

Package ‘nlmixr2lib’

October 22, 2025

Type Package

Version 0.3.1

Title A Model Library for 'nlmixr2'

Description A model library for 'nlmixr2'. The models include (and plan to include) pharmacokinetic, pharmacodynamic, and disease models used in pharmacometrics. Where applicable, references for each model are included in the meta-data for each individual model. The package also includes model composition and modification functions to make model updates easier.

Depends R (>= 4.1)

Imports checkmate, cli, methods, nlmixr2est, rxode2 (>= 2.0.12)

License GPL (>= 2)

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

URL <https://github.com/nlmixr2/nlmixr2lib>,
<https://nlmixr2.github.io/nlmixr2lib/>

Encoding UTF-8

Language en-US

Suggests covr, dplyr, knitr, rmarkdown, testthat (>= 3.0.0), qs,
tidyr, ggplot2, withr

Config/testthat/edition 3

NeedsCompilation no

Author Richard Hooijmaijers [aut],
Matthew Fidler [aut] (ORCID: <<https://orcid.org/0000-0001-8538-6691>>),
Bill Denney [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5759-428X>>)

Maintainer Bill Denney <wdenney@humanpredictions.com>

Repository CRAN

Date/Publication 2025-10-22 14:00:02 UTC

Contents

addBaseline1exp	2
addBaselineConst	3
addBaselineExp	4
addBaselineLin	5
addCmtProp	6
addDepot	7
addDirectLin	8
addDirToModelDb	9
addEffectCmtLin	9
addEta	10
addIndirect	11
addIndirectLin	13
addLogEstimates	14
addResErr	15
addTransit	16
addWeibullAbs	17
combinePaste2	18
convertEmax	19
convertEmaxHill	20
convertKinR0	21
convertLogLin	22
convertMM	23
convertQuad	24
defaultCombine	25
fakeCc	26
modeldb	27
modellib	28
pkTrans	28
readModelDb	31
removeDepot	32
removeLinesAndInis	32
removeTransit	33
searchReplace	34
setCombineType	35
updateOncologyXenograftSimeoni2004	35

Index	37
--------------	-----------

addBaseline1exp	<i>Add baseline that decays exponential with time</i>
-----------------	---

Description

Add baseline that decays exponential with time

Usage

```
addBaseline1exp(ui, effect = "effect", eb = "Eb", time = "time", kb = "kb")
```

Arguments

ui	rxode2 model
effect	the effect variable that will be modeled
eb	baseline constant parameter
time	the time or other variable used for baseline decay
kb	the first order baseline decay constant

Value

model with baseline constant

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertQuad() |>
  addBaseline1exp()
```

addBaselineConst	<i>Add an estimated baseline constant</i>
------------------	---

Description

Add an estimated baseline constant

Usage

```
addBaselineConst(ui, effect = "effect", eb = "Eb")
```

Arguments

ui	rxode2 model
effect	the effect variable that will be modeled
eb	baseline constant parameter

Value

model with baseline constant

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaseline1exp\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertQuad() |>
  addBaselineConst()
```

addBaselineExp

Add baseline that decays exponential with time

Description

Add baseline that decays exponential with time

Usage

```
addBaselineExp(ui, effect = "effect", eb = "Eb", time = "time", kb = "kb")
```

Arguments

ui	rxode2 model
effect	the effect variable that will be modeled
eb	baseline constant parameter
time	the time or other variable used for baseline decay
kb	the first order baseline decay constant

Value

model with baseline constant

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaselineExp\(\)](#), [addBaselineConst\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertQuad() |>
  addBaselineExp()
```

addBaselineLin	<i>Add an estimated baseline linear constant</i>
----------------	--

Description

Add an estimated baseline linear constant

Usage

```
addBaselineLin(ui, effect = "effect", eb = "Eb", time = "time")
```

Arguments

ui	rxode2 model
effect	the effect variable that will be modeled
eb	baseline constant parameter
time	the time or other variable used for baseline decay

Value

model with baseline linear constant

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaselineExp\(\)](#), [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertQuad() |>
  addBaselineLin()
```

addCmtProp	<i>Add a property to a compartment</i>
------------	--

Description

Add a property to a compartment

Usage

```
addCmtProp(ui, prop = c("f", "lag", "dur", "rate", "ini"), cmt)

addBioavailability(ui, cmt)

addLag(ui, cmt)

addDur(ui, cmt)

addRate(ui, cmt)

addIni(ui, cmt)
```

Arguments

ui	rxode2 ui object
prop	property to add to a compartment: - F: bioavailability - lag: absorption lag time - dur: modeled duration of infusion - rate: modeled infusion rate - ini: initial value of the compartment
cmt	compartment to apply the property to

Value

rxode2 ui object with property applied

Functions

- `addBioavailability()`: Adds the bioavailability to a compartment in the model
- `addLag()`: Adds the lag-time to a compartment in the model
- `addDur()`: Adds the modeled duration to a compartment in the model
- `addRate()`: Adds the modeled rate to a compartment in the model
- `addIni()`: Adds the initial value to the compartment

Author(s)

Matthew L. Fidler

Examples

```
readModelDb("PK_3cmt_des") |> addCmtProp("f", "depot")

readModelDb("PK_3cmt_des") |> addBioavailability(depota)

readModelDb("PK_3cmt_des") |> addLag(depota)

readModelDb("PK_3cmt_des") |> addDur(depota)

readModelDb("PK_3cmt_des") |> addRate(depota)

readModelDb("PK_3cmt_des") |> addIni(depota)
```

addDepot

To convert from infusion/intravenous administration to first-order oral absorption

Description

To convert from infusion/intravenous administration to first-order oral absorption

Usage

```
addDepot(ui, central = "central", depot = "depot", ka = "ka")
```

Arguments

ui	The model as a function (or something convertible to an rxUi object)
central	central compartment name
depot	depot compartment name
ka	absorption rate parameter name

Value

a model with the depot added

Examples

```
# most of the examples in the model library already have a depot
# the PK_2cmt_no_depot is an exception
readModelDb("PK_2cmt_no_depot") |> addDepot()
```

addDirectLin	<i>Add direct linear effect with baseline=0</i>
--------------	---

Description

Add direct linear effect with baseline=0

Usage

```
addDirectLin(ui, ek = "Ek", cc = c("Ce", "Cc"), effect = "effect")
```

Arguments

ui	rxode2 model
ek	simulation linear constant
cc	the concentration value
effect	the effect variable that will be modeled

Value

model with direct linear effect added (baseline=0)

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaseline1exp\(\)](#), [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
# Direct linear model
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin()

# Direct emax model
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertEmax()
```

addDirToModelDb	<i>Add a directory to the modeldb</i>
-----------------	---------------------------------------

Description

Add a directory to the modeldb

Usage

```
addDirToModelDb(dir, modeldb = data.frame())

addFileToModelDb(dir, file, modeldb)
```

Arguments

dir	Directory name containing model files
modeldb	The starting modeldb data.frame
file	The file name (without the directory name)

Value

The updated modeldb data.frame
the model database

Functions

- addFileToModelDb(): Add a file to the modeldb

addEffectCmtLin	<i>Add effect compartment</i>
-----------------	-------------------------------

Description

Add effect compartment

Usage

```
addEffectCmtLin(
  ui,
  ke0 = "ke0",
  cc = "Cc",
  ce = "Ce",
  ek = "Ek",
  effect = "effect"
)
```

Arguments

ui	rxode2 model
ke0	This is the effect compartment keo rate
cc	the concentration value
ce	This is the concentration in the effect compartment
ek	simulation linear constant
effect	the effect variable that will be modeled

Value

a model with an effect compartment attached

Author(s)

Matthew L. Fidler

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addEffectCmtLin()

# Can also be changed to the more typical Emax with constant (estimated) baselie
readModelDb("PK_2cmt_no_depot") |>
  addEffectCmtLin() |>
  convertEmaxHill() |>
  addBaselineConst()
```

addEta	<i>Add random effects to a model</i>
--------	--------------------------------------

Description

Add random effects to a model

Usage

```
addEta(
  ui,
  eta,
  priorName = getOption("nlmixr2lib.priorEta", TRUE),
  etaCombineType = c("default", "snake", "camel", "dot", "blank")
)
```

Arguments

ui	The model as a function
eta	vector with the parameters to add random effects (sometimes referred to as inter-individual variability, IIV) on
priorName	logical, if TRUE, the parameter name specified in 'eta' will be used to add the eta value prior name is used instead of the left handed side of the equation.
etaCombineType	the option for the how to combine the eta with the parameter name. Can be: "default", "snake", "camel", "dot", "blank"

Value

The model with eta added to the requested parameters

Author(s)

Bill Denney, Richard Hooijmaijers & Matthew L. Fidler

Examples

```
library(rxode2)
readModelDb("PK_1cmt") |> addEta("ka")
```

addIndirect	<i>Add an indirect response model to a PK model</i>
-------------	---

Description

Add an indirect response model to a PK model

Usage

```
addIndirect(
  ui,
  stim = c("in", "out"),
  inhib = c("in", "out"),
  hill = FALSE,
  ek = "Ek",
  ik = "Ik",
  emax = "Emax",
  ec50 = "EC50",
  imax = "Imax",
  ic50 = "IC50",
  kin = "kin",
  kout = "kout",
  g = "g",
  cc = "Cc",
```

```

    R = "R",
    effect = "effect"
  )

```

Arguments

ui	rxode2 model
stim	what type of stimulation indirect response model: - 'in': stimulation of input - 'out': stimulation of output
inhib	what type of inhibition indirect response model: - 'in': inhibition of input - 'out': inhibition of output
hill	Boolean stating if a hill sigmoid coefficient will be added
ek	simulation linear constant
ik	inhibition linear constant
emax	Emax parameter
ec50	EC50 parameter
imax	maximum inhibitory concentration
ic50	concentration where half of the I _{max} occurs
kin	this is the kin parameter name
kout	this is the kout parameter name
g	hill coefficient
cc	the concentration value
R	drug response compartment
effect	the effect variable that will be modeled

Value

pk model with indirect response model added

Author(s)

Matthew L. Fidler

See Also

Other Indirect response: [addIndirectLin\(\)](#), [convertKinR0\(\)](#)

Examples

```

readModelDb("PK_2cmt_no_depot") |>
  addIndirect(stim="in",hill=TRUE)

readModelDb("PK_2cmt_no_depot") |>
  addIndirect(inhib="out", imax=1)

```

addIndirectLin	<i>Add linear indirect response model</i>
----------------	---

Description

Add linear indirect response model

Usage

```
addIndirectLin(  
  ui,  
  stim = c("in", "out"),  
  inhib = c("in", "out"),  
  ek = "Ek",  
  ik = "Ik",  
  kin = "kin",  
  kout = "kout",  
  cc = "Cc",  
  R = "R",  
  effect = "effect"  
)
```

Arguments

ui	rxode2 model
stim	what type of stimulation indirect response model: - 'in': stimulation of input - 'out': stimulation of output
inhib	what type of inhibition indirect response model: - 'in': inhibition of input - 'out': inhibition of output
ek	simulation linear constant
ik	inhibition linear constant
kin	this is the kin parameter name
kout	this is the kout parameter name
cc	the concentration value
R	drug response compartment
effect	the effect variable that will be modeled

Value

model with linear indirect effect added

Note that while linear indirect effects are not common, it allows an easier hook to produce other standard effect curves like Emax/Imax, Hill, etc.

Author(s)

Matthew L. Fidler

See AlsoOther Indirect response: [addIndirect\(\)](#), [convertKinR0\(\)](#)**Examples**

```
readModelDb("PK_2cmt_no_depot") |> addIndirectLin(stim="in")
readModelDb("PK_2cmt_no_depot") |> addIndirectLin(stim="out")
readModelDb("PK_2cmt_no_depot") |> addIndirectLin(inhib="in")
readModelDb("PK_2cmt_no_depot") |> addIndirectLin(inhib="out")
```

addLogEstimates	<i>Add log estimates to a model</i>
-----------------	-------------------------------------

Description

Add log estimates to a model

Usage

```
addLogEstimates(ui, vars, extraLines = NULL, beforeCmt = NULL)
```

Arguments

ui	rxode2 model
vars	estimates to add they will be parameterized as: var <- exp(lvar) where var is the variable name in the model and lvar is the log transformed variable that will be estimated
extraLines	this is a list of additional lines to add to the model just after the variables are defined. It must be NULL or a list of language objects.
beforeCmt	if the model is compartmental you can specify the preferred names where the estimates and extra lines are added before

Value

rxode2 model with log estimates added (and possibly extra lines)

Author(s)

Matthew L. Fidler

Examples

```
# Change the transformation of the PK model from cl to k

readModelDb("PK_3cmt_des") |>
  removeLinesAndInis(c("kel", "k12", "k21", "k13", "k31", "vc")) |>
  addLogEstimates(c("kel", "k12", "k21", "k13", "k31", "vc"))

# You can also label the parameters by using a named character
# vector with the names of the parameters representing the
# variables and the values representing the labels:

readModelDb("PK_3cmt_des") |>
  removeLinesAndInis(c("kel", "k12", "k21", "k13", "k31", "vc")) |>
  addLogEstimates(c(kel="elimination", k12="k12 constant",
                    k21="k21 constant",
                    k13="k13 constant",
                    k31="k31 constant",
                    vc="volume of central compartment"))
```

addResErr

*Add residual error to a model***Description**

Add residual error to a model

Usage

```
addResErr(ui, reserr, endpoint)
```

Arguments

ui	The model as a function
reserr	The type or types of residual error (currently "addSd", "propSd", and "lnormSd" are accepted)
endpoint	the endpoint to apply the error; will default to the first error in the model

Details

For reserr, the parameter will be named with the dependent variable from the model as a prefix. For example, if the dependent variable in the model is Cc, the parameter name for propSd will become CcpropSd.

Value

The model with residual error modified

Examples

```
library(rxode2)
readModelDb("PK_1cmt") |> addResErr("addSd")
readModelDb("PK_1cmt") |> addResErr("lnormSd")
readModelDb("PK_1cmt") |> addResErr(c("addSd", "propSd"))
```

addTransit

To add transit compartments to the model

Description

To add transit compartments to the model

Usage

```
addTransit(
  ui,
  ntransit,
  central = "central",
  depot = "depot",
  transit = "transit",
  ktr = "ktr",
  ka = "ka"
)
```

Arguments

ui	The model as a function
ntransit	the number of transit compartments to be added
central	central compartment name
depot	depot compartment name
transit	the transit compartment prefix
ktr	the parameter name for the transit compartment rate
ka	absorption rate parameter name

Value

a model with transit compartment added

This matches

```
‘dose->a0->a1->abs cmt->central’
```

But ‘a0’ is depot so dosing records labeled depot do not need to be changed

The abs cmt becomes the last "transit" compartment

This is simply for convenience

See Also

Other absorption: [addWeibullAbs\(\)](#), [removeTransit\(\)](#)

Examples

```
readModelDb("PK_1cmt_des") |> addTransit(3)
```

addWeibullAbs	<i>Converts first order absorption model to Weibull absorption model</i>
---------------	--

Description

Converts first order absorption model to Weibull absorption model

Usage

```
addWeibullAbs(  
  ui,  
  ntransit,  
  central = "central",  
  depot = "depot",  
  transit = "transit",  
  wa = "wa",  
  wb = "wb",  
  ka = "ka",  
  ktr = "ktr"  
)
```

Arguments

ui	The model as a function
ntransit	the number of transit compartments to be added
central	central compartment name
depot	depot compartment name
transit	the transit compartment prefix
wa	Weibull alpha parameter name
wb	Weibull beta parameter name
ka	absorption rate parameter name
ktr	the parameter name for the transit compartment rate

Value

model where first order absorption is changed to Weibull absorption model

Author(s)

Matthew L. Fidler

See Also

Other absorption: [addTransit\(\)](#), [removeTransit\(\)](#)

Examples

```
readModelDb("PK_1cmt_des") |>
  addWeibullAbs()
```

combinePaste2

Combine two strings using a naming convention

Description

Combine two in a manner similar to ‘paste()’ strings using the default combine type

Usage

```
combinePaste2(
  a,
  b,
  combineType = c("default", "snake", "camel", "dot", "blank")
)
```

Arguments

a	first string to combine
b	second string to combine
combineType	is the type of combination; can be: - "default": default combine (set with ‘defaultCombine()’) - "camel": camelCase combine - "snake": snake_case combine - "dot": dot combine (i.e. "a.b") - "blank": no separator (i.e. "ab")

Value

Combined strings separated with ‘defaultCombine()’

Author(s)

Matthew L. Fidler

Examples

```

combinePaste2("f", "depot")

combinePaste2("f", "depot", "snake")

combinePaste2("f", "depot", "dot")

combinePaste2("f", "depot", "blank")

```

convertEmax	<i>Convert linear effect to Emax effect</i>
-------------	---

Description

Convert linear effect to Emax effect

Usage

```

convertEmax(
  ui,
  emax = "Emax",
  ec50 = "EC50",
  imax = "Imax",
  ic50 = "IC50",
  ek = c("Ik", "Ek"),
  cc = c("Ec", "Cc")
)

```

Arguments

ui	rxode2 model
emax	Emax parameter
ec50	EC50 parameter
imax	Imax parameter used when input model contains "Ik" instead of "Ek"
ic50	IC50 parameter used when input model contains "Ik" instead of "Ek"
ek	simulation linear constant
cc	the concentration value

Value

Model with the linear effect converted to an Emax effect

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaseline1exp\(\)](#), [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertLogLin\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="in") |>
  convertEmax()

# When emax=1
readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="in") |>
  convertEmax(emax=1)
```

convertEmaxHill	<i>Convert linear effect to Emax-Hill effect</i>
-----------------	--

Description

Convert linear effect to Emax-Hill effect

Usage

```
convertEmaxHill(
  ui,
  emax = "Emax",
  ec50 = "EC50",
  g = "g",
  imax = "Imax",
  ic50 = "IC50",
  ek = c("Ik", "Ek"),
  cc = c("Ec", "Cc")
)
```

Arguments

ui	rxode2 model
emax	Emax parameter
ec50	EC50 parameter
g	hill coefficient
imax	Imax parameter used when input model contains "Ik" instead of "Ek"
ic50	IC50 parameter used when input model contains "Ik" instead of "Ek"
ek	simulation linear constant
cc	the concentration value

Value

Model with the linear effect converted to an Emax effect

Author(s)

Matthew L. Fidler

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="in") |>
  convertEmaxHill()
```

can also specify as emax=1

```
readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="in") |>
  convertEmaxHill(emax=1)
```

 convertKinR0

 Convert a kin/kout indirect response model to R0 and kout

Description

This replaces the kin/kout parameterization to the R0 and kout parameterization

Usage

```
convertKinR0(ui, kin = "kin", kout = "kout", R = "R", R0 = "R0")
```

Arguments

ui	a rxode2 user function
kin	the kin variable (by default is "kin")
kout	the kout variable (by default is "kout")
R	the compartment variable (by default is "R")
R0	the R0 variable (by default is "R0")

Value

a model where the estimated kin is changed to the estimated R0

Author(s)

Matthew L. Fidler

See Also

Other Indirect response: [addIndirect\(\)](#), [addIndirectLin\(\)](#)

Examples

```
addIndirect(stim="in") |> convertKinR0()
```

convertLogLin	<i>Converts a linear effect to a log-linear effect</i>
---------------	--

Description

Converts a linear effect to a log-linear effect

Usage

```
convertLogLin(ui, ek = c("Ik", "Ek"), cc = c("Ce", "Cc"))
```

Arguments

ui	rxode2 model
ek	simulation linear constant
cc	the concentration value

Value

model converted from linear to log-linear effect

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaseline1exp\(\)](#), [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertQuad\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertLogLin()

readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="out") |>
  convertLogLin()
```

convertMM	<i>Convert models from linear elimination to Michaelis-Menten elimination</i>
-----------	---

Description

Convert models from linear elimination to Michaelis-Menten elimination

Usage

```
convertMM(  
  ui,  
  central = "central",  
  elimination = "kel",  
  vm = "vm",  
  km = "km",  
  vc = "vc"  
)
```

Arguments

ui	model to convert
central	the central compartment where the elimination is present
elimination	variable for the elimination constant in the model
vm	variable name for Vmax in the model
km	variable name for Km in the model
vc	variable name for Vc in the model

Value

new model changing linear elimination to Michaelis-Menten elimination

Author(s)

Matthew L. Fidler

Examples

```
readModelDb("PK_1cmt_des") |> convertMM()  
  
readModelDb("PK_2cmt_des") |> convertMM()  
  
readModelDb("PK_3cmt_des") |> convertMM()  
  
readModelDb("PK_3cmt_des") |> removeDepot() |> convertMM()
```

`convertQuad`*Convert linear elimination to quadratic elimination*

Description

Convert linear elimination to quadratic elimination

Usage

```
convertQuad(ui, ek = c("Ik", "Ek"), cc = c("Ce", "Cc"), ek2 = "Ek2")
```

Arguments

<code>ui</code>	rxode2 model
<code>ek</code>	simulation linear constant
<code>cc</code>	the concentration value
<code>ek2</code>	quadratic coefficient

Value

model with linear effect converted to quadratic effect

Author(s)

Matthew L. Fidler

See Also

Other PD: [addBaseline1exp\(\)](#), [addBaselineConst\(\)](#), [addBaselineExp\(\)](#), [addBaselineLin\(\)](#), [addDirectLin\(\)](#), [convertEmax\(\)](#), [convertLogLin\(\)](#)

Examples

```
readModelDb("PK_2cmt_no_depot") |>
  addIndirectLin(stim="out") |>
  convertQuad()

readModelDb("PK_2cmt_no_depot") |>
  addDirectLin() |>
  convertQuad()

readModelDb("PK_2cmt_no_depot") |>
  addEffectCmtLin() |>
  convertQuad()
```

defaultCombine	<i>Default combine strings</i>
----------------	--------------------------------

Description

Default combine strings

Usage

```
defaultCombine(...)
```

```
snakeCombine(...)
```

```
camelCombine(...)
```

```
dotCombine(...)
```

```
blankCombine(...)
```

Arguments

... uses default to combine strings

Value

combined strings

Functions

- snakeCombine(): use snake_case to combine 2 strings
- camelCombine(): use camelCase to combine strings
- dotCombine(): use the default method for combining two strings
- blankCombine(): combine using a blank separator

Author(s)

Matthew L. Fidler

Examples

```
# default combine

defaultCombine("f", "depot")

defaultCombine(list(c("a", "funny", "c"))))

defaultCombine(c("a", "funny", "c"))
```

```

# snake combine

snakeCombine("f", "depot")

snakeCombine(list(c("a", "funny", "c")))

snakeCombine(c("a", "funny", "c"))

# dot combine

dotCombine("f", "depot")

dotCombine(list(c("a", "funny", "c")))

dotCombine(c("a", "funny", "c"))

# blank combine

blankCombine("f", "depot")

blankCombine(list(c("a", "funny", "c")))

blankCombine(c("a", "funny", "c"))

```

fakeCc

Fake blank Cc for creating PD only models

Description

Fake blank Cc for creating PD only models

Usage

```
fakeCc(fun, ..., cc = "Cc")
```

Arguments

fun	function to that requires Cc
...	arguments sent PD function
cc	character name of the concentration in the central compartment that will be faked to allow models that require Cc to change to models with Cc as a covariate

Value

Model where Cc is a covariate

Author(s)

Matthew L. Fidler

Examples

```
fakeCc(addDirectLin) |> convertEmaxHill()
```

modeldb	<i>Model library for nlmixr2</i>
---------	----------------------------------

Description

This is a data frame of the available models in nlmixr2lib, it is generated with the package. Custom modeldb may be used.

Usage

```
modeldb
```

Format

A data frame with 55 rows and 9 columns

name Model name that can be used to extract the model from the model library

description Model description in free from text; in model itself

parameters A comma separated string listing either the parameter in the model defined by population/individual effects or a population effect parameter

DV The definition of the dependent variable(s)

linCmt Logical flag indicating if solved models are used (TRUE) or not (FALSE)

algebraic Logical flag indicating if the model is purely algebraic: TRUE no linCmt() and no ODEs; FALSE otherwise

dosing A comma separated string of identified dosing compartments

depends A comma separated string of objects the model depends on

filename Filename of the model. By default these are installed in the model library and read on demand

modellib

Get the model from the model library

Description

This function gets a model from the available model library

Usage

```
modellib(name = NULL, eta = NULL, reserr = NULL)
```

Arguments

name	character with the name of the model to load (if NULL, lists all available base models)
eta	vector with the parameters to add random effects (sometimes referred to as inter-individual variability, IIV) on
reserr	The type or types of residual error (currently "addSd", "propSd", and "lnormSd" are accepted)

Details

This is a very first draft just to look at the proof of concept

Value

The function returns a function the model code (or NULL if the model = NULL)

Examples

```
modellib(name = "PK_1cmt")
modellib(name = "PK_1cmt", eta = c("ka", "vc"), reserr = "addSd")
modellib(name = "PK_1cmt", reserr = "addSd")
```

pkTrans

Change the transformation type for PK models

Description

Change the transformation type for PK models

Usage

```
pkTrans(
  ui,
  type = c("k", "k21", "vss", "aob", "alpha"),
  k13 = "k13",
  k31 = "k31",
  k12 = "k12",
  k21 = "k21",
  kel = "kel",
  vc = "vc",
  cl = "cl",
  vp = "vp",
  q = "q",
  vp2 = "vp2",
  q2 = "q2",
  vss = "vss",
  aob = "aob",
  alpha = "alpha",
  beta = "beta",
  gam = "gam",
  A = "A",
  B = "B",
  C = "C",
  s = "s",
  p = "p",
  tmp = "tmp",
  beforeCmt = c("depot", "central")
)
```

Arguments

ui	A model in terms of Clearance
type	the type of PK transformation to make: - "k": Change to rate constants (kel, k12, k21, k13, k31) - "vss": Change to volume of distribution at steady state (cl, vc, q, vss) - "aob": Change to A/B ratio (aob, alpha, beta, vc) - "k21": Change to k21 constant (k21, alpha, beta, vc) or (k21, k31, alpha, beta, gam, vc) - "alpha": Change to macro constants (alpha, beta, gam, A, B, C, vc)
k13	name of rate constant from central to periph2
k31	name of rate constant from periph2 to central
k12	name of rate constant from central to periph1
k21	name of rate constant from periph1 to central
kel	name of elimination rate constant
vc	name of central compartment volume

c1	name of clearance
vp	name of volume of periph1
q	name of intercompartmental clearance between central and periph1
vp2	name of volume of periph2
q2	name of intercompartmental clearance between central and periph2
vss	name of volume of distribution at steady state
aob	A/B ratio
alpha	macro constant name for first exponential decay term
beta	macro constant name for second exponential decay term
gam	macro constant name for third exponential decay term
A	macro coefficient for the first exponential decay term (corresponds with alpha)
B	macro coefficient for the second exponential decay term (corresponds with beta)
C	macro coefficient for the third exponential decay term (corresponds with gam)
s	sum constant name for the k12 three compartment
p	product constant name for the k12 three compartment
tmp	name of temporary variable for the three compartment with 'A', 'B', 'C', 'alpha', 'beta' and 'gam'.
beforeCmt	if the model is compartmental you can specify the preferred names where the estimates and extra lines are added before

Value

ui with no PK parameters estimated

Author(s)

Matthew L. Fidler

Examples

```
# Three compartment model translations

readModelDb("PK_3cmt_des") |>
  pkTrans("k")

readModelDb("PK_3cmt_des") |>
  pkTrans("k21")

readModelDb("PK_3cmt_des") |>
  pkTrans("alpha")

# The most types of transformations are
# available for 2 compartment models
```

```
readModelDb("PK_2cmt_des") |>
  pkTrans("k")

readModelDb("PK_2cmt_des") |>
  pkTrans("vss")

readModelDb("PK_2cmt_des") |>
  pkTrans("aob")

readModelDb("PK_2cmt_des") |>
  pkTrans("k21")

readModelDb("PK_2cmt_des") |>
  pkTrans("alpha")

# One compartment transformations are also available:

readModelDb("PK_1cmt_des") |>
  pkTrans("k")

readModelDb("PK_1cmt_des") |>
  pkTrans("alpha")

# also works without depot:

readModelDb("PK_3cmt_des") |>
  removeDepot() |>
  pkTrans("k")
```

readModelDb	<i>Read a model from the nlmixr2 model database</i>
-------------	---

Description

Read a model from the nlmixr2 model database

Usage

```
readModelDb(name)
```

Arguments

name	The name of the model (must be one of modeldb\$name)
------	--

Value

The model as a function

Examples

```
readModelDb("PK_1cmt")
```

removeDepot	<i>To convert from first order oral absorption to IV/Intravenous</i>
-------------	--

Description

To convert from first order oral absorption to IV/Intravenous

Usage

```
removeDepot(ui, central = "central", depot = "depot", ka = "ka")
```

Arguments

ui	The model as a function (or something convertible to an rxUi object)
central	central compartment name
depot	depot compartment name
ka	absorption rate parameter name

Value

Returns a model with the depot from a first order absorption model removed

Examples

```
readModelDb("PK_1cmt_des") |> removeDepot()
```

removeLinesAndInis	<i>Removes lines and inis from a model</i>
--------------------	--

Description

Removes lines and inis from a model

Usage

```
removeLinesAndInis(ui, vars)
```

Arguments

ui	A rxode2 model
vars	A character vector of variables to remove

Value

model with rxode2 lines and any estimates associate with lines removed

Author(s)

Matthew L. Fidler

Examples

```
readModelDb("PK_3cmt_des") |> removeLinesAndInis(c("kel", "k12", "k21"))
```

removeTransit	<i>To remove transit compartments from the model</i>
---------------	--

Description

To remove transit compartments from the model

Usage

```
removeTransit(  
  ui,  
  ntransit,  
  central = "central",  
  depot = "depot",  
  transit = "transit",  
  ktr = "ktr",  
  ka = "ka"  
)
```

Arguments

ui	The model as a function
ntransit	the number of transit compartments to be added
central	central compartment name
depot	depot compartment name
transit	The number of transit compartments to remove
ktr	the parameter name for the transit compartment rate
ka	absorption rate parameter name

Value

rxode2 model with transit compartment removed

See Also

Other absorption: `addTransit()`, `addWeibullAbs()`

Examples

```
# In this example the transit is added and then a few are removed
readModelDb("PK_1cmt_des") |> addTransit(4) |> removeTransit(3)
readModelDb("PK_1cmt_des") |> addTransit(4) |> removeTransit()
```

searchReplace

Search within a model to replace part of the model

Description

Search within a model to replace part of the model

Usage

```
searchReplace(object, find, replace)
searchReplaceHelper(object, find, replace)
```

Arguments

object	function specifying the nlmixr2 model
find, replace	Character scalars of parts of the model to replace

Value

object with find replaced with replace

Functions

- `searchReplaceHelper()`: A helper function for `searchReplace` (not intended for users to use directly)

setCombineType	<i>Change the default combine type for the package</i>
----------------	--

Description

Change the default combine type for the package

Usage

```
setCombineType(combineType = c("snake", "camel", "dot", "blank"))
```

Arguments

combineType	this is the default combine type: - "default": default combine - "snake": snake_case combine - "camel": camelCase combine - "dot": dot combine (i.e. "a.b") - "blank": no separator (i.e. "ab")
-------------	---

Author(s)

Matthew L. Fidler

Examples

```
# Change to the popular snake_case
setCombineType("snake")
defaultCombine("a", "b")

# Change back to nlmixr2/rxode2 default camelCase

setCombineType("camel")
defaultCombine("a", "b")

# This is used to change the naming convention for parameters
# produced by this package
```

updateOncologyXenograftSimeoni2004	<i>Update an oncology xenograft model based on Simeoni 2004</i>
------------------------------------	---

Description

Update an oncology xenograft model based on Simeoni 2004

Usage

```
updateOncologyXenograftSimeoni2004(  
  object,  
  ncmt,  
  damagedCmtName = "damagedCells",  
  drugEffectName = "drugEffectCyclingCells",  
  undamagedCmtName = "cyclingCells",  
  tumorVolName = "tumorVol",  
  transitRateName = "damageTransit"  
)
```

Arguments

object	Fitted object or function specifying the model.
ncmt	The desired number of damaged cell compartments
damagedCmtName, undamagedCmtName, tumorVolName	character string names for the compartments for damaged cells, undamaged cells, and the calculated tumor volume (the sum of undamaged and damaged cells)
drugEffectName, transitRateName	character string names of the drug effect and transit rate (as used in the model block)

Value

An updated model with the new number of compartments

Examples

```
library(rxode2)  
readModelDb("oncology_xenograft_simeoni_2004") %>%  
  updateOncologyXenograftSimeoni2004(ncmt = 5)
```

Index

- * **Indirect response**
 - addIndirect, [11](#)
 - addIndirectLin, [13](#)
 - convertKinR0, [21](#)
 - * **Internal**
 - searchReplace, [34](#)
 - * **PD**
 - addBaseline1exp, [2](#)
 - addBaselineConst, [3](#)
 - addBaselineExp, [4](#)
 - addBaselineLin, [5](#)
 - addDirectLin, [8](#)
 - convertEmax, [19](#)
 - convertLogLin, [22](#)
 - convertQuad, [24](#)
 - * **absorption**
 - addTransit, [16](#)
 - addWeibullAbs, [17](#)
 - removeTransit, [33](#)
 - * **datasets**
 - modeldb, [27](#)
- addBaseline1exp, [2](#), [4](#), [5](#), [8](#), [20](#), [22](#), [24](#)
addBaselineConst, [3](#), [3](#), [5](#), [8](#), [20](#), [22](#), [24](#)
addBaselineExp, [3](#), [4](#), [4](#), [5](#), [8](#), [20](#), [22](#), [24](#)
addBaselineLin, [3–5](#), [5](#), [8](#), [20](#), [22](#), [24](#)
addBioavailability (addCmtProp), [6](#)
addCmtProp, [6](#)
addDepot, [7](#)
addDirectLin, [3–5](#), [8](#), [20](#), [22](#), [24](#)
addDirToModelDb, [9](#)
addDur (addCmtProp), [6](#)
addEffectCmtLin, [9](#)
addEta, [10](#)
addFileToModelDb (addDirToModelDb), [9](#)
addIndirect, [11](#), [14](#), [22](#)
addIndirectLin, [12](#), [13](#), [22](#)
addIni (addCmtProp), [6](#)
addLag (addCmtProp), [6](#)
addLogEstimates, [14](#)
addRate (addCmtProp), [6](#)
addResErr, [15](#)
addTransit, [16](#), [18](#), [34](#)
addWeibullAbs, [17](#), [17](#), [34](#)
blankCombine (defaultCombine), [25](#)
camelCombine (defaultCombine), [25](#)
combinePaste2, [18](#)
convertEmax, [3–5](#), [8](#), [19](#), [22](#), [24](#)
convertEmaxHill, [20](#)
convertKinR0, [12](#), [14](#), [21](#)
convertLogLin, [3–5](#), [8](#), [20](#), [22](#), [24](#)
convertMM, [23](#)
convertQuad, [3–5](#), [8](#), [20](#), [22](#), [24](#)
defaultCombine, [25](#)
dotCombine (defaultCombine), [25](#)
fakeCc, [26](#)
modeldb, [27](#)
modllib, [28](#)
pkTrans, [28](#)
readModelDb, [31](#)
removeDepot, [32](#)
removeLinesAndInis, [32](#)
removeTransit, [17](#), [18](#), [33](#)
searchReplace, [34](#)
searchReplaceHelper (searchReplace), [34](#)
setCombineType, [35](#)
snakeCombine (defaultCombine), [25](#)
updateOncologyXenograftSimeoni2004, [35](#)