Simulator

Version 1.1.0

December 2014

A model simulation engine for R, Matlab and C/C++ developers

Contents

1 Introduction

Simulator is a model simulation engine for computing models encoded in Mlxtran . It is the core of the SimulX clinical trial simulator for R and Matlab .

2 Installation and troubleshooting

2.1 Prerequisites

- Matlab users: Matlab version 2010b or higher;
  - MacOsX 10.6 or higher: you need `install_name_tool` to configure the Monolix package (generally provided within the system).
- R users:
  - Windows: R version 3.0 or higher, `Rcpp` version 0.11 or higher (or 0.10.6 or higher if the user is willing to recompile the package).
  - Linux: R version 3.0 or higher, `Rcpp` version 0.10.6 or higher; make sure you have `gcc/g++/make` installed or install them.
  - MacOsX 10.6 or higher: R version 3.0 or higher, `Rcpp` version 0.10.6 or higher; make sure you have `gcc/g++/make` installed or install them by installing the command-line tools of Xcode.

2.2 Installation

The Simulator installer is a self-extracting binary, which will guide the user through the installation. At the end of the process the user is requested to enter a license either as a key or through a license file.

The installer will create the following directories:

- `/Path/To/mlxlibrary-x.x.x/doc`: documentation. Matlab and R demos can also be found in `.../doc/ANNEX/mlxComputeDEMOS`
- `/Path/To/mlxlibrary-x.x.x/lib`: this directory contains all the necessary libraries. It also contains a directory `.../lib/mlxComputeR` with the source code for the R version of Simulator .

Important note: All Lixoft applications use the `lixoft` directory under the user directory to store application settings and temporary files (for instance the cpp modules that are generated on the fly from the `mlxTran` code). Therefore the user needs read and write access to this directory, besides MlxLibrary should not be installed in this directory.

Important note: The R version of Simulator uses `Rcpp`. Under Linux and under MacOsX, R users will have to compile the library `mlxComputeR.so` before using Simulator .
Under Windows, R users will only have to compile it if they use a `Rcpp` version prior to `Rcpp-0.11`.

2.3 Compilation with SimulX

The R and Matlab code of SimulX are available in a separate package, see:
`http://simulx.webpopix.org/`
The compilation of Simulator is then automatically launched the first time `initMlxR.R` is sourced in R or `initMlxM` is launched in Matlab.

2.4 Troubleshooting

1) With the Matlab version of Simulator , it is assumed that the library `mlxCompute.mex` is in the path. The user will receive a warning `Undefined function 'mlxCompute'` if the library cannot be found. The problem can be fixed by modifying the path (Matlab command `addpath`)

Note: see file
`/Path/To/mlxlibrary-x.x.x/doc/ANNEX/mlxComputeDEMOS/demosMATLAB/playAll.m`

2) The Matlab and R versions of Simulator will display a warning if the environment variable `LIXOFT_HOME` is not set. It shall be set to `/Path/To/mlxlibrary-x.x.x` (see `demosMATLAB/playAll.m` and `demosR/playAll.R`)

3) With the R version of Simulator , it is assumed that the R module `Rcpp` is installed. The user will receive a R warning if the module cannot be found. It can be installed by the usual command:

```
install.packages("Rcpp")
```

4) Under Windows, if an R user receives an error message

```
Function 'enterRNGScope' not provided by package 'Rcpp'
```

then the installed version of Rcpp is not compatible with Rcpp-0.11. There are then two possibilities:

- either update Rcpp simply by entering at R prompt:

  ```
  install.packages("Rcpp")
  ```

- or delete file /Path/To/mlxLibrary-x.x.x/runtime/lib/mlxComputeR.dll. The library mlxComputeR.dll will then be recompiled with the installed version of Rcpp (versions 0.10.6 or higher are supported).
  Note: this solution supposes that Rtools have been installed and is then recommended only for expert R users who have a good knowledge of modules installation.

5) Linux users and WINDOWS users who have to compile mlxComputeR.dll can receive following error message when trying to compile the library:

```
Error in dyn.load(mlxComputeFileName) :
    unable to load shared object
    '/Path/To/mlxLibrary-x.x.x/lib/mlxComputeR.so'
    /.../mlxComputeR.so: undefined symbol:mlxCompute_CAPIKill
```

This error message is du to a Makevars conflict: compilation and link flags are set through PKG_CPPFLAGS and PKG_LIBS which are defined in
/Path/To/mlxLibrary-x.x.x/runtime/lib/mlxComputeR/src/Makevars.
When this Makevars file conflicts with /.R/Makevars the flags are not correctly set. There are two solutions to solve this problem:

- delete /.R/Makevars
- modify PKG_LIBS and PKG_CPPFLAGS definitions in /.R/Makevars from

  PKG_CPPFLAGS = sthg
  PKG_LIBS = sthg

  to

  PKG_CPPFLAGS += sthg
  PKG_LIBS += sthg