

Installation

Before installation, make sure you have R 3.4.1 or higher and at least the `nlmixr` 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 `shinydashboard` packages. Additional packages are of course `nlmixr`, and for `xpose` type plotting `xpose.nlmixr`.

Introduction

The shinyMixR package is developed as a model management tool for the `nlmixr` package. The package includes 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 `nlmixr` 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:

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

Interface usage

The interface can be started from the projects root folder:

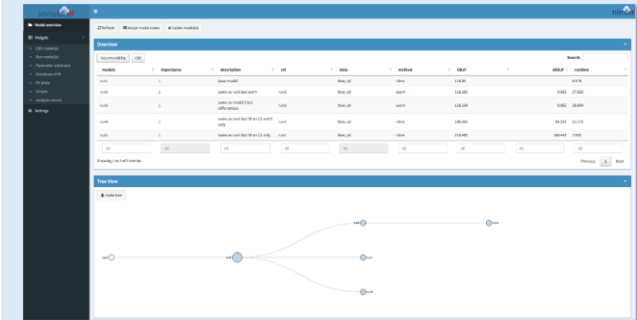
```
run_shinymixr()
```

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.



Model overview	Starting point, contains overview of models including most important information. Possibility to add/adapt meta information. Change views, export overview. Visualize relationship between models.
Edit model	Create, save, duplicate and edit models. Possibility to use template models. Syntax highlighted editor using <code>shinyAce</code> editor.
Run model	Run one or multiple <code>nlmixr</code> 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*.
Scripts	Run user generated scripts on selected model outputs. Possibility to run scripts included in the package or write your own scripts to create project specific output (scripts can also be used in interactive session).
Analysis results	Manage and view results created with <code>shinyMixR</code> either using the widgets, app scripts or ad-hoc (if named/saved according app conventions).
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.