Package: linpk (via r-universe)

September 13, 2024

Type Package

Version 1.1.4

Date 2024-04-16

Title Generate Concentration-Time Profiles from Linear PK Systems

Author Benjamin Rich [aut, cre]

Maintainer Benjamin Rich <mail@benjaminrich.net>

URL https://github.com/benjaminrich/linpk

BugReports https://github.com/benjaminrich/linpk/issues

Description Generate concentration-time profiles from linear pharmacokinetic (PK) systems, possibly with first-order absorption or zero-order infusion, possibly with one or more peripheral compartments, and possibly under steady-state conditions. Single or multiple doses may be specified. Secondary (derived) PK parameters (e.g. Cmax, Ctrough, AUC, Tmax, half-life, etc.) are computed.

License GPL-3

Imports graphics,utils,mvtnorm

Suggests knitr,rmarkdown,shiny

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

Repository https://benjaminrich.r-universe.dev

RemoteUrl https://github.com/benjaminrich/linpk

RemoteRef HEAD

RemoteSha 5bcfc4a44ccb748f37ac8a5390deb9f2116d47c8

Contents

`as.data.frame.pkprofile` *Coerce a* `pkprofile` *to a* `data.frame`

Description

Coerce a `pkprofile` to a `data.frame`

Usage

```
## S3 method for class 'pkprofile'
as.data.frame(x, ..., col.names = c("time", "conc"), .state = FALSE)
```

Arguments

`x`	An object of class `pkprofile`.
`...`	Further arguments passed along.
`col.names`	Character vector of length 2 giving the names for the time and concentration columns.
`.state`	Include the complete state along with `time` and `conc`?

Value

A `data.frame` with columns `time` and `conc` (or the names specified in `col.names`). If `.state == TRUE`, then the complete state is appended (as a matrix column).

blockdiag *Construct a block-diagonal matrix.*

Description

Construct a block-diagonal matrix.

Usage

```
blockdiag(..., .names = NULL, .colnames = .names, .rownames = .names)
```

Arguments

`...`	Any number of square matrices making up the diagonal blocks of the matrix.
`.names`, `.colnames`, `.rownames`	Optionally, specify the row and column names of the resulting matrix.

Value

A block-diagonal matrix.

Examples

```
blockdiag(matrix(1, 2, 2), 2, matrix(3, 4, 4))
blockdiag(c(a=5, b=6), Diag(c=7, d=8), LTmat(c(1, 2, 3), .names=c("e", "f")))
blockdiag(a=5, b=6, Diag(c=7, d=8), LTmat(c(1, 2, 3), .names=c("e", "f")))
blockdiag(5, 6, Diag(7, 8), LTmat(c(1, 2, 3)), .names=c("a", "b", "c", "d", "e", "f"))
```

cor2cov *Convert from standard deviation and correlation matrix to covariance matrix.*

Description

Convert from standard deviation and correlation matrix to covariance matrix.

Usage

```
cor2cov(cor, sd)
```

Arguments

`cor`	A correlation matrix. If `sd` is missing, the diagonal entries are taken to be the standard deviations, otherwise they are ignored.
`sd`	A vector of standard deviations (optional).

Value

A covariance matrix.

Examples

```
cor2cov(matrix(c(1, 0.5, 0.5, 1), 2, 2), 0.1)
cor2cov(LTmat(c(0.39, 0.67, 0.28), .names=c("CL", "VC")))
```

`Diag` *Diagonal Matrix*

Description

Like the base `diag` function, except that vectors of length one are converted to 1-by-1 matrices, values can be specified either as a single vector argument or multiple arguments, and row and column names can be specified.

Usage

```
Diag(x, ..., .names = names(x), .colnames = .names, .rownames = .names)
```

Arguments

`x`	A numeric (or number-like) vector (possibly named).
`...`	Additional numeric (or number-like) vectors (possibly named).
`.names`, `.colnames`, `.rownames`	Optionally, specify the row and column names of the resulting diagonal matrix.

Examples

```
Diag(6)
Diag(3.14, .names="pi")
Diag(1:6, .colnames=LETTERS[1:6], .rownames=letters[1:6])
Diag(1, 2, 3)
Diag(a=1, b=2, c=3)
Diag(a=1, c(b=2, c=3))
Diag(2+3i, 4+5i)
```

dose.frame *Get the doses from a PK profile.*

Description

Get the doses from a PK profile.

Usage

```
dose.frame(x)
```

Arguments

x	A object of class pkprofile.

Value

A data.frame containing the realized doses, one per row. The data.frame has all the columns described in pkprofile, except addl, since all additional doses have been expanded to individual rows. It also has a conc column with the simulated concentration at the time of the dose.

See Also

pkprofile

Examples

```
t.obs <- seq(0, 6*24, 0.5)
y <- pkprofile(t.obs, cl=0.5, vc=11, ka=1.3,
    dose=list(t.dose=c(0, 24*2 + 14), amt=c(100, 50), addl=c(4, 0), ii=24))
dose.frame(y)
```

finalstate *Get the final state or time of a PK profile.*

Description

Get the final state or time of a PK profile.

Usage

```
finalstate(x)

finaltime(x)
```

Arguments

x	A object of class pkprofile.

Value

A numeric vector containing the state of each compartment at the final observation time (finalstate), or the final observation time itself (finaltime).

See Also

- pkprofile for generating a PK profile.
- pkprofile.pkprofile for appending to an existing PK profile.

Examples

```
# Administer a dose at time 0 and a second dose using the final state
# from the first dose (at 12h) as the initial state for the second dose.
t.obs <- seq(0, 12, 0.1)
y <- pkprofile(t.obs, cl=0.25, vc=5, ka=1, dose=list(t.dose=0, amt=1))
finalstate(y)
y2 <- pkprofile(t.obs, cl=0.25, vc=5, ka=1, dose=list(t.dose=0, amt=1), initstate=finalstate(y))
plot(y, xlim=c(0, 24), ylim=c(0, max(y2)), col="blue")  # First dose
lines(t.obs+12, y2, col="red")                          # Second dose

# Add a vertical line to show where the first profile ends.
abline(v=finaltime(y), col="gray75", lty=2)
```

---

generateETA — *Generate individual random effects from a multivariate normal distribution.*

Description

Generate individual random effects from a multivariate normal distribution.

Usage

```
generateETA(n, omegaLT, omega = LTmat(omegaLT), eta.names = colnames(omega))
```

Arguments

n	The number of individuals.
omegaLT	A numeric vector giving the elements of the lower triangle of the covariance matrix by row.
omega	The covariance matrix.
eta.names	A character vector of names for each random effect (defaults to the column names of omega, or if NULL then to ETA1, ETA2, ...).

Value

An $n \times p$ matrix, where each row contains the vector of random effects for one individual (p is the size of the covariance matrix).

See Also

LTmat blockdiag

Examples

```
omegaLT <- c(0.123, 0.045, 0.678)
generateETA(10, omegaLT)
```

---

halflife *Half-lives of a linear PK system.*

Description

Half-lives of a linear PK system.

Usage

```
halflife(x)
```

Arguments

x	A object of class pkprofile.

Value

A numeric vector containing the half-lives for the different phases of the system. The number of phases generally equal the number of compartments, plus one for the absorption phase if the system has first order absorption (i.e. if ka is specified). The values are returned sorted in ascending order, so the first corresponds to the alpha phase, the second beta, the third gamma, and so on. The absorption half-life, if present, comes last (it can also be identified by comparing it to the value of log(2)/ka).

Examples

```
y <- pkprofile(0, cl=0.25, vc=5, ka=1.1)
halflife(y)
log(2)/1.1

y <- pkprofile(0, cl=0.25, vc=5, ka=0.01)  # Flip-flop kinetics
halflife(y)
log(2)/0.01

# Three-compartment model
```

```
y <- pkprofile(0, cl=2, vc=10, q=c(0.5, 0.3), vp=c(30, 40))
halflife(y)

# The terminal half-life can be used to obtain the terminal slope of the
# concentration-time curve on the semi-log scale:
t.obs <- seq(0, 36, 0.1)
y <- pkprofile(t.obs, cl=0.25, vc=5, ka=1, dose=list(t.dose=0, amt=1))
plot(log2(y))
abline(-2.247927, -1/halflife(y)[1], col=adjustcolor("blue", 0.2), lwd=12)
```

`linpkApp` *Runs the interactive shiny app.*

Description

Runs the interactive shiny app.

Usage

```
linpkApp(...)
```

Arguments

`...`	Arguments passed to `shiny::runApp()`.

Value

Called for its side effects.

Note

The app requires the following packages:

- ‘shiny‘
- ‘shinyjs‘
- ‘shinyAce‘
- ‘dygraphs‘

Make they are installed or the app won’t work.

Examples

```
## Not run:
linpkApp()

## End(Not run)
```

LTmat — *Construct a symmetric matrix from its lower triangle.*

Description

Construct a symmetric matrix from its lower triangle.

Usage

```
LTmat(
  LT,
  ...,
  .names = attr(LT, ".names"),
  .colnames = .names,
  .rownames = .names
)
```

Arguments

`LT`	A numeric (or number-like) vector giving the elements of the lower triangle of the matrix by row (see examples).
`...`	Additional numeric (or number-like) vectors, appended to `LT`.
`.names, .colnames, .rownames`	Optionally, specify the row and column names of the resulting diagonal matrix. The names can also be specified as the LHS of a 2-sided formula (see Examples).

Value

A symmetric matrix.

Examples

```
LTmat(1:6)
LTmat(1, 3, 5, .names=c("a", "b"))
LTmat(c(a, b) ~ c(1, 3, 5))  # Names can also be specified as LHS of a 2-sided formula
LTmat(1+2i, 3+4i, 5+6i)
```

pkprofile — *Generate a concentration-time profile.*

Description

This function generates concentration-time profiles from a linear pharmacokinetic (PK) system, possibly with first-order absorption or zero-order infusion, possibly with one or more peripheral compartments, and possibly under steady-state conditions. Single or multiple doses may be specified.

Usage

```
pkprofile(...)

## Default S3 method:
pkprofile(
  t.obs = seq(0, 24, 0.1),
  cl = 1,
  vc = 5,
  q = numeric(0),
  vp = numeric(0),
  ka = 0,
 dose = list(t.dose = 0, amt = 1, rate = 0, dur = 0, ii = 24, addl = 0, ss = 0, cmt = 0,
    lag = 0, f = 1),
  sc = vc,
  initstate = NULL,
  ...
)

## S3 method for class 'matrix'
pkprofile(
  A,
  t.obs = seq(0, 24, 0.1),
 dose = list(t.dose = 0, amt = 1, rate = 0, dur = 0, ii = 24, addl = 0, ss = 0, cmt = 0,
    lag = 0, f = 1),
  defdose = 1,
  sc = 1,
  initstate = NULL,
  ...
)
```

Arguments

...	Further arguments passed along to other methods.
t.obs	A numeric vector of times at which to observe concentrations.
cl	Central clearance parameter.
vc	Central volume parameter.
q	Inter-compartmental clearance. Can be a vector for more than one peripheral compartment, or empty for none. Must match vp in length.
vp	Peripheral volume. Can be a vector for more than one peripheral compartment, or empty for none. Must match q in length.
ka	First-order absorption rate parameter. Set to 0 to indicate that there is no first-order absorption (i.e. bolus or infusion).
dose	A list or data.frame containing dose information. May contain the following elements: t.dose Dose time (default 0). amt Dose amount (default 1).

`rate` Rate of zero-order infusion, or 0 to ignore (default 0). Only one of `rate` and `dur` should be specified unless `amt` is missing.

`dur` Duration of zero-order infusion, or 0 to ignore (default 0). Only one of `rate` and `dur` should be specified unless `amt` is missing.

`ii` Interdose interval (default 24). Only used if addl or ss are used.

`addl` Number of *additional* doses (default 0). The total number of doses given is `addl + 1`.

`ss` Indicates that a dose is given under steady-state conditions (default 0 or FALSE; converted to `logical` internally).

`cmt` The number of the compartment into which the dose is administered. The default value is 0, which indicates the depot compartment for first-order absorption (i.e. `ka > 0`), and central compartment otherwise.

`lag` Time lag (default 0).

`f` Bioavailable fraction (default 1).

`sc` A scaling constant for the central compartment. Concentrations are obtained by dividing amounts by this constant.

`initstate` A numeric vector containing values to initialize the compartments.

`A` A matrix of first-order rate constants between the compartments.

`defdose` The default dose compartment when the compartment is missing or 0.

Value

An object of class "pkprofile", which is simply a numeric vector of concentration values with some attributes attached to it. This object has its own methods for `print`, `plot`, `lines` and `points`.

Methods (by class)

- `pkprofile(default)`: Default method.
- `pkprofile(matrix)`: Matrix method.

Warning

Pay attention to the default arguments. They are there for convenience, but may lead to undesired results if one is not careful.

See Also

- `halflife`
- `secondary`
- `print.pkprofile`
- `plot.pkprofile`
- `lines.pkprofile`
- `points.pkprofile`

Examples

```
# Default values, a bolus injection
y <- pkprofile()
plot(y)

t.obs <- seq(0, 24, 0.1)
dur <- 1
amt <- 1
ka <- 1
cl <- 0.25
vc <- 5
q <- 2.5
vp <- 10

# One-compartment model with first-order absorption, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, ka=ka, dose=list(amt=amt))
plot(y)

# Two-compartment model with first-order absorption, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, vp=vp, q=q, ka=ka, dose=list(amt=amt))
plot(y)

# One-compartment model with zero-order infusion, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, dose=list(dur=dur, amt=amt))
plot(y)

# Two-compartment model with zero-order infusion, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, vp=vp, q=q, dose=list(dur=dur, amt=amt))
plot(y)

# Two-compartment model with bolus injection, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, vp=vp, q=q, dose=list(amt=amt))
plot(y)

# Two-compartment model with bolus injection into the peripheral compartment, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, vp=vp, q=q, dose=list(amt=amt, cmt=2))
plot(y)

# Two-compartment model with zero-order infusion into the peripheral compartment, single dose
y <- pkprofile(t.obs, cl=cl, vc=vc, vp=vp, q=q, dose=list(amt=amt, cmt=2, dur=dur))
plot(y)

t.obs <- seq(0, 24*6, 1)

# One-compartment model with first-order absorption, multiple doses
y <- pkprofile(t.obs, cl=cl, vc=vc, ka=ka, dose=list(t.dose=seq(0, 24*5, 12), amt=amt))
plot(y)

# One-compartment model with first-order absorption, multiple doses specified by addl and ii
y <- pkprofile(t.obs, cl=cl, vc=vc, ka=ka, dose=list(t.dose=0, amt=amt, addl=9, ii=12))
plot(y, type="b")
points(y, col="blue")
```

```
# One-compartment model with first-order absorption, multiple doses under steady-state conditions
yss <- pkprofile(t.obs, cl=cl, vc=vc, ka=ka, dose=list(t.dose=0, amt=amt, addl=9, ii=12, ss=1))
lines(yss, col="red")
points(yss, col="green")

# One-compartment model with zero-order infusion, multiple doses specified by addl and ii
y <- pkprofile(t.obs, cl=cl, vc=vc, dose=list(dur=dur, amt=amt, addl=9, ii=12))
plot(y, log="y")

# One-compartment model with zero-order infusion, multiple doses  under steady-state conditions
yss <- pkprofile(t.obs, cl=cl, vc=vc, dose=list(dur=dur, amt=amt, addl=9, ii=12, ss=1))
lines(yss, col="red")
```

`pkprofile.pkprofile` *Continue an existing concentration-time profile.*

Description

This method can be used to append to an existing PK profile, for instance to simulate a PK profile with parameters that change over time. Each time the parameters change, a new call to this method is used to advance the system with the new parameter values.

Usage

```
## S3 method for class 'pkprofile'
pkprofile(obj, t.obs = finaltime(obj) + seq(0, 24, 0.1), ..., append = TRUE)
```

Arguments

`obj`	An object returned from a previous call to `pkprofile`.
`t.obs`	A numeric vector of times at which to observe concentrations.
`...`	Further arguments passed along.
`append`	Should the new profile be appended to the current samples? Otherwise, only the new samples are returned.

Value

An object of class "pkprofile".

Warning

The new parameters take effect at the time when the previous profile ends. If the previous profile ends before the new sampling starts, the *new* parameters will be used to advance the system to the start of the new sampling.

Any ongoing zero-order infusion at the end of the previous profile is dropped. The remaining infusion amount will NOT be carried forward.

See Also

pkprofile

Examples

```
t.obs <- seq(0, 24, 0.1)
amt <- 1
ka <- 1
cl <- 0.25
vc <- 5

# One-compartment model with first-order absorption
# First dose at time 0
y <- pkprofile(t.obs, cl=cl, vc=vc, ka=ka, dose=list(t.dose=0, amt=amt))

# Second dose at 24h with a lower clearance
y <- pkprofile(y, t.obs+24, cl=0.5*cl, vc=vc, ka=ka, dose=list(t.dose=24, amt=amt))

# Third dose at 48h with a higher clearance
y <- pkprofile(y, t.obs+48, cl=2*cl, vc=vc, ka=ka, dose=list(t.dose=48, amt=amt))
plot(y)
```

---

secondary *Derive secondary PK parameters.*

Description

Derive secondary PK parameters.

Usage

```
secondary(x, From = NULL, To = NULL, include.dose.times = T)
```

Arguments

x	A object of class pkprofile.
From	A vector of interval start times. The defaults is the times of the doses.
To	A vector of interval end times. The defaults is the time of the next dose, or last observation time.
include.dose.times	Should dose times (and end of infusion times) be considered in addition to the simulation times?

Value

A `data.frame` with one row for each time interval and with the following columns:

`From` The time of the start of the interval. Can differ from the specified start time because it always corresponds to an actual data point.

`To` The time of the end of the interval. Can differ from the specified end time because it always corresponds to an actual data point.

`N` The number of distinct data points in the interval used to derive `AUC`, `Cmax`, etc.

`Ctrough` Concentration at the time of dose (i.e. just prior to the dose). Only present if the start of the interval corresponds to a dose time.

`Cmin` Minimum concentration over the interval.

`Tmin` Time of the minimum concentration over the interval.

`Cmax` Maximum concentration over the interval.

`Tmax` Time of the maximum concentration over the interval.

`Cave` Average concentration over the interval (calculated by the trapezoid rule).

`AUC` Area under the concentration-time curve over the interval (calculated by the trapezoid rule).

Examples

```
t.obs <- seq(0, 24*4, 0.1)
y <- pkprofile(t.obs, cl=0.25, vc=5, ka=1, dose=list(t.dose=0, amt=1, addl=6, ii=12))
secondary(y)
secondary(y, 0, 48)
secondary(y, 0, Inf)
sum(secondary(y)$AUC)  # Same as above
plot(y)
with(secondary(y), points(Tmax, Cmax, pch=19, col="blue"))
with(secondary(y), points(Tmin, Cmin, pch=19, col="red"))
with(secondary(y), points(From, Ctrough, pch=19, col="green"))
with(secondary(y), points(From + 6, Cave, pch=19, col="purple", cex=2))
```

Index