs, probe = efficacy_probs,
  tlow = tox_low, tupp = tox_upp,
  elow = eff_low, eupp = eff_upp
)

# Display results with transformations
results <- data.frame(
  Dose = 1:5,
  Toxicity = toxicity_probs,
  Efficacy = efficacy_probs,
  Tox_Transform = ifelse(toxicity_probs <= tox_low, 1,
                  ifelse(toxicity_probs >= tox_upp, 0,
                         1 - (toxicity_probs - tox_low)/(tox_upp - tox_low))),
  Eff_Transform = ifelse(efficacy_probs <= eff_low, 0,
                  ifelse(efficacy_probs >= eff_upp, 1,
                         (efficacy_probs - eff_low)/(eff_upp - eff_low))),
  Utility = round(utilities, 3)
)
print(results)

# Optimal dose selection
optimal_dose <- which.max(utilities)
cat("\\nOptimal dose:", optimal_dose, "with utility:",
    round(max(utilities), 3), "\\n")
```

---

utility.weighted   *Weighted Utility Function for Toxicity-Efficacy Trade-off*

Description

Calculates utility scores for dose selection based on a weighted function that balances efficacy benefits against toxicity costs. This utility function provides a linear trade-off approach with an additional penalty for doses exceeding a toxicity threshold, making it particularly suitable for dose-finding scenarios where both the overall toxicity rate and severe toxicity cases need to be appropriately penalized.

The weighted utility approach is intuitive for clinicians as it directly reflects the clinical decision-making process: maximize efficacy while minimizing toxicity, with extra caution for doses that exceed acceptable toxicity levels.

Usage

```
utility.weighted(probt, probe, w1, w2, tox.upper)
```

Arguments

probt
: Numeric vector of estimated toxicity probabilities for each dose. Values should be between 0 and 1.

probe
: Numeric vector of estimated efficacy probabilities for each dose. Values should be between 0 and 1. Must have same length as `probt`. Can be binary probabilities or normalized equivalent scores.

w1
: Numeric value specifying the weight for the toxicity-efficacy trade-off. Represents the rate at which toxicity is traded against efficacy. Higher values indicate greater toxicity aversion. Typically ranges from 0.2 to 1.0.

w2
: Numeric value specifying the additional penalty weight for doses exceeding the toxicity threshold (`tox.upper`). Applied as extra penalty beyond the base w1 penalty. Typically ranges from 0.5 to 2.0.

tox.upper
: Numeric value specifying the upper bound of acceptable toxicity probability. Doses with toxicity exceeding this threshold receive additional penalty w2.

Details

Mathematical Formulation:

The utility function is defined as:

$$U(p_T, p_E) = p_E - w_1 \cdot p_T - w_2 \cdot p_T \cdot I(p_T > \phi_2)$$

Where:

- p_E: Efficacy probability (benefit component)
- p_T: Toxicity probability (cost component)

- w_1: Basic toxicity penalty weight
- w_2: Additional penalty for excessive toxicity
- ϕ_2: Toxicity threshold (tox.upper)
- $I(\cdot)$: Indicator function (1 if condition true, 0 otherwise)

Interpretation of Components:

Base Efficacy Term (p_E):

- Represents the direct clinical benefit
- Higher efficacy increases utility linearly
- Assumes all efficacy is equally valuable

Linear Toxicity Penalty ($w_1 \cdot p_T$):

- Constant penalty per unit increase in toxicity
- Reflects baseline toxicity aversion
- Applied regardless of toxicity level

Threshold Toxicity Penalty ($w_2 \cdot p_T \cdot I(p_T > \phi_2)$):

- Additional penalty only when toxicity exceeds threshold
- Models clinical concern about "unacceptable" toxicity levels
- Creates discontinuity in utility function at threshold

Comparison with Other Utility Functions:

vs. Truncated Linear:

- Simpler parameter structure (3 vs 4 thresholds)
- Less flexible but more interpretable
- Better for scenarios with clear toxicity limits

vs. Scoring:

- Continuous rather than discrete outcomes
- More appropriate when probability estimates are reliable
- Less intuitive than discrete outcome scoring

Value

Numeric vector of utility values for each dose, with the same length as the input probability vectors. Higher utility values indicate more desirable doses. Utility values can be negative when toxicity costs outweigh efficacy benefits.

Note

- Utility values can be negative when toxicity costs exceed efficacy benefits
- The function creates a discontinuity at the toxicity threshold (tox.upper)
- The toxicity threshold (tox.upper) should align with design parameters (typically phi2)

References

- Thall, P. F., & Cook, J. D. (2004). Dose-finding based on efficacy-toxicity trade-offs. *Biometrics*, 60(3), 684-693.
- Yin, G., Li, Y., & Ji, Y. (2006). Bayesian dose-finding in phase I/II clinical trials using toxicity and efficacy odds ratios. *Biometrics*, 62(3), 777-784.

See Also

`utility.truncated.linear` for piecewise linear utility function, `utility.scoring` for discrete outcome scoring approach, `obd.select` for optimal biological dose selection using utilities.

Examples

```
toxicity_probs <- c(0.05, 0.15, 0.30, 0.45, 0.60)
efficacy_probs <- c(0.20, 0.40, 0.60, 0.70, 0.65)

# Moderate toxicity aversion
w1 <- 0.5  # Base penalty
w2 <- 1.0  # Additional penalty for high toxicity
threshold <- 0.35  # 35% toxicity threshold

utilities <- utility.weighted(
  probt = toxicity_probs,
  probe = efficacy_probs,
  w1 = w1, w2 = w2,
  tox.upper = threshold
)

# Display results
dose_comparison <- data.frame(
  Dose = 1:5,
  Toxicity = toxicity_probs,
  Efficacy = efficacy_probs,
  Utility = round(utilities, 3),
  Exceeds_Threshold = toxicity_probs > threshold
)
print(dose_comparison)

# Identify optimal dose
optimal_dose <- which.max(utilities)
cat("Optimal dose:", optimal_dose,
    "with utility:", round(max(utilities), 3), "\n")
```

Index