

Package ‘monolix2rx’

July 15, 2025

Title Converts 'Monolix' Models to 'rxode2'

Version 0.0.5

Description 'Monolix' is a tool for running mixed effects model using 'saem'. This tool allows you to convert 'Monolix' models to 'rxode2' (Wang, Hallow and James (2016) <[doi:10.1002/psp4.12052](https://doi.org/10.1002/psp4.12052)>) using the form compatible with 'nlmixr2' (Fidler et al (2019) <[doi:10.1002/psp4.12445](https://doi.org/10.1002/psp4.12445)>). If available, the 'rxode2' model will read in the 'Monolix' data and compare the simulation for the population model individual model and residual model to immediately show how well the translation is performing. This saves the model development time for people who are creating an 'rxode2' model manually. Additionally, this package reads in all the information to allow simulation with uncertainty (that is the number of observations, the number of subjects, and the covariance matrix) with a 'rxode2' model. This is complementary to the 'babelmixr2' package that translates 'nlmixr2' models to 'Monolix' and can convert the objects converted from 'monolix2rx' to a full 'nlmixr2' fit. While not required, you can get/install the 'lixoftConnectors' package in the 'Monolix' installation, as described at the following url <<https://monolixsuite.slp-software.com/r-functions/2024R1/installation-and-initialization>>. When 'lixoftConnectors' is available, 'Monolix' can be used to load its model library instead manually setting up text files (which only works with old versions of 'Monolix').

License MIT + file LICENSE

URL <https://nlmixr2.github.io/monolix2rx/>,
<https://github.com/nlmixr2/monolix2rx/>

Encoding UTF-8

RoxygenNote 7.3.2

LinkingTo dparser, rxode2, Rcpp

Imports Rcpp, checkmate, cli, dparser, withr, ggplot2, ggforce, stringi, crayon, lotri, magrittr, rxode2 (>= 3.0.4)

Suggests devtools, testthat (>= 3.0.0), xgxr, vdiff

Config/testthat/edition 3

Config/Needs/website rmarkdown
NeedsCompilation yes
Author Matthew Fidler [aut, cre] (ORCID: <https://orcid.org/0000-0001-8538-6691>),
Justin Wilkins [ctb] (ORCID: <https://orcid.org/0000-0002-7099-9396>)
Maintainer Matthew Fidler <matthew.fidler@gmail.com>
Repository CRAN
Date/Publication 2025-07-15 21:20:01 UTC

Contents

mlxtran	2
mlxTxt	3
monolix2rx	4
Index	6

mlxtran	<i>Read and parse mlxtran lines</i>
---------	-------------------------------------

Description

Read and parse mlxtran lines

Usage

mlxtran(file, equation = FALSE, update = FALSE)

Arguments

- file mlxtran file to process
- equation parse the equation block to rxode2 (some models cannot be translated)
- update when true, try to update the parameter block to the final parameter estimates

Value

mlxtran object

Author(s)

Matthew L. Fidler

Examples

```
# First load in the model; in this case the theo model
# This is modified from the Monolix demos by saving the model
# File as a text file (hence you can access without model library)
# setup.
#
# This example is also included in the monolix2rx package, so
# you refer to the location with `system.file()`:

pkgTheo <- system.file("theo", package="monolix2rx")

mlx <- mlxtran(file.path(pkgTheo, "theophylline_project.mlxtran"))

mlx
```

mlxTxt

Get equation block from a Monolix model txt file

Description

Get equation block from a Monolix model txt file

Usage

```
mlxTxt(file, retFile = FALSE)
```

Arguments

file	string representing the model text file. Can be lib:fileName.txt if library setup/available
retFile	boolean that tells mlxTxt() to return the file name instead of error if the file does not exist

Value

parsed equation or file name

Author(s)

Matthew L. Fidler

Examples

```
# First load in the model; in this case the theo model
# This is modified from the Monolix demos by saving the model
# File as a text file (hence you can access without model library)
# setup.
#
# This example is also included in the monolix2rx package, so
# you refer to the location with `system.file()`:
```

```
pkgTheo <- system.file("theo", package="monolix2rx")

mod <- mlxTxt(file.path(pkgTheo, "oral1_1cpt_kaVC1.txt"))

mod
```

monolix2rx

*Translate a monolix file to rxode2***Description**

Translate a monolix file to rxode2

Usage

```
monolix2rx(
  mlxtran,
  update = TRUE,
  thetaMatType = c("sa", "lin"),
  sd = 1,
  cor = 1e-05,
  theta = 0.5,
  ci = 0.95,
  sigdig = 3,
  envir = parent.frame()
)
```

Arguments

<code>mlxtran</code>	file name for mlxtran to translate to rxode2
<code>update</code>	is a boolean that represents if the final parameter estimates should be used for the translation (when present)
<code>thetaMatType</code>	This lists the preferred source for thetaMat covariance matrix. By default it is sa for simulated annealing, though you could use lin for linearized covariance calculation. If only one is present, then use whatever is present
<code>sd</code>	Default standard deviation for between subject variability/inter-occasion variability that are missing.
<code>cor</code>	Default correlation for missing correlations estimate
<code>theta</code>	default population estimate
<code>ci</code>	confidence interval for validation, by default 0.95
<code>sigdig</code>	number of significant digits for validation, by default 3
<code>envir</code>	represents the environment used for evaluating the corresponding rxode2 function

Value

rxode2 model

Author(s)

Matthew L. Fidler

Examples

```
# First load in the model; in this case the theo model
# This is modified from the Monolix demos by saving the model
# File as a text file (hence you can access without model library)
# setup.
#
# This example is also included in the monolix2rx package, so
# you refer to the location with `system.file()`:

pkgTheo <- system.file("theo", package="monolix2rx")

rx <- monolix2rx(file.path(pkgTheo, "theophylline_project.mlxtran"))

pkgCov <- system.file("cov", package="monolix2rx")

rx <- monolix2rx(file.path(pkgCov, "warfarin_covariate3_project.mlxtran"))

rx
```

Index

mlxtran, [2](#)
mlxTxt, [3](#)
monolix2rx, [4](#)