Package: nlmixr2targets (via r-universe)

July 18, 2024

Title Targets for 'nlmixr2' Pipelines

Version 0.0.0.9000

Description 'nlmixr2' often has long runtimes. A pipeline toolkit
tailored to 'nlmixr2' workflows leverages 'targets' and
'nlmixr2' to ease reproducible workflows. 'nlmixr2targets'
ensures minimal rework in model development with 'nlmixr2' and
'targets' by simplifying and standardizing models and datasets.

License GPL (>= 2)

Suggests covr, knitr, nlmixr2data, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports checkmate, digest, nlmixr2est, rxode2 (>= 2.0.14), targets

URL https://nlmixr2.github.io/nlmixr2targets/

VignetteBuilder knitr

Repository https://nlmixr2.r-universe.dev

RemoteUrl https://github.com/nlmixr2/nlmixr2targets

RemoteRef HEAD

RemoteSha 83a437ed4d0bee13287804092416f4f903cadb15

Contents

---

assign_origData   *Replace the fit data with the original data, then return the modified fit*

Description

This function is intended for use within `nlmixr2targets` target creation, and it's not typically invoked by users.

Usage

```
assign_origData(fit, data)
```

Arguments

fit	an estimated `nlmixr2` object
data	the data from the original fit

Value

The fit with the data added back in as `fit$env$origData`

---

nlmixr_data_simplify   *Standardize and simplify data for nlmixr2 estimation*

Description

This function is typically not needed by end users.

Usage

```
nlmixr_data_simplify(data, object, table = list())
```

Arguments

data	nlmixr data
object	an nlmixr_ui object (e.g. the output of running `nlmixr(object = model)`
table	The output table control object (like 'tableControl()')

Details

The standardization keeps columns that rxode2 and nlmixr2 use along with the covariates. Column order is standardized (rxode2 then nlmixr2 then alphabetically sorted covariates), and rxode2 and nlmixr2 column names are converted to lower case.

Value

The data with the nlmixr2 column lower case and on the left and the covariate columns on the right and alphabetically sorted.

See Also

Other Simplifiers: [nlmixr_object_simplify](#)()

```
nlmixr_object_simplify
```
Simplify an nlmixr object

Description

This function is typically not needed by end users.

Usage

```
nlmixr_object_simplify(object)
```

Arguments

object   Fitted object or function specifying the model.

Details

The object simplification removes comments (so please use label() instead of comments to label parameters) and then converts the object to a "nlmixrui" object.

Since setting initial conditions with cmt(0) does not work with targets, the function definition of the object must set it with cmt(initial). cmt(initial) will be converted to cmt(0) before passing to nlmixr2.

Value

object converted to a nlmixrui object. The model name is always "object".

See Also

Other Simplifiers: [nlmixr_data_simplify](#)()

---

tar_nlmixr   *Generate a set of targets for nlmixr estimation*

Description

The targets generated will include the name as the final estimation step, paste(name, "object_simple", sep = "_tar_") (e.g. "pheno_tar_object_simple") as the simplified model object, and paste(name, "data_simple", sep = "_tar_") (e.g. "pheno_tar_data_simple") as the simplified data object.

Usage

```
tar_nlmixr(
  name,
  object,
  data,
  est = NULL,
  control = list(),
  table = nlmixr2est::tableControl(),
  env = parent.frame()
)

tar_nlmixr_raw(
  name,
  object,
  data,
  est,
  control,
  table,
  object_simple_name,
  data_simple_name,
  fit_simple_name,
  env
)
```

Arguments

name   Symbol, name of the target. A target name must be a valid name for a symbol in
                   R, and it must not start with a dot. Subsequent targets can refer to this name sym-
                   bolically to induce a dependency relationship: e.g. tar_target(downstream_target,
                   f(upstream_target)) is a target named downstream_target which depends
                   on a target upstream_target and a function f(). In addition, a target's name
                   determines its random number generator seed. In this way, each target runs with
                   a reproducible seed so someone else running the same pipeline should get the
                   same results, and no two targets in the same pipeline share the same seed. (Even
                   dynamic branches have different names and thus different seeds.) You can re-
                   cover the seed of a completed target with tar_meta(your_target, seed) and

                        run `tar_seed_set()` on the result to locally recreate the target's initial RNG state.

`object`	Fitted object or function specifying the model.
`data`	nlmixr data
`est`	estimation method (all methods are shown by 'nlmixr2AllEst()'). Methods can be added for other tools
`control`	The estimation control object. These are expected to be different for each type of estimation method
`table`	The output table control object (like 'tableControl()')
`env`	The environment where the model is setup (not needed for typical use)

`object_simple_name, data_simple_name, fit_simple_name`

                        target names to use for the simplified object, simplified data, fit of the simplified object with the simplified data, and fit with the original data re-inserted.

Details

For the way that the objects are simplified, see `nlmixr_object_simplify()` and `nlmixr_data_simplify()`. To see how to write initial conditions to work with targets, see `nlmixr_object_simplify()`.

Value

A list of targets for the model simplification, data simplification, and model estimation.

Functions

- `tar_nlmixr_raw()`: An internal function to generate the targets

Examples

```
## Not run:
library(targets)
targets::tar_script({
pheno <- function() {
  ini({
    lcl <- log(0.008); label("Typical value of clearance")
    lvc <-  log(0.6); label("Typical value of volume of distribution")
    etalcl + etalvc ~ c(1,
                         0.01, 1)
    cpaddSd <- 0.1; label("residual variability")
  })
  model({
    cl <- exp(lcl + etalcl)
    vc <- exp(lvc + etalvc)
    kel <- cl/vc
    d/dt(central) <- -kel*central
    cp <- central/vc
    cp ~ add(cpaddSd)
  })
}
```

```
list(
  tar_nlmixr(
    name = pheno_model,
    object = pheno,
    data = nlmixr2data::pheno_sd,
    est = "saem"
  )
)
})
targets::tar_make()

## End(Not run)
```

---

tar_nlmixr_multimodel   *Generate a list of models based on a single dataset and estimation*
                         method

Description

Generate a list of models based on a single dataset and estimation method

Usage

```
tar_nlmixr_multimodel(
  name,
  ...,
  data,
  est,
  control = list(),
  table = nlmixr2est::tableControl(),
  env = parent.frame()
)
```

Arguments

name   Symbol, name of the target. A target name must be a valid name for a symbol in
                    R, and it must not start with a dot. Subsequent targets can refer to this name sym-
                    bolically to induce a dependency relationship: e.g. tar_target(downstream_target,
                    f(upstream_target)) is a target named downstream_target which depends
                    on a target upstream_target and a function f(). In addition, a target's name
                    determines its random number generator seed. In this way, each target runs with
                    a reproducible seed so someone else running the same pipeline should get the
                    same results, and no two targets in the same pipeline share the same seed. (Even
                    dynamic branches have different names and thus different seeds.) You can re-
                    cover the seed of a completed target with tar_meta(your_target, seed) and
                    run tar_seed_set() on the result to locally recreate the target's initial RNG
                    state.

...   Named arguments with the format "Model description" = modelFunction
```

| | |
|---|---|
| data | nlmixr data |
| est | estimation method (all methods are shown by 'nlmixr2AllEst()'). Methods can be added for other tools |
| control | The estimation control object. These are expected to be different for each type of estimation method |
| table | The output table control object (like 'tableControl()') |
| env | The environment where the model is setup (not needed for typical use) |

### Value

A list of targets for the model simplification, data simplification, and model estimation.

---

tar_nlmixr_multimodel_has_self_reference

*Does the model list refer to another model in the model list?*

---

### Description

Does the model list refer to another model in the model list?

### Usage

```
tar_nlmixr_multimodel_has_self_reference(model_list, name)

tar_nlmixr_multimodel_has_self_reference_single(model, name)
```

Arguments

model_list	A named list of calls for model targets to be created
name	Symbol, name of the target. A target name must be a valid name for a symbol in R, and it must not start with a dot. Subsequent targets can refer to this name symbolically to induce a dependency relationship: e.g. tar_target(downstream_target, f(upstream_target)) is a target named downstream_target which depends on a target upstream_target and a function f(). In addition, a target's name determines its random number generator seed. In this way, each target runs with a reproducible seed so someone else running the same pipeline should get the same results, and no two targets in the same pipeline share the same seed. (Even dynamic branches have different names and thus different seeds.) You can recover the seed of a completed target with tar_meta(your_target, seed) and run tar_seed_set() on the result to locally recreate the target's initial RNG state.
model	A single model call for the model target to be created

Value

A logical vector the same length as model_list indicating if the model is self-referential to another model in the list

Functions

- `tar_nlmixr_multimodel_has_self_reference_single()`: A helper function to look at each call for each model separately

---

tar_nlmixr_multimodel_parse

Generate nlmixr multimodel target set for all models in one call to `tar_nlmixr_multimodel()`

Description

Generate nlmixr multimodel target set for all models in one call to `tar_nlmixr_multimodel()`

Usage

```
tar_nlmixr_multimodel_parse(name, data, est, control, table, model_list, env)
```

Arguments

name   Symbol, name of the target. A target name must be a valid name for a symbol in
                R, and it must not start with a dot. Subsequent targets can refer to this name sym-
                bolically to induce a dependency relationship: e.g. `tar_target(downstream_target,`
                `f(upstream_target))` is a target named `downstream_target` which depends
                on a target `upstream_target` and a function `f()`. In addition, a target's name
                determines its random number generator seed. In this way, each target runs with
                a reproducible seed so someone else running the same pipeline should get the
                same results, and no two targets in the same pipeline share the same seed. (Even
                dynamic branches have different names and thus different seeds.) You can re-
                cover the seed of a completed target with `tar_meta(your_target, seed)` and
                run [`tar_seed_set()`](#) on the result to locally recreate the target's initial RNG
                state.

data   nlmixr data

est   estimation method (all methods are shown by 'nlmixr2AllEst()'). Methods can
                be added for other tools

control   The estimation control object. These are expected to be different for each type
                of estimation method

table   The output table control object (like 'tableControl()')

model_list   A named list of calls for model targets to be created

env   The environment where the model is setup (not needed for typical use)

tar_nlmixr_multimodel_single

Generate a single nlmixr multimodel target set for one model

Description

Generate a single nlmixr multimodel target set for one model

Usage

```
tar_nlmixr_multimodel_single(object, name, data, est, control, table, env)
```

Arguments

object
: Fitted object or function specifying the model.

name
: Symbol, name of the target. A target name must be a valid name for a symbol in R, and it must not start with a dot. Subsequent targets can refer to this name symbolically to induce a dependency relationship: e.g. `tar_target(downstream_target, f(upstream_target))` is a target named `downstream_target` which depends on a target `upstream_target` and a function `f()`. In addition, a target's name determines its random number generator seed. In this way, each target runs with a reproducible seed so someone else running the same pipeline should get the same results, and no two targets in the same pipeline share the same seed. (Even dynamic branches have different names and thus different seeds.) You can recover the seed of a completed target with `tar_meta(your_target, seed)` and run `tar_seed_set()` on the result to locally recreate the target's initial RNG state.

data
: nlmixr data

est
: estimation method (all methods are shown by 'nlmixr2AllEst()'). Methods can be added for other tools

control
: The estimation control object. These are expected to be different for each type of estimation method

table
: The output table control object (like 'tableControl()')

env
: The environment where the model is setup (not needed for typical use)

Index