Package 'shinyMixR'

November 14, 2024

Title Interactive 'shiny' Dashboard for 'nlmixr2'

Version 0.5.0

Description An R shiny user interface for the 'nlmixr2' (Fidler et al (2019) <doi:10.1002/psp4.12445>) package, designed to simplify the modeling process for users. Additionally, this package includes supplementary functions to further enhances the usage of 'nlmixr2'.

Depends R (>= 3.5.0), shiny, ggplot2

Imports gridExtra, collapsibleTree, shinyAce, DT, bs4Dash, shinyWidgets, stringi, R3port, whisker, plotly, patchwork, shinyjs, ps, xfun, fresh, nlmixr2, nlmixr2est, magrittr, cli

Suggests xpose, xpose.nlmixr2, nlme, testthat, shinytest2, knitr, rmarkdown, rlang, miniUI, shinyFiles, rstudioapi

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Richard Hooijmaijers [aut, cre, cph], Teun Post [aut, cph], LAPP Consultants [fnd, cph], Matthew Fidler [ctb], Veerle van Leemput [ctb]

Maintainer Richard Hooijmaijers <richardhooijmaijers@gmail.com>

Repository CRAN

Date/Publication 2024-11-14 16:10:03 UTC

Contents

adpt_meta *Adapt meta information inside a nlmixr UIF*

Description

regular expressions are used to search for meta data inside a model file. This meta data is then updated with the provided new values

Usage

```
adpt_meta(mdl, newvals)
```

Arguments

mdl	character with the name of the model to adapt
newvals	list with characteristics/meta data to adapt

Value

character vector with model including the adapted meta data

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
  adpt_meta("model.r",newvals=list(imp=4,ref="run 1"))

## End(Not run)
```

create_proj *Creates a new project*

Description

Creates a new project which basically means that within the specified folder, the necessary folder structure will be created and some example models will be placed in it.

Usage

```
create_proj(loc = ".", overwrite = FALSE)
```

Arguments

loc	character with the location where the project should be created
overwrite	logical indicating if files should be overwritten if already exists

Value

nothing will be returned by the function (only system commands are issued)

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 create_proj()

## End(Not run)
```

---

exploreplot *Function to create plot from data exploration app*

Description

This function creates a text string for a ggplot based on an input list. This function is specifically written to be used with the shiny app for data exploration.

Usage

```
exploreplot(inputlist)
```

Arguments

inputlist	list with input items to create a plot

Value

a character string with the ggplot code

Author(s)

Richard Hooijmaijers

Examples

```
## Not run: exploreplot(input)
```

`fit_plot` *Create fit plot*

Description

Creates a fit plot either using the xpose.nlmixr package or using a default ggplot call

Usage

```
fit_plot(
  dfrm,
  type = "xpose",
  by = "ID",
  idv = "TIME",
  obs = "DV",
  pred = "PRED",
  ipred = "IPRED",
  grp = "ID",
  logy = TRUE,
  scales = "fixed",
  mdlnm = NULL,
  outnm = NULL,
  projloc = ".",
  ...
)
```

Arguments

`dfrm`	data frame as created by the nlmixr function
`type`	character defining the type of plot that should be created. currently "xpose" and "user" are supported for xpose or ggplot style of plots
`by`	character vector with variables for facetting
`idv`	independent variable or x variable
`obs`	variable with observed data points
`pred`	variable with predicted data points
`ipred`	variable with individual predicted data points
`grp`	variable for grouping (mainly to draw separate lines)
`logy`	logical if y-axis should be displayed on log scale
`scales`	character of length one defining the scale parameter of ggplot (e.g. "fixed", "free","free_y",etc)
`mdlnm`	character with name of the model
`outnm`	character with name of the output file (see details)
`projloc`	character with the base location of the shinyMixR project
`...`	additional arguments passed to `ltx_plot` or `html_plot`

Details

In case a model is saved, a directory with the name of the model is created within the analysis folder of the current project. Then within this folder the file is saved as outnm. This method was chosen so the interface can easily index applicable files for a certain model. However, this means that output is always saved in this directly regardless of the location of outnm

Value

in case no outnm is defined a ggplot object will be returned otherwise the results are saved to disk

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 fit_plot(res)

## End(Not run)
```

---

get_meta *Get the meta data out of a model function*

Description

This function gets only the meta data from a function

Usage

```
get_meta(mdl)
```

Arguments

mdl	character with the path of the model function

Value

A list with the models meta data

Examples

```
## Not run:
  get_meta("run1.r")

## End(Not run)
```

get_proj *Read in and update model results in project object*

Description

This function creates or updates a project object with models and/or results emerged from nlmixr2 runs. A check is performed to see if newer results are present and only updates these.

Usage

```
get_proj(projloc = ".", geteval = TRUE)
```

Arguments

projloc	character with the base location of the shinyMixR project
geteval	logical indicating if the model functions should be evaluated

Value

A named list with information for each model in the 'projloc'

Examples

```
## Not run:
  proj <- get_proj()

## End(Not run)
```

gof_plot *Create goodness of fit plots*

Description

Creates goodness of fit plots either using the xpose.nlmixr package or using a default ggplot call

Usage

```
gof_plot(
  dfrm,
  type = "xpose",
  mdlnm = NULL,
  colby = NULL,
  ptype = "all",
  outnm = NULL,
  projloc = ".",
  title = NULL,
```

```
  linscale = FALSE,
  ...
)
```

Arguments

dfrm	data frame as created by the nlmixr function
type	character defining the type of plot that should be created. currently "xpose" and "user" are supported for xpose or ggplot style of plots
mdlnm	character with name of the model
colby	character vector of length one specifying the variable to color on (for now can be only one variable)
ptype	The type of plots to create. Currently the following is accepted: "all", "ipred.dv", "pred.dv", "idv.res", "pred.res"
outnm	character with name of the output file (see details)
projloc	character with the base location of the shinyMixR project
title	character with the title to place above the plot
linscale	Logical indicating if the scales should be set to linear for DV, PRED and IPRED plots
...	additional arguments passed to ltx_plot or html_plot

Details

In case a model is saved, a directory with the name of the model is created within the analysis folder of the current project. Then within this folder the file is saved as outnm. This method was chosen so the interface can easily index applicable files for a certain model. However, this means that output is always saved in this directly regardless of the location of outnm

Value

in case no outnm is defined a ggplot object will be returned otherwise the results are saved to disk

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 gof_plot(res)

## End(Not run)
```

incr_mdl *Increments a model name*

Description

A model name is incremented either by incrementing numerical or alpha numerical. Furthermore it is possible to check the existence of the incremented model and take this into account.

Usage

```
incr_mdl(mod, checkloc = NULL)
```

Arguments

mod	character with the model name
checkloc	character with the location to check for existence of a file

Value

character with the incremented name

Author(s)

Richard Hooijmaijers

Examples

```
incr_mdl("run01.r")
```

module_dataexplore_server *Data exploration module for server*

Description

Data exploration module for server

Usage

```
module_dataexplore_server(id, r)
```

Arguments

id	Module id
r	reactive values object that is defined top-level

Value

No return value, called for side effects

---

module_dataexplore_ui *Data exploration module for UI*

Description

Shiny module for data exploration

Usage

```
module_dataexplore_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

---

module_edit_server *Editor module for server*

Description

Editor module for server

Usage

```
module_edit_server(id, r, settings)
```

Arguments

id	Module id
r	reactive values object that is defined top-level
settings	reactive value with the app settings

Value

No return value, called for side effects

module_edit_ui *Editor module for UI*

Description

Shiny module for model editor

Usage

```
module_edit_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

module_fitplots_server *Fit plots module for server*

Description

Fit plots module for server

Usage

```
module_fitplots_server(id, r, settings)
```

Arguments

id	Module id
r	reactive values object that is defined top-level
settings	reactive value with the app settings

Value

No return value, called for side effects

module_fitplots_ui *Fit plots module for UI*

Description

Shiny module for fit plots

Usage

```
module_fitplots_ui(id, proj_obj)
```

Arguments

id	Module id
proj_obj	Project object

Value

A list of html tags used for th UI of the app

module_gof_server *GOF plots module for server*

Description

GOF plots module for server

Usage

```
module_gof_server(id, r, settings)
```

Arguments

id	Module id
r	reactive values object that is defined top-level
settings	reactive value with the app settings

Value

No return value, called for side effects

module_gof_ui *GOF plots module for UI*

Description

Shiny module for GOF plots

Usage

```
module_gof_ui(id, proj_obj)
```

Arguments

id	Module id
proj_obj	Project object

Value

A list of html tags used for th UI of the app

module_metadata_server *meta data module for server*

Description

meta data module for server

Usage

```
module_metadata_server(
  id,
  type,
  selline = NULL,
  sellmod = NULL,
  sellcont = NULL,
  r
)
```

Arguments

id	Module id
type	character with the type of action (either "save" or "overview")
selline	reactive with the selected line for a model (for type "overview")
sellmod	reactive with the selected model (for type "save")
sellcont	reactive with the content of the selected model (for type "save")
r	reactive values object that is defined top-level

Value

a reactive with the meta data information

`module_metadata_ui` *metadata module for UI*

Description

Shiny module for meta data

Usage

```
module_metadata_ui(id, type)
```

Arguments

`id`	Module id
`type`	character with the type of button to present (either "save" or "overview")

Value

A list of html tags used for th UI of the app

`module_overview_server` *Overview module for server*

Description

Overview module for server

Usage

```
module_overview_server(id, r)
```

Arguments

`id`	Module id
`r`	reactive values object that is defined top-level

Value

No return value, called for side effects

module_overview_ui *Overview module for UI*

Description

Shiny module for overview

Usage

```
module_overview_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

module_pt_server *Parameter table module for server*

Description

Parameter table module for server

Usage

```
module_pt_server(id, r)
```

Arguments

id	Module id
r	reactive values object that is defined top-level

Value

No return value, called for side effects

module_pt_ui *Parameter table module for UI*

Description

Shiny module for parameter table

Usage

```
module_pt_ui(id, proj_obj)
```

Arguments

id	Module id
proj_obj	Project object

Value

A list of html tags used for th UI of the app

module_reports_server *Reporting module for server*

Description

Reporting module for server

Usage

```
module_reports_server(id, r)
```

Arguments

id	Module id
r	reactive values object that is defined top-level

Value

No return value, called for side effects

module_reports_ui *Reporting module for UI*

Description

Shiny module for reporting

Usage

```
module_reports_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

module_run_server *Run model module for server*

Description

Run model module for server

Usage

```
module_run_server(id, r)
```

Arguments

id	Module id
r	reactive values object that is defined top-level

Value

No return value, called for side effects

module_run_ui *Run model module for UI*

Description

Shiny module for running models

Usage

```
module_run_ui(id, proj_obj)
```

Arguments

id	Module id
proj_obj	Project object

Value

A list of html tags used for th UI of the app

module_scripts_server *Run script module for server*

Description

Run script module for server

Usage

```
module_scripts_server(id, files = NULL, loc = "temp", r)
```

Arguments

id	Module id
files	character vector of files to apply the scripts on, usually a reactive
loc	character with the location where the temp scripts are saved (created when not existing)
r	reactive values object that is defined top-level

Value

No return value, called for side effects

module_scripts_ui *Run script module for UI*

Description

Shiny module for running scripts

Usage

```
module_scripts_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

module_settings_server *Settings module for server*

Description

Settings module for server

Usage

```
module_settings_server(id)
```

Arguments

id	Module id

Value

a reactive with all input elements

module_settings_ui *Settings module for UI*

Description

Shiny module for settings

Usage

```
module_settings_ui(id)
```

Arguments

id	Module id

Value

A list of html tags used for th UI of the app

myalert *wrapper function for sweetalert in shinywidgets*

Description

This function gets list of widgets to include in run_shinymixr

Usage

```
myalert(text, type, ...)
```

Arguments

text	character with the text to display
type	character with the type of alert to display
...	other arguments passed to class

Value

No return value, called for side effects

numfmt *set significant digits without rounding higher numbers*

Description

This function sets significant digits without rounding any numbers

Usage

```
numfmt(x, sdig = 3, snc = 6)
```

Arguments

x	a numerical vector
sdig	a single number defining the number of significant digits
snc	a single number defining the scientific notation cutoff (higher means notation is only used for very small or very large numbers)

Value

a character vector with formatted numbers

Author(s)

Richard Hooijmaijers

Examples

```
numfmt(c(0.012,12345,1))
```

overview *Creates model overview*

Description

Create an overview of the models within a project. This overview includes the meta data of the models and if results are available, also the objective function and run-times

Usage

```
overview(proj_obj, ...)
```

Arguments

proj_obj	a project object created with get_proj
...	additional arguments passed to get_proj

Value

a data frame is returned with the overview

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 overview(proj_obj)

## End(Not run)
```

`par_table` *Create parameter table*

Description

Creates a table with the final estimates and percentage CV for all parameters in an nlmixr output file. This can be done for one or multiple models for easy comparison

Usage

```
par_table(
  proj,
  models,
  outnm = NULL,
  projloc = ".",
  bsv = FALSE,
  shrink = FALSE,
  backt = FALSE,
  formatting = FALSE,
  ...
)
```

Arguments

`proj`	project object
`models`	character vector with model names to create table for
`outnm`	character with name of the output file (see details)
`projloc`	character with the base location of the shinyMixR project
`bsv`	logical indicating if between subject variability (BSV) should be added to table
`shrink`	logical indicating if shrinkage should be added to table
`backt`	logical indicating if the backtransformed parameters should be returned opposed to the original values

formatting	logical indicating if the formatting should be applied to present the table (not implemented for latex output)
...	additional arguments passed to ltx_plot or html_plot

Details

In case a model is saved, a directory with the name of the model is created within the analysis folder of the current project. Then within this folder the file is saved as outnm. This method was chosen so the interface can easily index applicable files for a certain model. However, this means that output is always saved in this directly regardless of the location of outnm In case multiple models are selected the result will be written to the name of the first model in the models vector.

Value

in case no outnm is defined a data frame will be returned otherwise the results are saved to disk

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 par_table(proj,"run1")

## End(Not run)
```

---

run_nmx *Runs a nlmixr model*

Description

Runs an nlmixr model from a project object with the possibility to run in an external rsession using a system call (tested within linux only)

Usage

```
run_nmx(
  mod,
  proj = proj,
  ext = TRUE,
  saverds = TRUE,
  autoupdate = TRUE,
  projloc = ".",
  addcwres = TRUE,
  addnpde = TRUE
)
```

Arguments

mod	character with the model file present in project object
proj	project object
ext	logical indicating if the model should be run external in a separate r session
saverds	logical indicating if the model results should be saved in a rds file
autoupdate	logical indicating if the project object should automatically update
projloc	character with the base location of the shinyMixR project
addcwres	logical indicating if CWRES should be added to the output
addnpde	logical indicating if NPDE should be added to the output

Details

the meta data is obtained by compiling the model. The dataset, estimation method and control list are then included in the nlmixr call. Meta data is included in the model function which is comparable with NONMEM. This method was chosen so that all information to run a model is kept together in one function

Value

In case the model is not submitted in a separate R session, the results from nlmixr are returned otherwise the result of the system call will be returned

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 run_nmx("run1",proj)

## End(Not run)
```

---

run_shinymixr *Creates and run the interface*

Description

Creates and run the interface

Usage

```
run_shinymixr(wd = getwd(), ...)
```

Arguments

wd	character with the working directory
...	arguments passed to the shiny runApp function

Value

No return value, runs the shinyMixR interface

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 if (interactive()) run_shinymixr(".")

## End(Not run)
```

---

shinymixr_gadget *Rstudio gadget to select project and start app*

Description

Rstudio gadget to select project and start app

Usage

```
shinymixr_gadget()
```

Value

No return value, runs a gadget to start the shinyMixR interface

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
 if (interactive()) shinymixr_gadget()

## End(Not run)
```

sigdigs *set significant digits without rounding higher numbers*

Description

This function sets significant digits without rounding any numbers

Usage

```
sigdigs(x, sdig = 3)
```

Arguments

x	a numerical vector
sdig	a single number defining the number of significant digits

Value

A character vector with formatted numbers

theme_shinyMixR *theme for ggplot output in the shinyMixR package*

Description

This function provides a custom theme for ggplot output

Usage

```
theme_shinyMixR(fontsize = 12)
```

Arguments

fontsize	numeric with the default fontsize passed through to theme

Value

A list with ggplot theme elements

tree_overview *Creates tree overview of models*

Description

Create a graphical collapsible tree overview of the models within a project. This is mostly relevant in case the reference of models is included to visualise the relationship between models

Usage

```
tree_overview(proj_obj, ...)
```

Arguments

proj_obj	a project object created with get_proj
...	additional arguments passed to overview

Value

a data frame is returned with the overview

Author(s)

Richard Hooijmaijers

See Also

collapsibleTreeNetwork which does most of the work

Examples

```
## Not run:
 if (interactive()) tree_overview(proj_obj)

## End(Not run)
```

update_inits *Update initial estimates from a final model run*

Description

This function update the initial estimates from a model using the final estimates from a model result file. Currently this function assumes all models were submitted using shinyMixR opposed to vanilla nlmixr

Usage

```
update_inits(mod, res, out)
```

Arguments

mod	character with the entire model function included
res	character with the path to the model result RDS which holds the final estimated
out	character with the path for the updated model to save

Value

nothing will be returned the function saves the updated model to disk

Author(s)

Richard Hooijmaijers

Examples

```
## Not run:
  update_inits(readLines("run2.r"),"shinyMixR/run2.res.rds","run3.r")

## End(Not run)
```

Index