Package ‘DTAT’

May 25, 2024

Type Package

Title Dose Titration Algorithm Tuning

Version 0.3-7

Date 2024-05-24

Maintainer David C. Norris <david@precisionmethods.guru>

Depends R (>= 3.5.0), survival

Imports km.ci, pomp, Hmisc, data.table, dplyr, r2d3, shiny, jsonlite, methods

Suggests knitr, rmarkdown, lattice, latticeExtra, widgetframe, tidyr, RColorBrewer, invgamma, zipfR, rms

Description Dose Titration Algorithm Tuning (DTAT) is a methodologic framework allowing dose individualization to be conceived as a continuous learning process that begins in early-phase clinical trials and continues throughout drug development, on into clinical practice. This package includes code that researchers may use to reproduce or extend key results of the DTAT research programme, plus tools for trialists to design and simulate a ‘3+3/PC’ dose-finding study. Please see Norris (2017a) <doi:10.12688/f1000research.10624.3> and Norris (2017c) <doi:10.1101/240846>.

URL https://precisionmethods.guru/

License MIT + file LICENSE

RoxygenNote 7.2.3

VignetteBuilder knitr, rmarkdown

Encoding UTF-8

NeedsCompilation no

Author David C. Norris [aut, cre]

Repository CRAN

Date/Publication 2024-05-25 02:20:03 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTAT-package</td>
<td>2</td>
</tr>
<tr>
<td>as_d3_data,DE-method</td>
<td>3</td>
</tr>
<tr>
<td>DE-class</td>
<td>3</td>
</tr>
<tr>
<td>de.bioRxiv.240846</td>
<td>4</td>
</tr>
<tr>
<td>dose.survfit</td>
<td>5</td>
</tr>
<tr>
<td>dose.survival</td>
<td>6</td>
</tr>
<tr>
<td>ds.curve</td>
<td>7</td>
</tr>
<tr>
<td>dtat1000</td>
<td>8</td>
</tr>
<tr>
<td>newton.raphson</td>
<td>10</td>
</tr>
<tr>
<td>Onoue.Friberg</td>
<td>11</td>
</tr>
<tr>
<td>plot,DE,missing-method</td>
<td>13</td>
</tr>
<tr>
<td>runDTATapp</td>
<td>14</td>
</tr>
<tr>
<td>scaled</td>
<td>15</td>
</tr>
<tr>
<td>seq.function</td>
<td>15</td>
</tr>
<tr>
<td>sim</td>
<td>16</td>
</tr>
<tr>
<td>titrate</td>
<td>17</td>
</tr>
<tr>
<td>titration</td>
<td>18</td>
</tr>
</tbody>
</table>

Index 20

---

**Description**

Dose Titration Algorithm Tuning (DTAT) is a methodologic framework allowing dose individualization to be conceived as a continuous learning process that begins in early-phase clinical trials and continues throughout drug development, on into clinical practice. This package includes code that researchers may use to reproduce or extend key results of the DTAT research programme, plus tools for trialists to design and simulate a ’3+3/PC’ dose-finding study. Please see Norris (2017a) doi:10.12688/f1000research.10624.3 and Norris (2017c) doi:10.1101/240846.

**Author(s)**

David C. Norris

**References**


---

### as_d3_data,DE-method

Convert a DE object to JSON

#### Description

Convert a DE object to JSON

#### Usage

```r
## S4 method for signature 'DE'
as_d3_data(x, ...)
```

#### Arguments

- `x` An object of class `DE`
- `...` Unused.

#### DE-class

*An S4 class for simulating dose-titration study designs*

#### Description

An S4 class for simulating dose-titration study designs

#### Slots

- `doses` A numeric vector of prospectively-determined discrete doses to trial.
- `units` A string indicating dose units, e.g. "mg/kg".
- `MTDi` A numeric vector of optimal doses for simulated study participants. Optionally a call to an `r<distribution>(...)` function which may be parsed to calculate the `mtd_quantiles` slot.
- `mtd_quantiles` A numeric vector of quantiles of the distribution from which the MTDi slot was simulated. Intended mainly to support visualization of this distribution, e.g. as an transparent overlay on the dose-survival plot. NULL in case MTDi is provided verbatim.
- `fractol` A numeric vector of probabilities for the simulated MTDi slot. Intended mainly to support visualization, e.g. plotting of 'MTD pointers' on the interactive dose-survival plot.
A data.frame with columns:

- **id**: Participant identifier
- **period**: DLT assessment period, numbered consecutively from 1
- **dose**: Dose level, numbered consecutively starting from 1
- **dlt**: A logical indicator: did this participant experience a DLT during this period?
- **stop_esc**: integer Period in which 'stop rule' was triggered
- **ds_conf_level**: numeric Confidence level for confidence band around Kaplan-Meier estimate of the dose-survival curve.
- **dose_drop_threshold**: numeric Threshold for triggering the 'bypass rule'.
- **stop_esc_under**: numeric Threshold for triggering the 'stop rule'.
- **undo_esc_under**: numeric Threshold for triggering the 'rollback rule'.

---

**Description**

This is a length-10 list of data frames, summarizing the simulated trial from this paper, at the end of periods 1, 2, ..., 10. This structure reflects an awkward S3 implementation that package DTAT v0.3 reimplemented using S4. This data set is retained to support regression tests.

**Format**

A length-10 list of data frames, each with the following columns:

- **id**: Participant identifier
- **period**: DLT assessment period, numbered consecutively from 1
- **dose**: Dose level, numbered consecutively starting from 1
- **dlt**: A logical indicator: did this participant experience a DLT during this period?

**Details**

A `stop.esc` attribute is attached to data frames in this list, indicating when escalation stopped during the simulated trial.

**References**

Examples

data(de.bioRxiv.240846)
# Demonstrate that the new S4 3+3/PC implementation reproduces the
# simulated trial from the paper:
set.seed(2017)
CV <- 0.7; mean_mtd <- 1.0
shape <- CV^-2; scale <- mean_mtd/shape
trial <- new("DE", doses=0.25 * 1.4^(0:6),
    MTDi=rgamma(24, shape=shape, scale=scale),
    units="mg")
trial <- titration(trial, periods=10)
stopifnot(all(trial@data == de.bioRxiv.240846[[10]])
stopifnot(trial@stop_esc == attr(de.bioRxiv.240846[[10]],'stop.esc'))

dose.survfit

Calculate a dose-survival curve from a dose titration study, adding a
confidence band

description

The 'dose-survival curve' is nothing other than an empirical cumulative distribution for MTDi in the
sampled population. The term 'survival' is suggested in part by our application of the Kaplan-Meier
estimator to interval-censored toxicity information.

Usage

dose.survfit(de, method = "rothman", avoid.degeneracy = TRUE, conf.level = 0.8)

Arguments

de A dose titration experiment like the data slot of class DE
method The method to be used by km.ci when calculating CI
avoid.degeneracy When TRUE, this parameter directs the function to introduce artificial events
into the dose titration experiment, to avoid degeneracies at the lower and upper
ends of the dose-survival curve.
conf.level Confidence level for KM confidence band.

Details

TODO: Describe details of degeneracy avoidance, once these have stabilized.

Value

An object of class survfit.
dose.survival

Author(s)

David C. Norris

See Also

dose.survival, km.ci

Examples

CV <- 0.7; mean_mtd <- 1.0
shape <- CV^-2; scale <- mean_mtd/shape
trial <- new("DE", doses=0.25 * 1.4^(0:6),
   MTDi=rgamma(24, shape=shape, scale=scale),
   units="mg")
trial <- titration(trial, periods=10)
sf <- dose.survfit(trial@data)
summary(sf)

dose.survival

Extract interval-censored dose tolerance data from a dose titration study

Description

Constructs a Surv object from a dose-escalation experiment, using interval-censoring constructs of type='interval2'.

Usage

dose.survival(de)

Arguments

de A data frame describing a dose-titration study

Value

A Surv object of type='interval2'

Author(s)

David C. Norris

See Also

dose.survfit
**Examples**

```r
CV <- 0.7; mean_mtd <- 1.0
shape <- CV^-2; scale <- mean_mtd/shape
trial <- new("DE", doses=0.25 * 1.4^(0:6),
           MTDi=rgamma(24, shape=shape, scale=scale),
           units="mg")
trial <- titration(trial, periods=10)
dose.survival(trial@data)
```

---

**ds.curve**  
*Extract the dose-survival curve, with its upper and lower confidence band limits*

---

**Description**

This utility function simply makes the results of `dose.survfit` available in the convenient form of a list.

**Usage**

```r
ds.curve(de, ...)
```

**Arguments**

- `de`  
  A data frame describing a dose-titration study.

- `...`  
  Passed through to function `dose.survfit`

**Value**

A list with components `surv`, `upper` and `lower`, each containing a vector that can be indexed by dose level.

**Author(s)**

David C. Norris

**See Also**

`dose.survfit`
Examples

CV <- 0.7; mean_mtd <- 1.0
shape <- CV^-2; scale <- mean_mtd/shape
trial <- new("DE", doses=0.25 * 1.4^(0:6),
        MTDi=rgamma(24, shape=shape, scale=scale),
        units="mg")
trial <- titration(trial, periods=10)
ds.curve(trial@data)


dtat1000  Precomputed neutrophil-guided chemotherapy dose titration for 1000
           simulated subjects.

Description

This dataset is provided to support fast reproduction of a forthcoming pharmacoeconomic paper
that includes examination of the empirical distribution of MTDi in N=1000 simulated subjects.

Format

A data frame showing end-of-cycle state of neutrophil-guided dose titration for 1000 simulated
subjects, across 10 cycles of chemotherapy.

  cycle  Cycle number 1..10
  id     Subject identifiers; an ordered factor with levels id1 < ... < id1000
  Cc     Central-compartment drug concentration
  Cp     Peripheral-compartment drug concentration
  Prol   Progenitor cells in proliferating compartment of Friberg et al. (2002) model
  Tx.1   Transit compartment 1
  Tx.2   Transit compartment 1
  Tx.3   Transit compartment 1
  Circ   Concentration (cells/mm^3) of circulating neutrophils
  dose   Dose of 1-hour infusion administered this cycle
  CircMin Neutrophil nadir (cells/mm^3)
  tNadir Time (days) of neutrophil nadir
  scaled.dose Fourth root of dose
  time   Time (weeks) of dose administration

Details

Running the examples interactively, you can verify the reproducibility of this dataset. (That demo
is included in a donttest block to spare the CRAN servers.)
References


Examples

data(dtat1000)
# 1. Extract the N final doses, assuming convergence by the tenth course
MTD_i <- with(dtat1000, dose[time==27])
MTD_i <- MTD_i[MTD_i < 5000] # Exclude few outliers
# 2. Do a kernel density plot
library(Hmisc)
library(latticeExtra)
hist <- histogram(~MTD_i, breaks=c(0,100,200,300,400,600,900,1500,2500,4000,5000)
, xlab=expression(MTD[i])
approx <- data.frame(mtd_i=seq(0, 5000, 10))
approx <- upData(approx,
    gamma = dgamma(mtd_i, shape=1.75, scale=200))
dist <- xyplot(gamma ~ mtd_i, data=approx, type='l', col='black', lwd=2)
library(grid)
hist + dist
grid.text(expression(MTD[i] %~% paste("Gamma(", alpha==1.75, ", ", beta==1/200,")"))
, x=unit(0.5,"npc")
, y=unit(0.75,"npc")
)

## A very long repro, which a user of this package may well wish to verify
## by running the examples interactively, although it takes many minutes
## to compute. (Enclosed in a dontest block to avoid overburdening CRAN.)

# Demonstrate close reproduction of original titration (the titration takes many minutes!)
set.seed(2016)
library(pomp)
Onoue.Friberg(N=1000)
# This titration may take an hour to run ...
chemo <- titrate(doserange = c(50, 3000),
    dta=newton.raphson(dose1 = 100,
        omega = 0.75,
        slope1 = -2.0)
)

dtat1k <- upData(chemo$course
    , time = 3*(cycle-1)
    , labels = c(time="Time")
    , units = c(time="weeks")
    , print = FALSE)
c10dose1k <- subset(dtat1k, cycle==10)$scaled.dose
c10dose1000 <- subset(dtat1000, cycle==10)$scaled.dose
stopifnot(0.999 < cor(c10dose1k, c10dose1000))

newton.raphson <- function(dose1, omega, slope1, slopeU)

Arguments

- **dose1**: The starting dose for titration
- **omega**: A relaxation parameter used to moderate dose increments
- **slope1**: Dose-response slope assumed prior to 2nd measured neutrophil nadir
- **slopeU**: Upper bound imposed on slope estimates

Details

This function manifests the core concept of Dose Titration Algorithm Tuning by delivering an objectively realized 'DTA'. It therefore enables a variety of DTAs to be implemented and compared.

Value

A dose titration function that advises dose for next cycle of chemotherapy.

Author(s)

David C. Norris

See Also

titraternewton.raphson

**newton.raphson**

*A dose titration algorithm (DTA) 'factory' based on the Newton-Raphson heuristic*

Description

This higher-order ('factory') function produces a simple dose titration algorithm for neutrophil-guided chemotherapy dosing.

Usage

newton.raphson(dose1, omega, slope1, slopeU)
Onoue.Friberg


Description

This function produces a POMP model combining docetaxel pharmacokinetics (PK) drawn from Table 2 of Onoue et al (2016) with myelosuppression dynamics drawn from Friberg et al (2002). This model enables simulation of neutrophil-guided dose titration of docetaxel, as done in Norris (2017).

Usage

Onoue.Friberg(
  N,
  cycle.length.days = 21,
  data = data.frame(time = c(seq(0, 1.95, 0.05), seq(2, cycle.length.days * 24, 1)), y = NA),
  delta.t = 0.1
)

Arguments

  N  Size of simulated population.
  cycle.length.days  Duration (in days) of chemotherapy cycle to be simulated.
  data  Passed through as the data argument of the pomp constructor.
  delta.t  Time-step (in hours) of pomp’s euler plug-in.

Value

No value is returned; rather, the function sets global variables in package environment DTAT::sim.

Author(s)

David C. Norris

References


See Also

pomp, sim

Examples

# Reproduce the sim$pkpd model and sim$pop population from reference #3:
library(pomp)
Onoue.Friberg(N=25)
sim$pop # NB: this differs from pop of original paper...

# Whereas the present version of Onoue.Friberg() draws simulated populations
# using pomp::rprior(), to reproduce the original F1000Research paper [3] we
# re-draw sim$pop as originally & prosaically done (see https://osf.io/vwnqz/):
set.seed(2016)
N <- 25
dtx.mm <- 0.808 # molar mass (mg/μM) of docetaxel
sim$pop$Circ0 <- rlnorm(N, meanlog=log(5050), sdlog=0.42) # units=cells/mm^3
sim$pop$MTT <- rlnorm(N, meanlog=log(89.3), sdlog=0.16) # mean transit time
sim$pop$gamma <- rlnorm(N, meanlog=log(0.163), sdlog=0.039) # feedback factor
sim$pop$Emax <- rlnorm(N, meanlog=log(83.9), sdlog=0.33)

# PK params from 2-compartment docetaxel model of Onoue et al (2016)
sim$pop$CL <- rlnorm(N, meanlog=log(32.6), sdlog=0.295)
sim$pop$Q <- rlnorm(N, meanlog=log(5.34), sdlog=0.551)
sim$pop$Vc <- rlnorm(N, meanlog=log(5.77), sdlog=0.1) # Onoue gives no CV% for V1

# Now we run the sim$pkpd model, separately for each of N simulated individuals:
allout <- data.frame() # accumulator for N individual ODE solutions
for (id in 1:sim$N) {
  out <- trajectory(sim$pkpd,
    params=c(sim$pop[[sim$pop$id==id, -which(names(sim$pop) %in% c('id', 'MTT'))]],
      sigma=0.05, dose=100, duration=1),
    format="data.frame"
  # drop 'traj' and shift 'time' to first column
  out <- out[,c('time', setdiff(colnames(out), c('time', 'traj')))]
  out$id <- paste("id", id, sep="")
  allout <- rbind(allout, out)
}

library(Hmisc)
allout <- upData(allout
  id = ordered(id, levels=paste("id", 1:sim$N, sep=""))
  units=c("cells/mm^3", "cells/mm^3",
    "cells/mm^3",
    "cells/mm^3","cells/mm^3")
library(tidyr)
cout <- gather(allout, key="Series", value="Concentration"
  , Cc, Cp
  , factor_key = TRUE)

label(cout$Concentration) <- "Drug Concentration"

# Figure 1 from reference [3]:
library(RColorBrewer)
xYplot(Concentration ~ time | id, group=Series
  , data=cout, subset=time<6
  , layout=c(5,NA)
  , type='l', as.table=TRUE
  , label.curves=FALSE
  , par.settings=list(superpose.line=list(lwd=2,col=brewer.pal(4,"PRGn")[c(1,4)]))
  , scales=list(y=list(log=TRUE, lim=c(10^-3,10^1)))
  , auto.key=list(points=FALSE, lines=TRUE, columns=2))

mout <- gather(allout, key="Series", value="ANC"
  , Prol, Tx.1, Tx.2, Tx.3, Circ
  , factor_key = TRUE)

mout <- upData(mout
  , time = time/24
  , units = c(time="days")
  , print = FALSE)

# Figure 3 from citation [3]:
xYplot(ANC ~ time | id, group=Series
  , data=mout
  , layout=c(5,5)
  , type='l', as.table=TRUE
  , label.curves=FALSE
  , par.settings=list(superpose.line=list(lwd=2,col=brewer.pal(11,"RdYlBu")[c(1,3,4,8,10)]))
  , scales=list(y=list(log=TRUE, lim=c(100,15000), at=c(200, 500, 1000, 2000, 5000, 10000)))
  , auto.key=list(points=FALSE, lines=TRUE, columns=5))
Usage

## S4 method for signature 'DE,missing'
plot(x, y, ..., devtree = FALSE)

Arguments

x An object of class DE

y Unused; included for S4 generic consistency

... Passed to \texttt{r2d3}, enabling caller to (e.g.) override the default \texttt{viewer} = "internal".

devtree Logical indicator used to select local package dir

---

runDTATapp \hspace{1cm} Run Shiny apps included in package DTAT

Description

Run Shiny apps included in package DTAT

Usage

runDTATapp(app)

Arguments

app Character vector of length 1. Name of app to run.

Value

Invoked for side effect. Runs the named Shiny app.

Examples

if(interactive()){
  runDTATapp("Sim33PC")
  runDTATapp("TheCost")
}
**scaled**

*Power-law scaling for doses*

**Description**

Implement an inverse power-law scaling for drug dose.

**Usage**

```r
scaled(dose, a = 4)
```

**Arguments**

- **dose**: A numeric vector of doses
- **a**: A numeric exponent for power-law rescaling

**Value**

A rescaled vector of doses

**Author(s)**

David C. Norris

**seq.function**

*A seq method supporting custom-scaled plot axes.*

**Description**

This provides a seq method for class `function`, supporting a natural axis scaling idiom.

**Usage**

```r
## S3 method for class 'function'
seq(scalefun, from, to, length.out, digits = NULL, ...)
```

**Arguments**

- **scalefun**: A numeric function that will be invoked componentwise, and so need not be vectorized)
- **from, to**: The starting and ending values of the sequence returned
- **length.out**: Desired length of the sequence
- **digits**: If non-NULL, returned value is rounded accordingly
- **...**: Unused; included for S3 generic/method consistency.
Value

A numeric vector that (not considering the effect of any rounding applied), becomes an arithmetic sequence after application of scalefun to it. The initial and final elements of that vector are from and to.

Author(s)

David C. Norris

Examples

# Provide evenly-spaced length-6 sequence from 100 to 1000, # evenly spaced on a fourth-root scale:
seq(function(dose, a=4.0) dose^(1/a), from=100, to=1000, length.out=6, digits=0)

---

sim

Environment for simulation global variables.

Description

To simplify the code of package DTAT, as well as client tasks, this exported environment contains a handful of global variables useful for the simulations.

Details

Global variables maintained within environment `sim` are:

1. pkpd: The population PK/PD model to be simulated.
2. pop: A sample drawn from the population model.
3. N: Restricts simulation to first N subjects in pop.
4. params.default: Default parameters.

Examples

# Even when nrow(pop) is large, one may easily restrict # time-consuming simulations to pop[1:N,], as follows:
sim$N <- 25
# Now perform simulation work
## Not run:
titrate(...)
Perform neutrophil-guided dose titration of a chemotherapy drug.

Description
This is included in package DTAT mainly for archival purposes, with the aim to document a reproduction of Figure 5 from the 2017 *F1000Research* paper (referenced below), using a clearer and more general software design than is found in the online code supplement available at https://osf.io/vwnqz/.

Usage
```r
titrater(draw.days = NULL, Ncycles = 10, doserange = c(50, 500), dta = NULL)
```

Arguments
- `draw.days` Integer days on which ANC is to be measured
- `Ncycles` Number of chemo cycles through which to simulate titration
- `doserange` Range of doses to consider
- `dta` A Dose Titration Algorithm (DTA) to drive the titration

Value
A list with 2 components:
- `course` A data frame containing cycle-wise measures of each id’s titration course
- `anc.ts` A data frame detailing high-frequency ANC measures for each id

Author(s)
David C. Norris

References

Examples
```r
if(interactive()){
  # Reproduce Figure 5 from the F1000Research paper (run time > 10 s).
  # 1. Set up sim$pop & sim$pkpd by running the repro for Figures 1 & 3:
  example(topic="Onoue.Friberg", package="DTAT", ask=FALSE)
  # 2. Do the neutrophil-nadir-guided dose titration:
  chemo <- titrate(doserange = c(50, 3000),
                   dta=newton.raphson(dose1 = 50,
                                      omega = 0.75,
                                      slope1 = -2.0,
```
titration

Simulate a '3+3/PC' dose-titration trial

description

Simulate a '3+3/PC' dose-titration trial

usage

titration(x, periods, ...)

## S4 method for signature 'DE, numeric'
titration(x, periods, ...)

slopeU = -0.2)

library(latticeExtra)
newton <- chemo$course
new.ts <- chemo$anc.ts
anc.tics <- c(200, 500, 1500, 4000, 10000)
right <- xYplot(ANC ~ time | id, data=new.ts
  , as.table=TRUE, type="l"
  , layout=c(5, 5)
  , scales=list(y=list(log=TRUE, lim=c(100, 1.5e4)
      , at=anc.tics
      , lab=as.character(anc.tics)),
    x=list(at=seq(0, 30, 3)))
}
newton <- upData(newton
  , time = 3*(cycle-1)
  , labels = c(time="Time")
  , units = c(time="weeks")
  , print = FALSE)
dose.tics <- c(50, 200, 600, 1500, 3000)
left <- xYplot(scaled.dose ~ time | id, data=newton
  , as.table=TRUE, type="p", pch="+", cex=1.5
  , layout=c(5, 5)
  , scales=list(y=list(log=TRUE, lim=c(30, 3200))
      , at=DTAT::scaled(dose.tics)
      , lab=as.character(dose.tics)),
    x=list(lim=c(-1, 31)
      , at=seq(0, 30, 3)
      , lab=c(0", ", 6", ", 12", ", 18", ", 24", ", 30")

)
update(doubleYScale(left, right, add.ylab2=TRUE)
  , par.settings = simpleTheme(col=brewer.pal(4, "PRGn")(c(4, 1)))
)
Arguments

- **x**: An object of S4 class `de`

- **periods**: The number of DLT assessment periods to titrate over. Should be a positive integer.

- **...**: May be used to pass `verbatim = 'TRUE'` to internal `step_time` method.

References

Index

* datasets
  de.bioRxiv.240846, 4
dtat1000, 8
sim, 16
* survival
  dose.survfit, 5
dose.survival, 6
ds.curve, 7

as_d3_data,DE-method, 3

DE, 5, 19
DE-class, 3
de.bioRxiv.240846, 4
dose.survfit, 5, 6, 7
dose.survival, 6, 6
ds.curve, 7
DTAT (DTAT-package), 2
DTAT-package, 2
dtat1000, 8

km.ci, 5, 6

newton.raphson, 10

Onoue.Friberg, 11

plot,DE,missing-method, 13
pomp, 12

r2d3, 14
runDTATapp, 14

scaled, 15
seq.function, 15
sim, 12, 16
Surv, 6

titrate, 10, 17
titration, 18
titration,DE,numeric-method (titration), 18