

Package: dtptite (via r-universe)

October 22, 2026

Title Dose Transition Pathways for Time-to-Event Dose-Finding Designs

Version 0.1.0

Description Implements the Dose Transition Pathways (DTP) look-ahead strategy for time-to-event phase I dose-finding clinical trials with delayed or pending toxicity outcomes. Provides a DTP decorator for the 'escalation' framework, supporting both TITE-CRM (model-based) and TITE-BOIN (model-assisted) base designs.

License GPL (>= 3)

URL <https://github.com/ICR-EPAT/dtptite>

BugReports <https://github.com/ICR-EPAT/dtptite/issues>

Depends R (>= 4.1.0)

Imports BOIN, dplyr, escalation (>= 0.2.0), stats, tibble

Suggests dfcrm, ggplot2, testthat (>= 3.0.0), TITEgBOIN

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev libx11-dev

Repository <https://icr-epat.r-universe.dev>

Date/Publication 2026-10-09 13:42:47 UTC

RemoteUrl <https://github.com/icr-epat/dtptite>

RemoteRef HEAD

RemoteSha c8d6b69994fed4d89e3db66e1a151246918ebbe9

Contents

apply_dtp	2
dtpt_projected_dose	3

.dtp_should_wait	4
.dtp_wait_events	5
.dtp_wait_summary	6
.dtp_wait_time	8
.get_boin_tite	9
.plot.time_to_first_mtd	10
.scenario_mtd	11
.set_dtp_queue_size	12
.stop_for_beta_binomial_toxicity	13
.summary.time_to_first_mtd	14
.time_to_first_mtd	15
.tite_patient_samples	17

Index	18
--------------	-----------

apply_dtp	<i>Apply the DTP look-ahead to a TITE selector factory.</i>
-----------	---

Description

Wraps a parent TITE selector factory with the Dose Transition Pathways projection engine. At fit time, the decorator checks whether waiting for pending patients' follow-up (up to `t_max`) could change the dose recommendation upward.

Usage

```
apply_dtp(
  parent_selector_factory,
  t_max,
  obswin,
  projection_search = c("binary", "linear"),
  verbose = TRUE
)
```

Arguments

parent_selector_factory	A selector_factory (typically from <code>escalation::get_dfcrm_tite()</code> or similar TITE model).
t_max	Maximum allowable wait time (numeric ≥ 0 , or Inf for unlimited). Must use the same time unit as obswin.
obswin	DLT observation window length (e.g. 56 days or 8 weeks). Defines the time-weight mapping: $\text{weight} = \text{time_elapsed} / \text{obswin}$.
projection_search	Strategy used to locate the minimum wait time. "binary" (default) screens the top of the candidate range and then bisection, costing $O(\log t_{\max})$ model fits. "linear" scans every candidate day, costing $O(t_{\max})$ fits. Both return the

same answer whenever the projected dose is monotone non-decreasing in wait time, which holds for TITE-CRM and TITE-BOIN; "linear" is retained as a verification oracle and an escape hatch for non-monotone parents.

verbose If TRUE (default), emit an informational message when t_max covers all remaining follow-up. Set to FALSE in simulation, where this fires on nearly every projection. Affects messaging only, never the returned value.

Value

A dtb_selector_factory object.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
  target = 0.25) |>
  apply_dtb(t_max = 35, obswin = 56)

# Four patients at dose 2: one DLT, and the last patient is only 30% of the
# way through the observation window.
outcomes <- data.frame(dose = c(2, 2, 2, 2), tox = c(0, 1, 0, 0),
  weight = c(1, 1, 1, 0.3), cohort = 1:4)
x <- fit(design, outcomes)

recommended_dose(x)           # de-escalate now...
dtb_should_wait(x)            # ...but waiting could change that
dtb_wait_time(x)              # days to wait
dtb_wait_time(x, type = "weight") # the same wait as a weight increment
dtb_projected_dose(x)         # recommendation after the wait
```

dtb_projected_dose	<i>Projected dose after waiting.</i>
--------------------	--------------------------------------

Description

Projected dose after waiting.

Usage

```
dtb_projected_dose(x, ...)
```

Arguments

x	A dtb_selector object.
...	Additional arguments (unused).

Value

Integer dose level, or NA if waiting has no benefit.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                        target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56)

# Four patients at dose 2: one DLT, and the last patient is only 30% of the
# way through the observation window.
outcomes <- data.frame(dose = c(2, 2, 2, 2), tox = c(0, 1, 0, 0),
                       weight = c(1, 1, 1, 0.3), cohort = 1:4)
x <- fit(design, outcomes)

recommended_dose(x)           # de-escalate now...
dtp_should_wait(x)           # ...but waiting could change that
dtp_wait_time(x)              # days to wait
dtp_wait_time(x, type = "weight") # the same wait as a weight increment
dtp_projected_dose(x)         # recommendation after the wait
```

dtp_should_wait	<i>Should the trialist wait before dosing the next cohort?</i>
-----------------	--

Description

Returns TRUE if the DTP projection found that waiting could lead to a higher dose recommendation within `t_max`.

Usage

```
dtp_should_wait(x, ...)
```

Arguments

<code>x</code>	A <code>dtp_selector</code> object.
<code>...</code>	Additional arguments (unused).

Value

Logical scalar.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                        target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56)

# Four patients at dose 2: one DLT, and the last patient is only 30% of the
# way through the observation window.
outcomes <- data.frame(dose = c(2, 2, 2, 2), tox = c(0, 1, 0, 0),
```

```

weight = c(1, 1, 1, 0.3), cohort = 1:4)
x <- fit(design, outcomes)

recommended_dose(x)          # de-escalate now...
dtp_should_wait(x)           # ...but waiting could change that
dtp_wait_time(x)             # days to wait
dtp_wait_time(x, type = "weight") # the same wait as a weight increment
dtp_projected_dose(x)        # recommendation after the wait

```

dtp_wait_events	<i>Raw DTP wait event log from simulated trials.</i>
-----------------	--

Description

Returns a tidy tibble with one row per DTP wait event fired across all simulated replicates. This is the single source of truth from which all DTP wait summaries are derived; users who want metrics not covered by [dtp_wait_summary\(\)](#) can compute them directly from this table.

Usage

```
dtp_wait_events(x, ...)
```

Arguments

x	A simulations or simulations_collection object.
...	Unused.

Details

Replicates with zero waits contribute zero rows (but are still reflected in `dtp_wait_summary()` with explicit 0s). Replicates from non-DTP designs contribute zero rows and are flagged by the design carrying no event log at all, not by reporting zero waits.

Value

A tibble with columns replicate (and design for a simulations_collection), cohort_idx, dose_before, time_in, time_out, wait_duration, projected_dose, dose_after, ended_by (factor: wait_end/dtp_decision/stopped), dose_delta, effective, num_extensions, n_held, n_missed.

wait_duration is the total elapsed time between when the wait was triggered and when it ended, including any extensions caused by in-wait DLT events: each such event prompts a re-fit, and if the updated fit still requests a wait the original wait_end is bumped out. num_extensions records how many times that happened (0 means the wait ran exactly to its initial projection); as a result wait_duration can exceed the design's t_max.

n_held is the number of patients held through the wait and dosed at wait_end, **including** the cohort opener (the patient the wait is for); it is at least 1 for any wait that doses and ranges up to the cohort size. n_missed is the number who arrived while the wait was running to a full waiting room and were never enrolled. See [set_dtp_queue_size\(\)](#) for the room capacity that bounds n_held and produces n_missed.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                        target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56, verbose = FALSE) |>
  stop_at_n(n = 9)

set.seed(456)
sims <- simulate_trials(
  design,
  num_sims = 2,
  true_prob_tox = c(0.05, 0.15, 0.25, 0.35, 0.60),
  sample_patient_arrivals = function(df) {
    cohorts_of_n(n = 3, mean_time_delta = 1)
  },
  max_time = 56
)

dtp_wait_events(sims)
dtp_wait_summary(sims)
dtp_wait_summary(sims, by_dose = TRUE)
```

dtp_wait_summary	<i>Per-trial DTP wait summary from simulated trials.</i>
------------------	--

Description

Aggregates the raw event log returned by `dtp_wait_events()` into a curated per-replicate (or per-replicate × starting dose) summary.

Usage

```
dtp_wait_summary(x, by_dose = FALSE, ...)
```

Arguments

<code>x</code>	A simulations or simulations_collection object.
<code>by_dose</code>	If TRUE, roll up by (replicate, dose_before); otherwise one row per replicate. Default FALSE.
<code>...</code>	Unused.

Details

With the default `by_dose = FALSE`, one row per replicate with columns:

- `num_waits` — how often DTP fired in this trial
- `total_wait_time` — sum of wait durations

- `wait_fraction` — $\text{total_wait_time} / \text{trial_duration}$, a single comparable "DTP overhead" number. The denominator is the full trial calendar time as returned by `escalation::trial_duration()`, which includes the post-loop "advance to full follow-up" tail. Use the raw event log if you need a decision-phase-only denominator.
- `max_wait_time` — longest single wait (for slot / IRB conversations). Can exceed `t_max` when an in-wait DLT triggers an extension; see `dtp_wait_events()`.
- `num_effective_waits` — number of waits where the post-wait recommendation was strictly higher than the pre-wait recommendation (`dose_after > dose_before`). Captures both outright escalations and waits that recovered from an otherwise unnecessary de-escalation.
- `num_missed` — per-trial sum of the event-log `n_missed`: patients who presented during a wait to a full waiting room and were never enrolled. This is DTP's accrual cost, and is zero whenever no wait ran longer than the room capacity allows.

With `by_dose = TRUE`, one row per (`replicate` × `dose_before`). `wait_fraction` (needs a whole-trial denominator) is dropped and a `mean_wait_time` column is added; `max_wait_time` and `num_effective_waits` are retained per dose level.

Non-DTP designs — those whose simulation did not attach a wait event log at all — return all-NA metric columns so that users can distinguish "design did not run under DTP" from "ran under DTP and never waited" (`num_waits = 0`).

On a `simulations_collection` (e.g. the return value of `simulate_compare`), both accessors prepend a design column naming the arm.

Value

A tibble. See Details for columns.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                       target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56, verbose = FALSE) |>
  stop_at_n(n = 9)

set.seed(456)
sims <- simulate_trials(
  design,
  num_sims = 2,
  true_prob_tox = c(0.05, 0.15, 0.25, 0.35, 0.60),
  sample_patient_arrivals = function(df) {
    cohorts_of_n(n = 3, mean_time_delta = 1)
  },
  max_time = 56
)

dtp_wait_events(sims)
dtp_wait_summary(sims)
dtp_wait_summary(sims, by_dose = TRUE)
```

dtp_wait_time	<i>Minimum wait time before the dose recommendation changes.</i>
---------------	--

Description

Minimum wait time before the dose recommendation changes.

Usage

```
dtp_wait_time(x, type = c("time", "weight"), ...)
```

Arguments

x	A dtp_selector object.
type	Character: "time" (default) returns wait in the user's time units (same as t_max and obswin); "weight" returns the corresponding weight increment.
...	Additional arguments (unused).

Value

Numeric scalar (0 if no benefit from waiting).

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                       target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56)

# Four patients at dose 2: one DLT, and the last patient is only 30% of the
# way through the observation window.
outcomes <- data.frame(dose = c(2, 2, 2, 2), tox = c(0, 1, 0, 0),
                      weight = c(1, 1, 1, 0.3), cohort = 1:4)
x <- fit(design, outcomes)

recommended_dose(x)           # de-escalate now...
dtp_should_wait(x)           # ...but waiting could change that
dtp_wait_time(x)              # days to wait
dtp_wait_time(x, type = "weight") # the same wait as a weight increment
dtp_projected_dose(x)         # recommendation after the wait
```

```
plot.time_to_first_mtd
```

Plot the cumulative incidence of reaching the MTD.

Description

Draws, for each design, the proportion of replicates in which a patient had been dosed at the true MTD by a given point. The curve plateaus at `reached_fraction`; replicates that never reached the MTD keep it below 1.

Usage

```
## S3 method for class 'time_to_first_mtd'
plot(x, axis = c("time", "patients"), ci = FALSE, conf = 0.95, ...)
```

Arguments

<code>x</code>	A <code>time_to_first_mtd</code> object.
<code>axis</code>	"time" (default) plots calendar time since trial start; "patients" plots the number of patients dosed beforehand.
<code>ci</code>	If TRUE, add pointwise Wilson intervals for Monte Carlo error. Default FALSE.
<code>conf</code>	Confidence level for the intervals. Default 0.95.
<code>...</code>	Unused.

Value

A ggplot object.

Examples

```
library(escalation)
skeleton <- c(0.124, 0.25, 0.398, 0.542, 0.666)
designs <- list(
  `TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    stop_at_n(n = 18),
  `DTP + TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    apply_dtp(t_max = 35, obswin = 56, verbose = FALSE) |>
    stop_at_n(n = 18)
)

set.seed(456)
sims <- simulate_compare(
  designs,
  num_sims = 10,
  true_prob_tox = c(0.05, 0.15, 0.25, 0.35, 0.60),
  sample_patient_arrivals = function(df) {
    cohorts_of_n(n = 3, mean_time_delta = 1)
  }
)
```

```

    },
    max_time = 56,
    i_like_big_trials = TRUE
  )

  ttm <- time_to_first_mtd(sims)
  ttm
  summary(ttm)
  if (requireNamespace("ggplot2", quietly = TRUE)) {
    plot(ttm)
  }

```

scenario_mtd	<i>True MTD of a simulated scenario.</i>
--------------	--

Description

Identifies the dose whose true toxicity probability lies within `tol` of the target. When several doses qualify the lowest is returned. When none qualifies the scenario has no MTD and NA is returned with a warning.

Usage

```
scenario_mtd(true_prob_tox, target, tol = 0.05)
```

Arguments

<code>true_prob_tox</code>	Numeric vector of true toxicity probabilities, one per dose.
<code>target</code>	Target toxicity probability.
<code>tol</code>	Half-width of the acceptance band. A dose qualifies when <code>abs(true_prob_tox - target) <= tol</code> . Default 0.05.

Value

Integer dose index, or NA_integer_ when no dose qualifies.

Examples

```

scenario_mtd(c(0.05, 0.15, 0.25, 0.35, 0.60), target = 0.25)

# Two doses inside the band: the lowest is the MTD
scenario_mtd(c(0.05, 0.22, 0.27, 0.50), target = 0.25)

```

set_dtp_queue_size	<i>Set the waiting-room capacity for DTP simulation.</i>
--------------------	--

Description

Recruitment onto the trial is shut for the duration of a DTP wait, but the pool of available patients keeps accruing. Patients who present during a wait are held — up to `queue_size` of them — and dosed at the post-wait recommendation the moment the wait ends. Patients who present when the room is already full are **not enrolled**; they are the accrual cost of waiting, reported as the `n_missed` column of `dtp_wait_events()`.

Usage

```
set_dtp_queue_size(dtp_selector_factory, queue_size)
```

Arguments

<code>dtp_selector_factory</code>	A <code>dtp_selector_factory</code> created by <code>apply_dtp()</code> .
<code>queue_size</code>	Integer ≥ 0 , and no greater than the cohort size. Default (when not set) equals the cohort size, inferred at simulation time from the first <code>sample_patient_arrivals</code> call. Values below 1 behave as 1; see Details.

Details

A patient is therefore missed exactly when `wait_duration > queue_size * accrual_gap`, which concentrates the cost at small cohort sizes.

The patient whose arrival opened the cohort is always held — they are the reason the wait is being taken — so they occupy one slot and values below 1 behave as 1. At cohort size 1 the capacity therefore cannot bind: a single-patient cohort has no room for anyone else whatever this is set to. The knob only does work for cohorts of 2 or more.

Value

The modified `dtp_selector_factory` (invisibly).

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
                       target = 0.25) |>
  apply_dtp(t_max = 35, obswin = 56)

# Hold at most two patients while a wait runs
set_dtp_queue_size(design, 2)
```

stop_for_beta_binomial_toxicity

Stop for excessive toxicity using a Beta-Binomial conjugate posterior.

Description

Creates a stopping decorator that monitors one or more doses and halts the trial when the posterior probability of the true toxicity rate exceeding `tox_threshold` is greater than confidence.

Usage

```
stop_for_beta_binomial_toxicity(
  parent_selector_factory,
  dose = 1,
  tox_threshold,
  confidence = 0.88,
  a = NULL,
  b = NULL,
  n_min = 1
)
```

Arguments

<code>parent_selector_factory</code>	A <code>selector_factory</code> object.
<code>dose</code>	Which dose(s) to monitor. A numeric scalar (e.g. 1), numeric vector (e.g. <code>c(1, 2)</code>), "recommended", or "any". Default: 1.
<code>tox_threshold</code>	Toxicity probability threshold. Required.
<code>confidence</code>	Posterior exceedance cutoff (default: 0.88).
<code>a, b</code>	Beta prior hyperparameters. When NULL (default), resolved at fit time from <code>tox_target()</code> of the parent selector: <code>a = tox_target</code> , <code>b = 1 - tox_target</code> .
<code>n_min</code>	Minimum number of patients at a dose required before the exceedance check fires (default: 1). Mirrors TITEgBOIN's <code>Neli</code> parameter — set to 3 to match TITEgBOIN's default elimination rule.

Value

A `beta_binom_tox_selector_factory` object.

Examples

```
library(escalation)
design <- get_dfcrm_tite(skeleton = c(0.124, 0.25, 0.398, 0.542, 0.666),
  target = 0.25) |>
  stop_for_beta_binomial_toxicity(dose = 1, tox_threshold = 0.25,
    confidence = 0.70)
```

```
# Three DLTs in four patients at the lowest dose stops the trial
outcomes <- data.frame(dose = c(1, 1, 1, 1), tox = c(1, 1, 0, 1),
                       weight = c(1, 1, 1, 1), cohort = 1:4)
x <- fit(design, outcomes)
continue(x)
recommended_dose(x)
```

```
summary.time_to_first_mtd
```

Summarise time-to-first-MTD across replicates.

Description

Summarise time-to-first-MTD across replicates.

Usage

```
## S3 method for class 'time_to_first_mtd'
summary(object, probs = NULL, ...)
```

Arguments

object	A time_to_first_mtd object.
probs	Optional numeric vector of probabilities for additional quantile columns. Must be distinct values between 0 and 1. Default NULL adds none.
...	Unused.

Details

Columns:

- num_sims — replicates simulated
- mtd — dose treated as the true MTD
- num_reached — replicates in which a patient was dosed at mtd
- reached_fraction — num_reached / num_sims; NA when the scenario has no MTD
- median_time_to_mtd — median of the cumulative incidence of reaching the MTD. NA unless more than half of replicates reached it, because the quantile is otherwise undefined — including when exactly half did
- median_n_before_mtd — median patients dosed before the MTD, over replicates that reached it

Replicates that never reached the MTD remain in the denominator; they are not dropped. Any probability supplied in probs adds a q<pct>_time_to_mtd column, computed the same way and likewise NA beyond reached_fraction. The percentage is not rounded, so 0.125 gives q12.5_time_to_mtd; distinct probabilities always get distinct columns.

Value

A tibble with one row per design.

Examples

```
library(escalation)
skeleton <- c(0.124, 0.25, 0.398, 0.542, 0.666)
designs <- list(
  `TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    stop_at_n(n = 18),
  `DTP + TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    apply_dtp(t_max = 35, obswin = 56, verbose = FALSE) |>
    stop_at_n(n = 18)
)

set.seed(456)
sims <- simulate_compare(
  designs,
  num_sims = 10,
  true_prob_tox = c(0.05, 0.15, 0.25, 0.35, 0.60),
  sample_patient_arrivals = function(df) {
    cohorts_of_n(n = 3, mean_time_delta = 1)
  },
  max_time = 56,
  i_like_big_trials = TRUE
)

ttm <- time_to_first_mtd(sims)
ttm
summary(ttm)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(ttm)
}
```

time_to_first_mtd	<i>Time until the first patient is dosed at the true MTD.</i>
-------------------	---

Description

Returns one row per simulated replicate recording when, if ever, a patient was first treated at the scenario's true MTD.

Usage

```
time_to_first_mtd(x, mtd = NULL, tol = 0.05, ...)
```

Arguments

<code>x</code>	A simulations or simulations_collection object.
<code>mtd</code>	Dose index to treat as the true MTD. NULL (default) derives it with <code>scenario_mtd()</code> ; NA declares the scenario to have no MTD.
<code>tol</code>	Passed to <code>scenario_mtd()</code> when deriving the MTD. Default 0.05.
<code>...</code>	Unused.

Details

Times are dosing times measured from the start of the trial ($t = 0$). For designs without DTP a patient is dosed on arrival, so dosing time equals recruitment time; for patients held in a DTP queue it is the instant the wait ends and they are dosed.

Columns:

- `replicate` — index of the simulated trial (and design for a simulations_collection)
- `mtd` — the dose treated as the true MTD
- `time_to_mtd` — dosing time of the first patient given mtd, or NA if no patient was given it before the trial ended
- `reached` — whether any patient was dosed at mtd; NA when the scenario has no MTD
- `n_before_mtd` — patients dosed strictly earlier than that patient, or NA if the MTD was never given

When the scenario has no MTD (no dose within `tol` of the target) every row carries `mtd = NA` and `reached = NA`, distinguishing "no MTD to reach" from "an MTD existed and was not reached" (`reached = FALSE`).

The number of replicates is recorded in a `num_sims` attribute and is used as the denominator by `summary.time_to_first_mtd()`.

Value

A tibble of class `time_to_first_mtd`. See Details for columns.

Examples

```
library(escalation)
skeleton <- c(0.124, 0.25, 0.398, 0.542, 0.666)
designs <- list(
  `TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    stop_at_n(n = 18),
  `DTP + TITE-CRM` = get_dfcrm_tite(skeleton = skeleton, target = 0.25) |>
    apply_dtp(t_max = 35, obswin = 56, verbose = FALSE) |>
    stop_at_n(n = 18)
)

set.seed(456)
sims <- simulate_compare(
  designs,
  num_sims = 10,
```

```

true_prob_tox = c(0.05, 0.15, 0.25, 0.35, 0.60),
sample_patient_arrivals = function(df) {
  cohorts_of_n(n = 3, mean_time_delta = 1)
},
max_time = 56,
i_like_big_trials = TRUE
)

ttm <- time_to_first_mtd(sims)
ttm
summary(ttm)
if (requireNamespace("ggplot2", quietly = TRUE)) {
  plot(ttm)
}

```

tite_patient_samples *Create patient samples with realistic TITE DLT timing.*

Description

Generates a list of [escalation::PatientSample](#) objects whose tox_time values are drawn from $U(0, \text{max_time})$ instead of the default $U(0, 1)$. Pass the result to [escalation::simulate_compare\(\)](#) or [escalation::simulate_trials\(\)](#) via patient_samples.

Usage

```
tite_patient_samples(num_sims, max_time, num_patients = 100)
```

Arguments

num_sims	Number of simulation replicates.
max_time	DLT observation window (e.g. 56 days). Must match the max_time argument passed to <code>simulate_compare / simulate_trials</code> .
num_patients	Number of latent patients per sample (default 100).

Value

A list of length num_sims, each element a PatientSample.

Examples

```

samples <- tite_patient_samples(num_sims = 2, max_time = 56)
length(samples)

```

Index

`apply_dtp`, [2](#)
`apply_dtp()`, [12](#)

`dtp_projected_dose`, [3](#)
`dtp_should_wait`, [4](#)
`dtp_wait_events`, [5](#)
`dtp_wait_events()`, [6](#), [7](#), [12](#)
`dtp_wait_summary`, [6](#)
`dtp_wait_summary()`, [5](#)
`dtp_wait_time`, [8](#)

`escalation::get_dfcrm_tite()`, [2](#)
`escalation::PatientSample`, [17](#)
`escalation::simulate_compare()`, [17](#)
`escalation::simulate_trials()`, [17](#)
`escalation::trial_duration()`, [7](#)

`get_boin_tite`, [9](#)

`plot.time_to_first_mtd`, [10](#)

`scenario_mtd`, [11](#)
`scenario_mtd()`, [16](#)
`set_dtp_queue_size`, [12](#)
`set_dtp_queue_size()`, [5](#)
`stop_for_beta_binomial_toxicity`, [13](#)
`summary.time_to_first_mtd`, [14](#)
`summary.time_to_first_mtd()`, [16](#)

`time_to_first_mtd`, [15](#)
`tite_patient_samples`, [17](#)