shinyMixR

Cheat Sheet

github.com/richardhooijmaijers/shinyMixR

Installation

Before installation, make sure you have R 3.4.1 or higher and at least the `nlmixr2` package installed and tested.

Installation of the package should be done using devtools:

```
install.packages('richardhooijmaijers/shinyMixR')
```

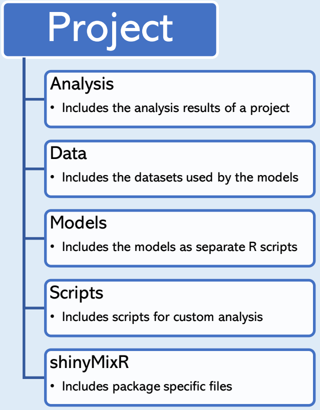
The package is based on the `shiny` and `bs4Dash` packages. Additional packages are off course `nlmixr2`, and for `xpose` type plotting `xpose.nlmixr2`.

Introduction

The shinyMixR package is developed as a model management tool for the nlmixr package. The package include a shiny (dashboard) interface but can also be used in an interactive R session. The package aims to help in managing, running, editing and analysing nlmixr2 models.

Folder Structure

A specific folder structure for a project is required and used by the package:



Project

- Analysis
  - Includes the analysis results of a project
- Data
  - Includes the datasets used by the models
- Models
  - Includes the models as separate R scripts
- Scripts
  - Includes scripts for custom analysis
- shinyMixR
  - Includes package specific files

A folder structure can be created including example data, models and scripts using:

```
create_proj()
```

This is the starting point for a project – you have to do this once per project.

Project Structure

Information regarding a project is maintained in the *project object*. This is a list within R with the following structure:

```
Object
|
|--- run 1
|     |--- model location
|     |--- model metadata
|     |--- model high level results
|
|--- run n
|
|--- general meta information
```

In case new models or results are available this is added to the object using the function `get_proj()`. When running interactively this function might be submitted to reflect the latest changes/results. The function will only search for newly created files. Within the interface this is done automatically or using the refresh button.

Interactive usage

For interactive usage, the most important functions are:

`create_proj()`	Create a folder structure for a `shinyMixR` project.
`run_nmx()`	Run a `nlmixr` model, possibly in a separate R session to overcome "freezing" of current session.
`overview()`	Create overview of all models in a project.
`tree_overview()`	Create a collapsible tree overview for visualizing relationship between models.
`par_table()`	Create dense parameter table for one or multiple models.
`gof_plot()`	Create a combination of most important goodness of fit plots.
`fit_plot()`	Create individual fit plots.
`get_proj()`	Get project information with available models and high level results.

Interface usage

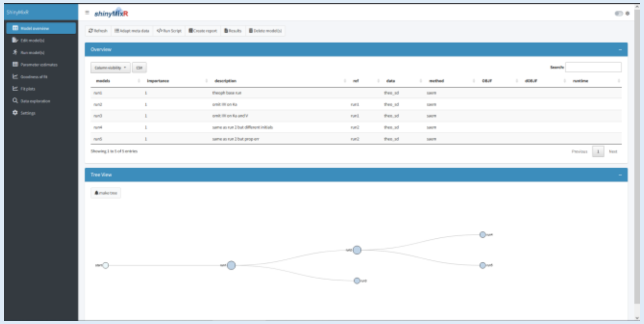
The interface can be started from the projects root folder:

```
run_shinymixr()
```

The app can be opened in an Rstudio window or web browser (using the launch.browser argument). The start window displays a dashboard with in the main window a (tree) overview of the models in the project structure.
The interface can be started at all times – even if the project was initially started in an interactive way; and *vice versa*.

Usage

There are various widgets available on the left side of the screen to run and analyse your models.



models	importance	description	ref	data	method	OBJF	dOBJF	runtime
run1	1	theoph base run		theo_sd	saem			
run2	1	omit IIV on Ka	run1	theo_sd	saem			
run3	1	omit IIV on Ka and V	run1	theo_sd	saem			
run4	1	same as run 2 but different initials	run2	theo_sd	saem			
run5	1	same as run 2 but prop err	run2	theo_sd	saem			

Showing 1 to 5 of 5 entries

Model overview	Starting point, contains overview of models including most important information. Possibility to add/adapt meta information, change views, run scripts/reports and visualize relationship between models.
Edit model	Create, save, duplicate and edit models. Possibility to use template models. Syntax highlighted editor using `shinyAce` editor.
Run model	Run one or multiple nlmixr models side-by-side. Perform model runs in separate R session(s). Keep track of progress of runs.
Parameter estimates	Create dense parameter table. Quickly observe parameter estimates and compare estimates between models. . Export to HTML or PDF output*.
Goodness of fit	Create a combination of most important goodness of fit plots. Export plots to HTML or PDF output*.
Fit plots	Create individual fit plots. Export to HTML or PDF output*.
Data exploration	Create exploratory data plots for input or model results. Includes option to make interactive plots and table view of your data
Settings	Adapt the settings of the app, like type of output and look of editor.

** Results are by default saved in the analysis subfolder, which makes them available for the interface. For creation of PDF, LaTeX including various packages is required.*