MlxPlore

Version 1.1.0

January 2014

A software for model exploration and visualization

Contents

1 Introduction

MlxPlore is a graphical and interactive software for the exploration and visualization of complex pharmacometric models. MlxPlore also includes the ability to study the statistical variability of the models, and to model and study complex administrations designs.
MlxPlore is based on the Mlxtran model description language popularized by the Monolix software, therefore benefiting from the remarkable flexibility and power of Mlxtran to easily encode complex pharmacometric, PKPD and statistical models. Mlxtran for MlxPlore is described in a separate document (file **'MlxtranModel_forMlxplore.pdf'**).
MlxPlore does not require Monolix, although they make for a powerful combination, enabling to use the same, human-readable model description, to finely explore the properties of the model on the one hand, and on the other hand use the same model for advanced parameter estimation in the context of population analysis and mixed effect statistics.
The examples studied in this guide are located into the directory

```
<user home path>/lixoft/mlxplore/mlxplore100/demos/examples
```

2 Installing MlxPlore

2.1 Downloading packages

The MlxPlore packages can be downloaded through the download manager hosted at `http://download.lixoft.com`. The download manager is available for users provided with an access key. Different MlxPlore packages are available, depending on the version of the operating system. MlxPlore currently supports Windows XP/Vista/Seven/8, Linux (all common distributions) 32/64 bits.

Choice of MlxPlore versions

- Linux (32 bits)
- Linux (64 bits)
- Windows

2.2 Prerequisites - Linux specifics

- install `sharutils` : uudecode is required to uncompress the MlxPlore package;
- make sure you have `gcc/g++/make` installed or install them.

2.3 About Installer

- Linux : the installer is a self-extractable archive.
  - run the following command (depending on your os version):
    ```
    #> sh MlxPlore-1.1.0-linux32.bin
    or
    #> sh MlxPlore-1.1.0-linux64.bin
    ```
  - you can specify the target installation directory by giving the path as argument
  - a directory containing MlxPlore will be created in the directory installation path
- Windows
  - copy the installer on your Desktop or in your windows temporary directory
  - Double click on the executable and follow the instructions.

2.4 Directory structure

The MlxPlore directory structure is divided in two parts:

- the software directory containing the MlxPlore program,
- the personal user directory containing the MlxPlore workspace and documentation

2.4.1 Installation directory

```
MlxPlore ..................................... MlxPlore root directory
└─ mlxplore110 ............................... MlxPlore version directory
   ├─ bin ..................................... tools directory
   ├─ config .................................. configuration files
   ├─ factory ................................. MlxTran C++ API
   ├─ lib ..................................... C++ library
   └─ tools ................................... external tool used by MlxPlore
```

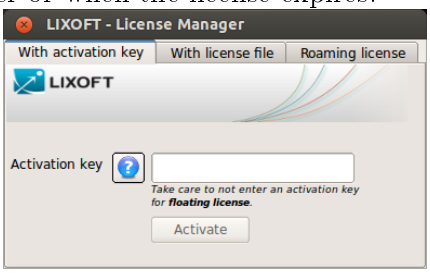
2.4.2 User directory

The user directory is created after the first launch of MlxPlore. This directory contains the basic configuration of MlxPlore, documentation, demos, log files, license file, ...

```
lixoft ....................................... lixoft data root directory
└─ mlxplore ................................... MlxPlore directory
   ├─ mlxplore110 ............................ MlxPlore version directory
   │  ├─ doc ................................ MlxPlore documentation
   │  ├─ log ................................ log files
   │  └─ tmp ................................ temporary directory
   ├─ license ................................ license directory
   └─ config ................................ configuration files
```

2.5 License

The activation key (provided by Lixoft) must be entered in the dialog box titled 'Lixoft license activation' ('With activation key' tab). This dialog box only appears when no license is available on the user's computer or when the license expires.



3 Running MlxPlore

3.1 Linux

To run MlxPlore under the Linux operating system, open a file browser, then browse the MlxPlore install folder and go to the `bin` directory. Click on the `mlxPloreRun.sh` executable file.

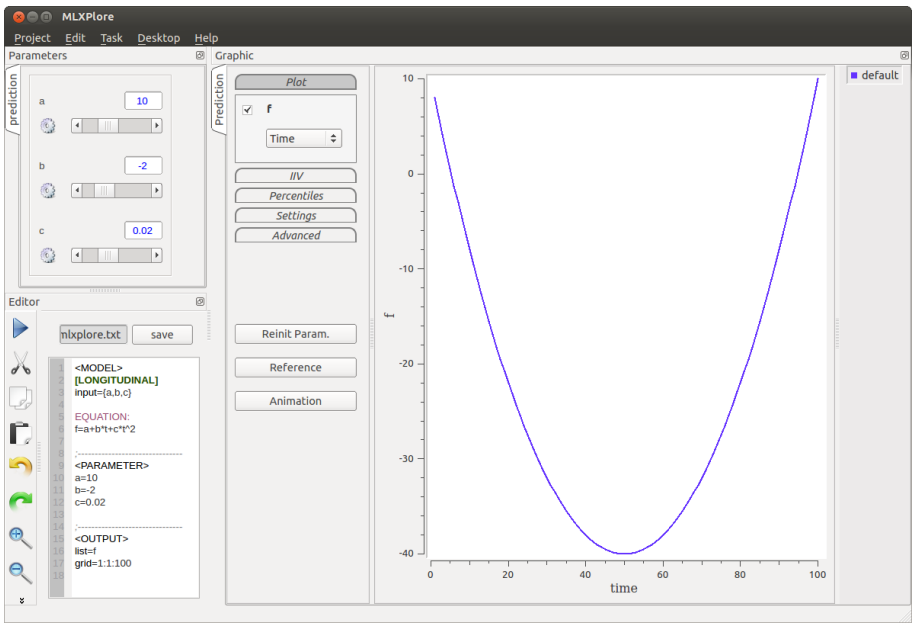
3.2 Windows

Under Windows operating system, go to the `Start` menu, go to the `Mlxplore` folder, then click on the MlxPlore icon.

4 Models

Models are written in the Mlxtran language. Mlxtran is a declarative, human-readable language describing simple and complex structural models as well as hierarchical statistical models. It supports explicit definition-based descriptions, as well as flexible equation-based descriptions. The following sections go through a series of examples to illustrate how to use a model description and explore the model.

4.1 Exploration of the structural model $f = a + bt + ct^2$



`model1b_mlxplore.txt` : $f = a + bt + ct^2$

4.1.1 Working material

- `model1a_mlxplore.txt`: the structural model is implemented using MLXTRAN in the file `model1_mlxtran.txt`.

  model1a_mlxplore

```
<MODEL>
file='model1_mlxtran.txt'

;-------------------------------
<PARAMETER>
a=10
b=-2
c=0.02

;-------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

  model1_mlxtran

```
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2
```

- `model1b_mlxplore.txt`: the same structural model is embedded in the MLXPLORE script.

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

;-------------------------------
<PARAMETER>
a=10
b=-2
c=0.02

;-------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

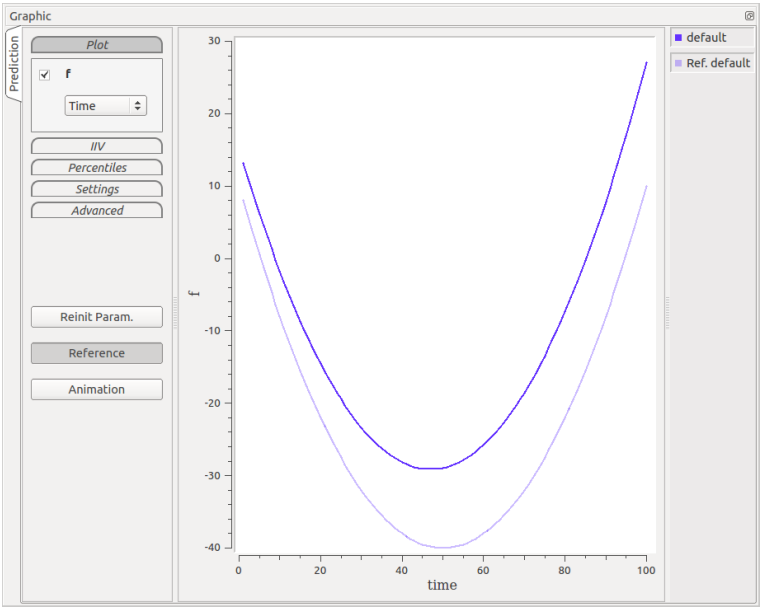
4.1.2 Model study

The parameters a, b, c of the model $f = a + b\,t + c\,t^2$ are modified through the corresponding sliders. The effect of a value change is immediately visible on the plot. The bounds of a slider can be modified with button  .

Button *Reinit. Param* resets the parameters a, b, c to the initial values define in section `<PARAMETER>`.

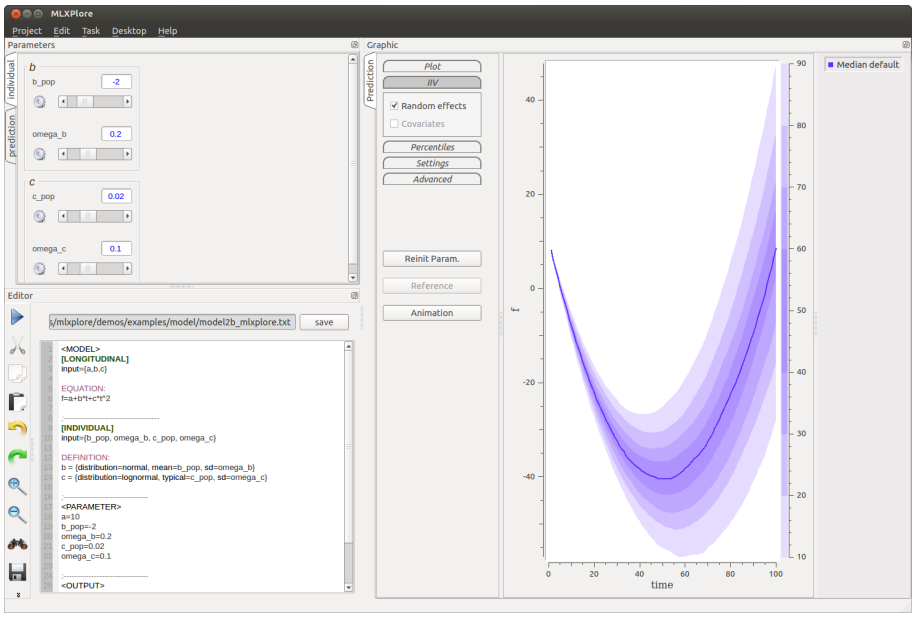
Button *Animation* animates the curve by changing the variable t of the studied model($f = a + b\,t + c\,t^2$).

MlxPlore allows to set a *reference* curve by clicking on button **Reference**.



Setting a reference curve with $a = 10, b = -2, c = 0.02$ and a 'current curve' with $a = 15.05, b = -1.88, c = 0.02$

4.2 Exploration of a the structural model $f = a + b\,t + c\,t^2$ where b is normally distributed and c is log-normally distributed



`model2b_mlxplore.txt`: $f = a + b\,t + c\,t^2$, where b is normally distributed and c is log-normally distributed

4.2.1 Working material

- `model2a_mlxplore.txt` : the model is implemented using MLXTRAN in the file `model2_mlxtran.txt`.

  model2a_mlxplore

```
<MODEL>
file='model2_mlxtran.txt'

;--------------------------------
<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1

;--------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

  model2_mlxtran

```
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

;-----------------------------------
[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c}

DEFINITION:
b = {distribution=normal, mean=b_pop, sd=omega_b}
c = {distribution=lognormal, typical=c_pop, sd=omega_c}
```

- `model2b_mlxplore.txt`: the same model is embedded in the MLXPLORE script.

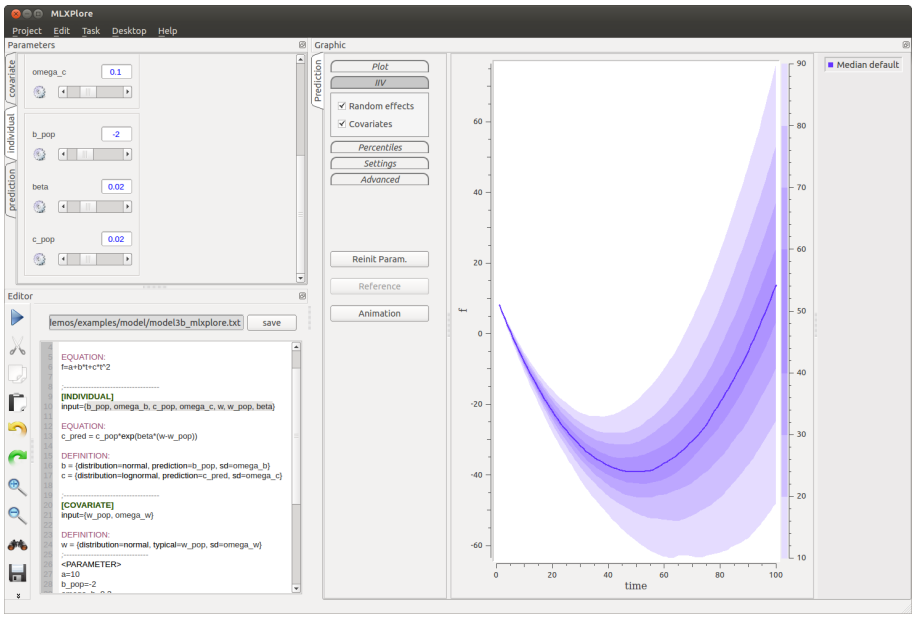
```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

;-----------------------------------
[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c}

DEFINITION:
b = {distribution=normal, mean=b_pop, sd=omega_b}
c = {distribution=lognormal, typical=c_pop, sd=omega_c}

;-------------------------------
<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1

;-------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

4.2.2 Model study

In this example, section `[INDIVIDUAL]` defines parameters b and c as respectively normally distributed and log-normally distributed.
Therefore $b \sim N(b_{pop}, omega_b^2)$ and $log(c) \sim N(c_{pop}, omega_c^2)$, the parameters b_{pop}, $omega_b$, c_{pop} and $omega_c$ are the inputs parameters for the section `[INDIVIDUAL]`.
a is the input parameter for the section `[LONGITUDINAL]` which can be modified through the interface. The other parameters (b and c) are computed using the definition of section `[INDIVIDUAL]`
On the MlxPlore interface, a new tab `individual` appears above the `prediction` tab. The actions on the sliders corresponding to the inputs parameters of the section `[INDIVIDUAL]` change the curve.

The checkbox *Random effect* allows to display the percentiles corresponding to the section `[INDIVIDUAL]`.

4.3 Exploration of a the structural model $f = a + bt + ct^2$ where c is function of a covariate w which is normally distributed



`model3b_mlxplore.txt`: $f = a + bt + ct^2$, where c is function of a covariate w which is normally distributed

4.3.1 Working material

- `model3a_mlxplore.txt`: the model is implemented using MLXTRAN in file `model3_mlxtran.txt`. **model3a_mlxplore**

```
<MODEL>
file='model3_mlxtran.txt'

;-------------------------------
<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

;-------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

model3_mlxtran

```
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

;-----------------------------------
[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

;-----------------------------------
[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}
```

- `model3b_mlxplore.txt`: the same joint model is embedded in the MlxPlore script.

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

;-----------------------------------
[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

;-----------------------------------
[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}
;-------------------------------
<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

;-------------------------------
<OUTPUT>
list=f
grid=1:1:100
```

4.3.2 Model study

In this example, section `[COVARIATE]` defines parameter w as normally distributed. Therefore $w \sim N(w_{pop}, omega_w^2)$ and the parameters w_{pop}, $omega_w$, are the inputs parameters of section `[COVARIATE]`. b_{pop}, $omega_b$, c_{pop}, $omega_c$, w, w_{pop} and $beta$ are the inputs parameter of section `[INDIVIDUAL]`. Here w is computed using the definition of section `[COVARIATE]`.

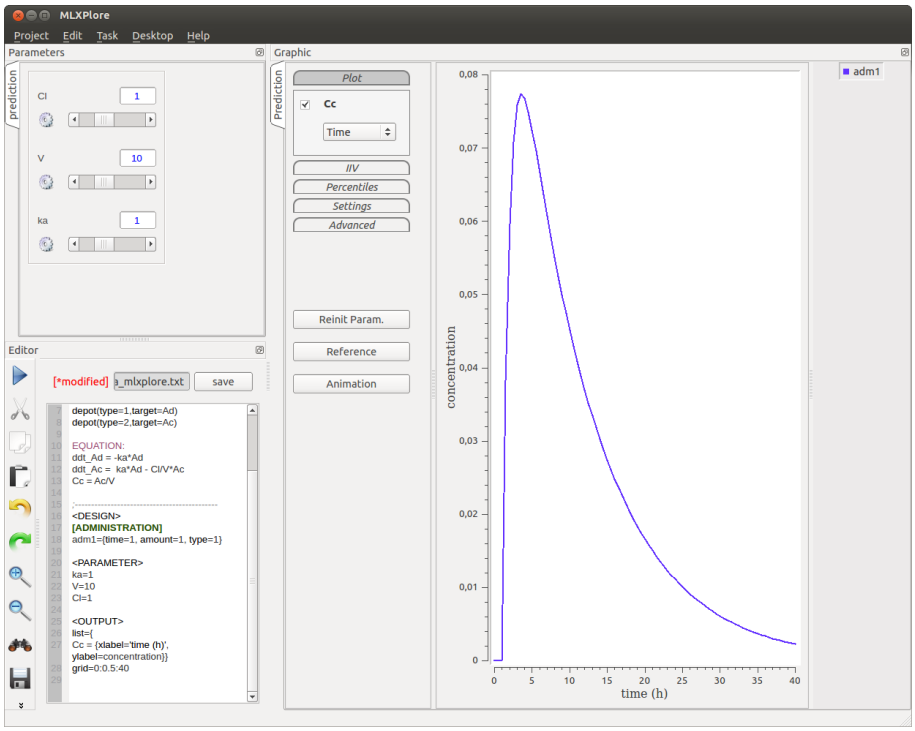
The other parameters b_{pop}, $omega_b$, c_{pop}, $omega_c$, w_{pop} are set (and can be modified) through the interface.
a is the input parameter of section `[LONGITUDINAL]` and can be modified through the interface. The other parameters (b and c) are computed using the definition in section `[INDIVIDUAL]`
On the MlxPlore interface, a new tab `covariate` appears. The actions on the sliders corresponding to the inputs parameters of the section `[COVARIATE]` are visible on the plot.

The checkbox *Random effect* and the checkbox *Covariate* allow to display the percentiles.

5 Administration and treatment design

MlxPlore allows to specify and try different dose regimens. The description of the administration design is written in section `<DESIGN>`. This section runs through a series of illustrative design and design comparison examples.

5.1 Single administration



`design1a_mlxplore.txt`: Single administration

5.1.1 Working material

- `design1a_mlxplore.txt`: single administration.

```
<MODEL>
[LONGITUDINAL]
input={ka, V, Cl}

PK:
depot(type=1,target=Ad)
depot(type=2,target=Ac)

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-------------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1, amount=1, type=1}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
list={
Cc = {xlabel=time (h), ylabel=concentration}}
grid=0:0.5:40
```

- `design1b_mlxplore.txt`: same model and same design. The target compartment is defined in the MlxPlore script using the reserved keyword `target`.

```
<MODEL>
[LONGITUDINAL]
input={ka, V, Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1, amount=1, target=Ad}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
list={
Cc = {xlabel=time (h), ylabel=concentration}}
grid=0:0.5:40
```

5.1.2 Design study

The section [ADMINISTRATION] describes a single administration :

```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={time=1, amount=1, type=1}
[...]
```

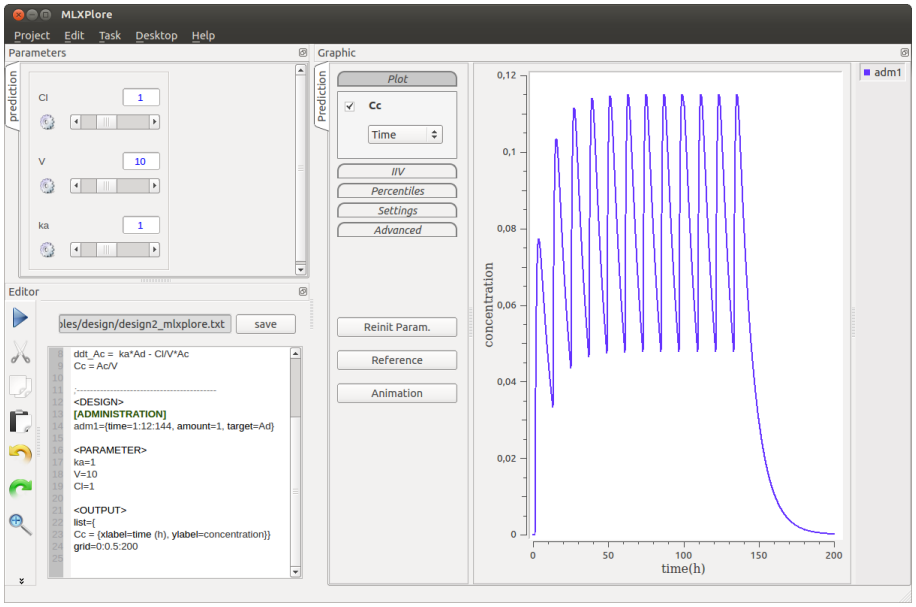
Design administration

At $time = 1$ the target Ad receives a dose of $amount = 1$.

```
[...]
PK:
depot(type=1,target=Ad)
[...]
```

Model : depot for administration where $type = 1$

5.2 Dose every 12 hours



`design2_mlxplore.txt`: Dose every 12 hours

5.2.1 Working material

`design2_mlxplore.txt`: dose every 12 hours

```
  <MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1:12:144, amount=1, target=Ad}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
list={
Cc = {xlabel=time (h), ylabel=concentration}}
grid=0:0.5:200
```

5.2.2 Design study

The section [ADMINISTRATION] contains the description of a dose administration every 12 hours:

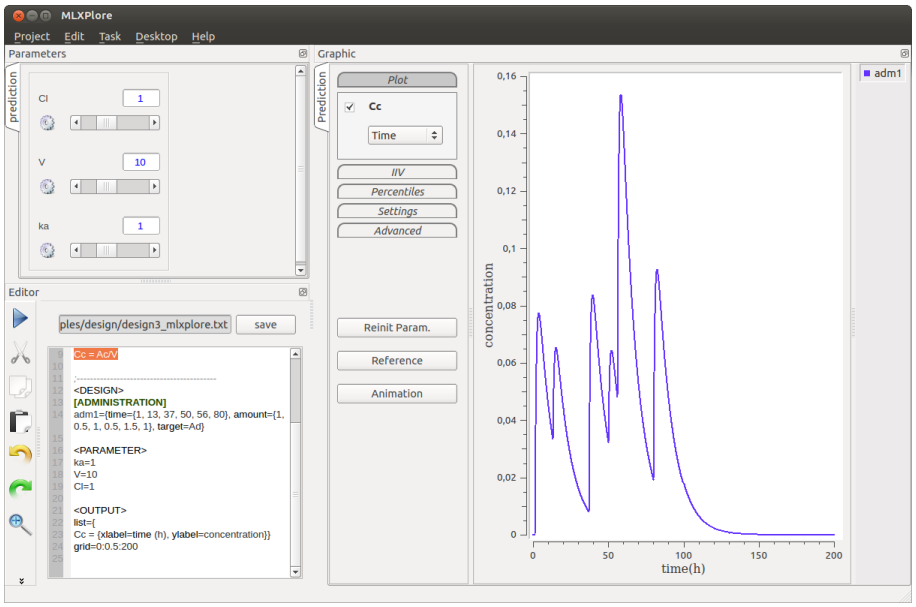
```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
        time=1:12:144,
        amount=1,
        target=Ad
      }
[...]
```

```
[...]
EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V
[...]
```

Model : depot for administration where $target = Ad$

Design administration
At times $(1, 13, 25, ...133)$ (times are processed using the loop syntax $< starttime >:< step >:< endtime >$) the target Ad receives a dose of $amount = 1$.

5.3 Different doses at different times



`design3_mlxplore.txt`: Different doses at different times

5.3.1 Working material

`design3_mlxplore.txt`: different doses at different times

```
<MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={
       time={1, 13, 37, 50, 56, 80},
       amount={1, 0.5, 1, 0.5, 1.5, 1},
       target=Ad
     }

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
list={
Cc = {xlabel=time (h), ylabel=concentration}}
grid=0:0.5:200
```

5.3.2 Design study

The section [ADMINISTRATION] contains the description of an administration with different doses at different times:

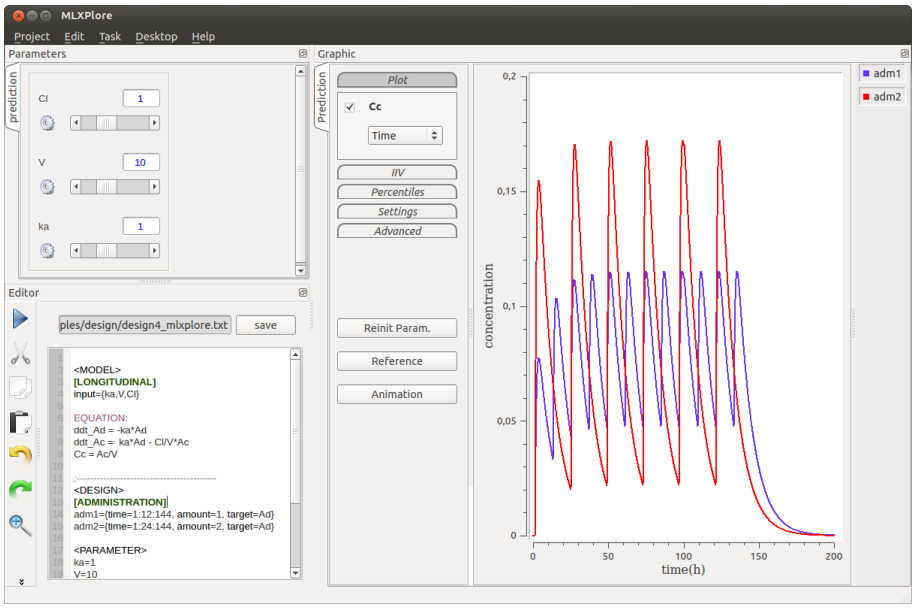
```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
      time={
              1, 13, 37,
              50, 56, 80
            },
      amount={
                1, 0.5, 1,
                0.5, 1.5, 1
              },
      target=Ad
     }
[...]
```

```
[...]
EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V
[...]
```

Model : depot for administration where $target = Ad$

Design administration
For times $(1, 13, 37, 50, 56, 80)$ the target Ad receives a dose of respectively $(1, 0.5, 1, 0.5, 1.5, 1)$.

5.4 Comparison of 2 dosing regimens



`design4_mlxplore.txt`: Comparison of 2 dosing regimens

5.4.1 Working material

`design4_mlxplore.txt`: comparison of 2 dosing regimens (oral administration).

```
  <MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1:12:144, amount=1, target=Ad}
adm2={time=1:24:144, amount=2, target=Ad}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
grid=0:0.5:200
list={
Cc = {xlabel=time (h), ylabel=concentration}}
```

5.4.2 Design study

The section [ADMINISTRATION] contains the description of an administration with different doses at different times:

```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
       time=1:12:144,
       amount=1,
       target=Ad}
adm2={
       time=1:24:144,
       amount=2,
       target=Ad
      }
[...]
```

Design administration

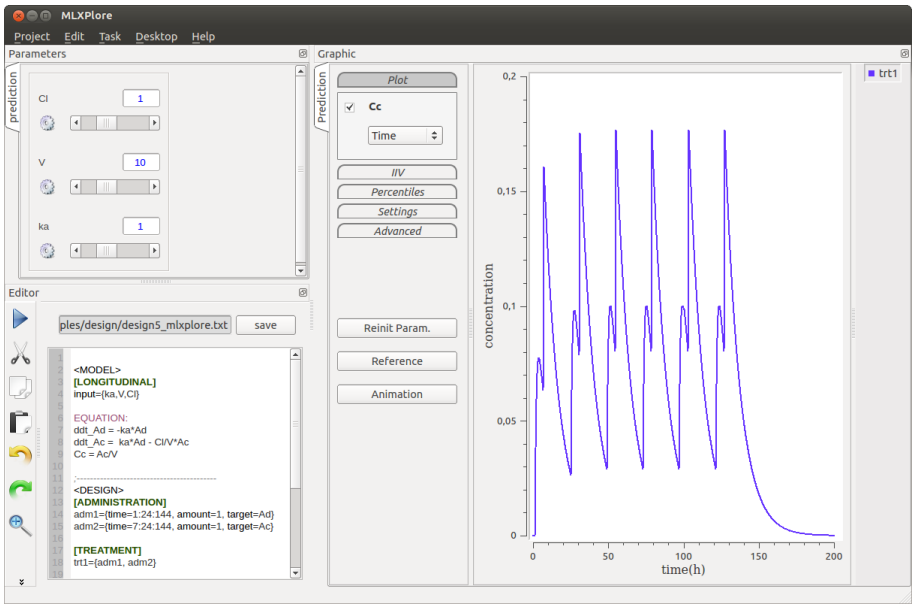
```
[...]
EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V
[...]
```

Model : depot for administration where $target = Ad$

The administration design allows to represent the dosing regimens $adm1$ and $adm2$. The curves

$adm1$ and $adm2$ are displayed on the same graphic that allows to compare both types of administrations.

5.5 Definition of a treatment as the combination of oral and IV administrations



`design5_mlxplore.txt`: Definition of a treatment as the combination of oral and IV administrations

5.5.1 Working material

`design5_mlxplore.txt`: definition of a treatment as the combination of oral and IV administrations.

```
  <MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1:24:144, amount=1, target=Ad}
adm2={time=7:24:144, amount=1, target=Ac}

[TREATMENT]
trt1={adm1, adm2}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
grid=0:0.5:200
list={
Cc = {xlabel=time (h), ylabel=concentration}}
```

5.5.2 Design study

The section [ADMINISTRATION] contains the description of an administration with different doses at different times. The section [TREATMENT] contains the description of the combination of oral and IV administrations:

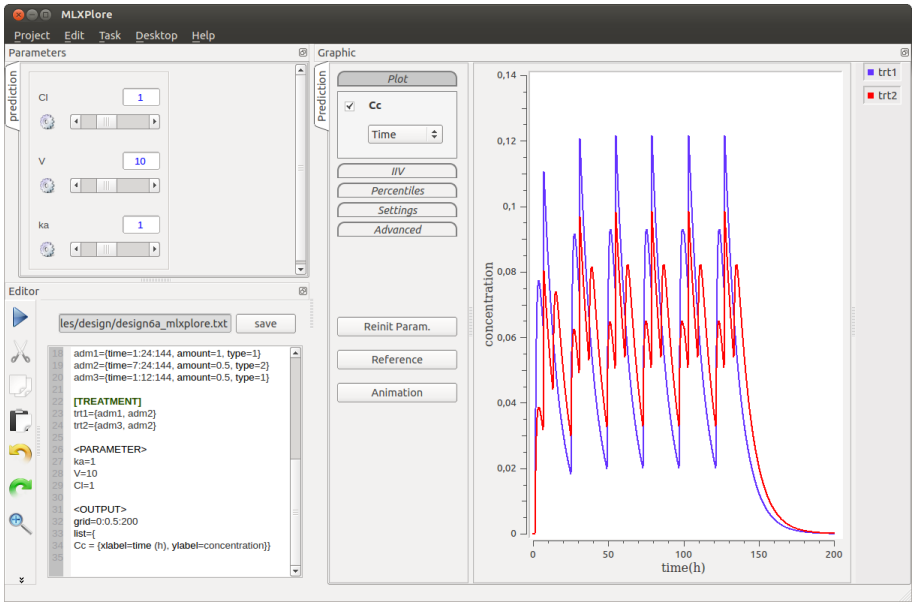
```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
        time=1:12:144,
        amount=1,
        target=Ad
      }
adm2={
        time=1:24:144,
        amount=2,
        target=Ac
       }
[TREATMENT]
trt1={adm1, adm2}
[...]
```

```
[...]
EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V
[...]
```

Model : depot for administration where $target = Ad$ for the administration $adm1$ and $target = Ac$ for the administration $adm2$

Design administration
This example introduces a new section `TREATMENT` on which the administrations $adm1$ and $adm2$ are combined. Therefore the plot associated to treatment $trt1$ integrates the informations of dose regimens $adm1$ and $adm2$.

5.6 Comparison of 2 treatments defined as different combinations of oral and IV administrations



`design6a_mlxplore.txt`: Comparison of 2 treatments defined as different combinations of oral and IV administrations

5.6.1 Working material

- `design6a_mlxplore.txt`: comparison of 2 treatments defined as different combinations of oral and IV administrations, using the reserved keyword `type`. Here, `type=1` for oral and `type=2` for IV).

```
<MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

PK:
depot(type=1,target=Ad)
depot(type=2,target=Ac)

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1:24:144, amount=1, type=1}
adm2={time=7:24:144, amount=0.5, type=2}
adm3={time=1:12:144, amount=0.5, type=1}

[TREATMENT]
trt1={adm1, adm2}
trt2={adm3, adm2}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
grid=0:0.5:200
list={
Cc = {xlabel=time (h), ylabel=concentration}}
```

- `design6b_mlxplore.txt`: same model and same design, using the reserved keyword `target`. Here, `target=Ad` for oral and `target=Ac` for IV.

```
<MODEL>
[LONGITUDINAL]
input={ka,V,Cl}

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V

;-----------------------------------------
<DESIGN>
[ADMINISTRATION]
adm1={time=1:24:144, amount=1, target=Ad}
adm2={time=7:24:144, amount=0.5, target=Ac}
adm3={time=1:12:144, amount=0.5, target=Ad}

[TREATMENT]
trt1={adm1, adm2}
trt2={adm3, adm2}

<PARAMETER>
ka=1
V=10
Cl=1

<OUTPUT>
grid=0:0.5:200
list={
Cc = {xlabel=time (h), ylabel=concentration}}
```

5.6.2 Design study

The section [ADMINISTRATION] contains the description of an administration with different doses at different times. The section [TREATMENT] contains the description of the combination of oral and IV administrations:

```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
      time=1:24:144,
      amount=1,
      type=1
     }
adm2={
      time=7:24:144,
      amount=0.5,
      type=2
     }
adm3={
      time=1:12:144,
      amount=0.5,
      type=1
     }

[TREATMENT]
trt1={adm1, adm2}
trt2={adm3, adm2}
[...]
```

```
[...]
PK:
depot(type=1,target=Ad)
depot(type=2,target=Ac)

EQUATION:
ddt_Ad = -ka*Ad
ddt_Ac =  ka*Ad - Cl/V*Ac
Cc = Ac/V
[...]
```

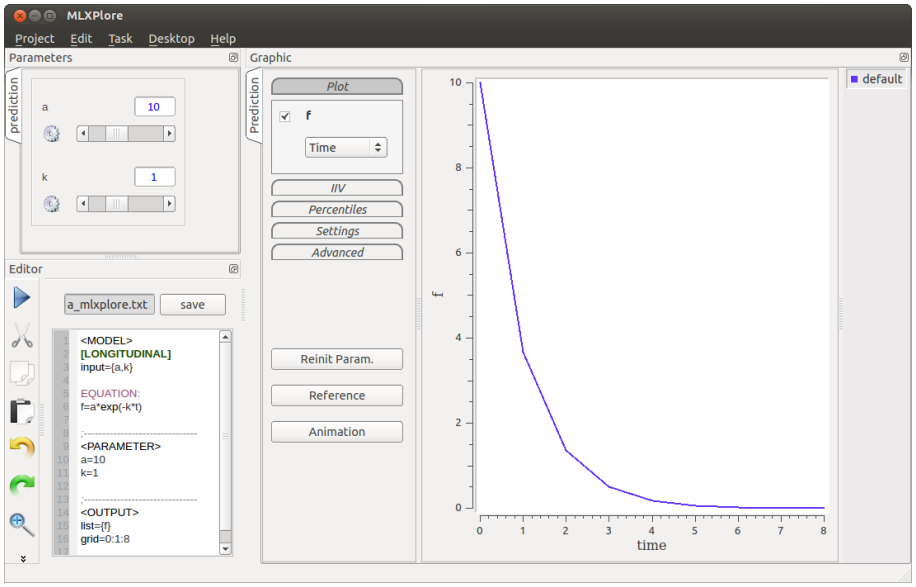
Model : depot for administration where $target = Ad$ for administrations $adm1$, $adm2$ and $target = Ac$ for the administrations $adm2$

Design administration
The example proposes to define two treatments as combinations of administrations. The treatment $trt1$ combines administrations $adm1$ and $adm2$, the treament $trt2$ combines the administrations $adm3$ and $adm2$.

6 Output

The <OUTPUT> section describes the response of a structural model.

6.1 Simple model for single output and different time grids



`output1a_mlxplore.txt`: A simple model for a single output and a regular time grid

6.1.1 Working material

- `output1a_mlxplore.txt`: a simple model for a single output and a regular time grid.

```
<MODEL>
[LONGITUDINAL]
input={a,k}

EQUATION:
f=a*exp(-k*t)

;-------------------------------
<PARAMETER>
a=10
k=1

;-------------------------------
<OUTPUT>
list={f}
grid=0:1:8
```

- `output1b_mlxplore.txt`: same model, regular time grid with another time step.

```
<MODEL>
[LONGITUDINAL]
input={a,k}

EQUATION:
f=a*exp(-k*t)

;-------------------------------
<PARAMETER>
a=10
k=1

;-------------------------------
<OUTPUT>
list={f}
grid=0:0.1:8
```

- `output1c_mlxplore.txt`: same model, irregular time grid.

```
<MODEL>
[LONGITUDINAL]
input={a,k}

EQUATION:
f=a*exp(-k*t)

;-------------------------------
<PARAMETER>
a=10
k=1

;-------------------------------
<OUTPUT>
list={f}
grid={0,1,1.5,2,2.5,3,5,8}
```

6.1.2 Output description

The section `OUTPUT` contains the names of output values (as provided by the model section `LONGITUDINAL`).

```
[...]
<OUTPUT>
list={f}
grid=0:1:8
[...]
```

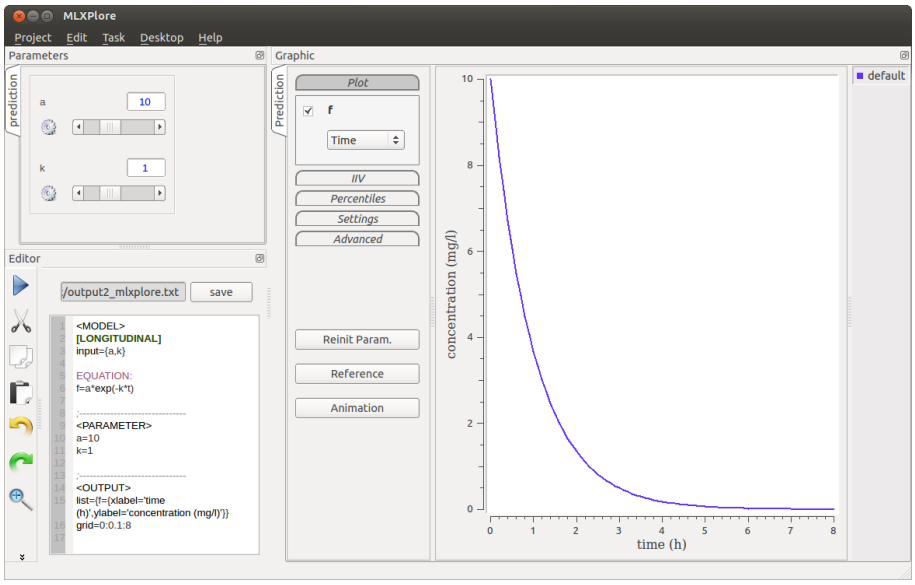
f is the ouput to plot

```
[...]
EQUATION:
f=a*exp(-k*t)
[...]
```

f is a function of inputs a, k and t.

This example illustrates the computation of function $f(a, k, t)$ (`list={f}`). a and k are defined through the interface. t has its values on grid $[0, 8]$ with step 1 (`grid=0:1:8`).

6.2 Definition of x-label and y-label



`output2_mlxplore.txt`: Definition of x-label and y-label

6.2.1 Working material

`output2_mlxplore.txt`: definition of x-label and y-label.

```
<MODEL>
[LONGITUDINAL]
input={a,k}

EQUATION:
f=a*exp(-k*t)

;-------------------------------
<PARAMETER>
a=10
k=1

;-------------------------------
<OUTPUT>
list={
      f={
         xlabel='time (h)',
         ylabel='concentration (mg/l)'
        }
     }
grid=0:0.1:8
```

6.2.2 Output description

Labels of plot axes are changed using keywords `xlabel` and `ylabel`

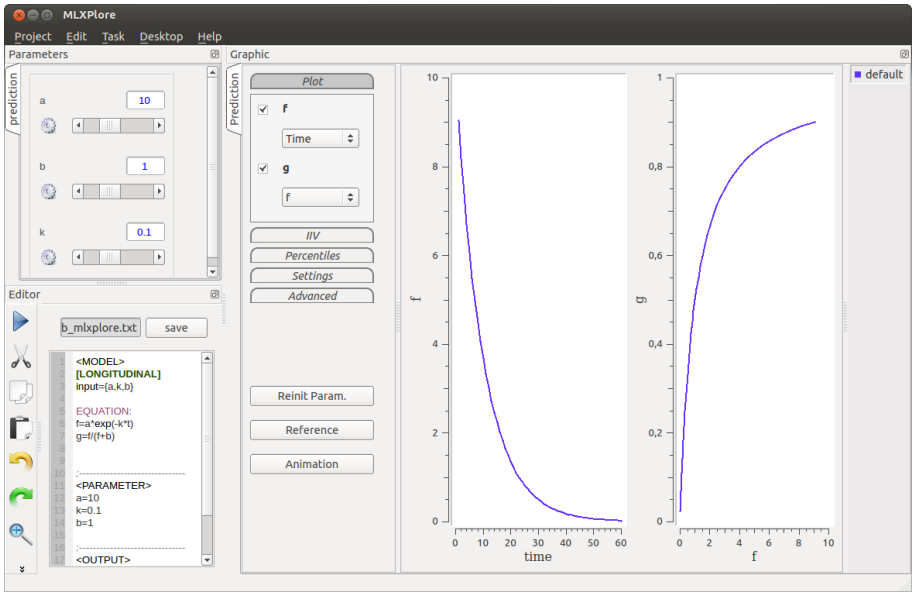
```
[...]
<OUTPUT>
list={
f={
 xlabel='time (h)',
 ylabel='concentration(mg/l)'
 }
}
grid=0:0.1:8
[...]
```

f is the ouput to plot. The label of x-axis is `time (h)`, the label of y-axis is `concentration (mg/l)`.

```
[...]
EQUATION:
f=a*exp(-k*t)
[...]
```

f is a function of inputs a, k and t.

6.3 A simple model for two outputs



`output3b_mlxplore.txt`: a simple model for two outputs (the second output is displayed as a function of the first output)

6.3.1 Working material

- `output3a_mlxplore.txt`: simple model for two outputs.

```
<MODEL>
[LONGITUDINAL]
input={a,k,b}

EQUATION:
f=a*exp(-k*t)
g=f/(f+b)


;-------------------------------
<PARAMETER>
a=10
k=0.1
b=1

;-------------------------------
<OUTPUT>
list={f,g}
grid=1:1:60
```

- `output3b_mlxplore.txt`: same model and same two outputs, the second output is displayed as a function of first output instead of a function of time.

```
<MODEL>
[LONGITUDINAL]
input={a,k,b}

EQUATION:
f=a*exp(-k*t)
g=f/(f+b)


;-------------------------------
<PARAMETER>
a=10
k=0.1
b=1

;-------------------------------
<OUTPUT>
list={f, g={xvar=f}}
grid=1:1:60
```

6.3.2 Output description

When several outputs are defined, it is possible to plot a given output as a function of another output with keyword `xvar`

```
[...]
EQUATION:
f=a*exp(-k*t)
g=f/(f+b)
[...]
<OUTPUT>
list={f, g={xvar=f}}
grid=1:1:60
[...]
```

f and g **are the ouputs to plot**. g **is displayed as a function of** f (`xvar=f`).

7 Settings

The section `SETTINGS` allows to configure the interface through a project.

Settings Example

```
[...]
<SETTINGS>
[GRAPHICS]
nb_simulations=1000
iiv_parameter=true
iiv_covariate=true
nb_band=100
pred_interval=80
```

7.1 Working material

- `setting1_mlxplore.txt`: the number of simulations used for computing the prediction intervals is set in the section `<SETTINGS>`.

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}

<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

<OUTPUT>
list=f
grid=1:1:100

<SETTINGS>
[GRAPHICS]
nb_simulations=1000
```

- `setting2a_mlxplore.txt` and `setting2b_mlxplore.txt`: the components of the IIV to display when the project is run are defined in the section `<SETTINGS>`
  settings2a_mlxplore

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}

<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

<OUTPUT>
list=f
grid=1:1:100

<SETTINGS>
[GRAPHICS]
nb_simulations=1000
iiv_parameter=true
iiv_covariate=false
```

settings2b_mlxplore

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}

<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

<OUTPUT>
list=f
grid=1:1:100

<SETTINGS>
[GRAPHICS]
nb_simulations=1000
iiv_parameter=false
iiv_covariate=true
```

- `setting3a_mlxplore.txt` and `setting3b_mlxplore.txt`: the number of prediction intervals and their levels are defined in the section `<SETTINGS>`.
  settings3a_mlxplore

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}

<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

<OUTPUT>
list=f
grid=1:1:100

<SETTINGS>
[GRAPHICS]
nb_simulations=1000
iiv_parameter=true
iiv_covariate=true
nb_band=4
pred_interval=80
```

settings3b_mlxplore

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2

[INDIVIDUAL]
input={b_pop, omega_b, c_pop, omega_c, w, w_pop, beta}

EQUATION:
c_pred = c_pop*exp(beta*(w-w_pop))

DEFINITION:
b = {distribution=normal, prediction=b_pop, sd=omega_b}
c = {distribution=lognormal, prediction=c_pred, sd=omega_c}

[COVARIATE]
input={w_pop, omega_w}

DEFINITION:
w = {distribution=normal, typical=w_pop, sd=omega_w}

<PARAMETER>
a=10
b_pop=-2
omega_b=0.2
c_pop=0.02
omega_c=0.1
w_pop=70
omega_w=10
beta=0.02

<OUTPUT>
list=f
grid=1:1:100

<SETTINGS>
[GRAPHICS]
nb_simulations=1000
iiv_parameter=true
iiv_covariate=true
nb_band=100
pred_interval=80
```

7.2 Keywords

- `nb_simulations`: number of simulations used to compute percentiles
- `iiv_parameter`: variability on individual parameters (checkbox *Random effect* is ticked when `iiv_parameter=true`.
- `iiv_covariate`: variability on covariate parameters (checkbox *Covariate* is ticked when `iiv_covariate=true`.
- `nb_band`: number of bands used to display the percentiles
- `pred_interval`: prediction interval

8 MlxPlore keywords

8.1 MODEL

The section <MODEL> specifies the Mlxtran model. There are two ways to define a model :

- using an external model file `file=<path to model file>`:
  Example:

```
<MODEL>
file='model1_mlxtran.txt'
```

- with an embedded Mlxtran model:
  Example:

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2
```

8.2 PARAMETER

The section PARAMETER gives the initial values of input parameters with the following syntax: `<parameter>` = <value>.
Example:

```
<PARAMETER>
a=10
b=-2
c=0.02
```

```
<MODEL>
[LONGITUDINAL]
input={a,b,c}

EQUATION:
f=a+b*t+c*t^2
```

The parameters a, b, c are defined as inputs of the model embedded in section `<MODEL>`.

8.3 OUTPUT

The `OUTPUT` section defines the outputs to plot.
Syntax of `OUTPUT` section:

```
list = {
          <output1>
                    [={
                        xlabel='<x-label name>',
                        ylabel='<y-label name>',
                        xvar=<another output>
                    }],
          <output2>,
          ...
          <outputn>
        }
```

The part within '[]' is optional.

```
grid=<start time>:<step>:<end time> | {t0, t1, ... , tn}
xlabel=<global x-label name>
ylabel=<global y-label name>
```

Note that the keywords `list` and `grid` are mandatories.

- Keyword **'list'**:
  The list of outputs is defined using the keyword `list`.

  Each output can be enriched with the name of x-axis and y-axis by using the following syntax:

  ```
  list={f={xlabel='label of x axis',ylabel='label of y axis'}}
  ```

  A given output can be represented according another output by using the keyword `xvar`

```
list={
       f={xlabel='label of x axis',ylabel='label of y axis'},
       g={ylabel='label of y axis', xvar=f}
     }
```

- Keyword **'grid'**:
  The time grid is defined with the keyword `grid` by using the following syntax:
  - `grid=<start time>:<step>:<end time>`
  - or `grid=`$\{t_0, t_1, ...t_n\}$
- Keyword **'xlabel'**:
  If the label of x-axis is the same on each output, the keyword `xlabel=<label of x axis>` sets all the labels of x axis with this same name.
  Example:

```
list={
       f,
       g
     }
xlabel='time(h)'
```

- Keyword **'ylabel'**: this keyword has the same behaviour than `xlabel` but it is applied on y-axis.

8.4 DESIGN

8.4.1 ADMINISTRATION

```
<administration name> = {
  type=<integer>,
  time=<value> | <start time>:<step>:<end time> | time={t_0, t_1, ... , t_n}
  amount = <value> | time={t_0, t_1, ... , t_n}
  target=<component target>,
  rate=<dose rate>
                        }
```

- Keyword **'type'** : dose compartment id
- Keyword **'target'** : target component
- Keyword **'time'** : time of the dose administration

- Keyword **'amount'** : amount of dose
- Keyword **'rate'** : dose rate

8.4.2 TREATMENT

A treatment combines a set of administrations by using the following syntax:

```
<treatment name = {
                    <adminstration name 1>,
                    <adminstration name 2> ,
                    ...
                   }
```

Example:

```
[...]
<DESIGN>
[ADMINISTRATION]
adm1={
       time=1:12:144,
       amount=1,
       target=Ad
     }
adm2={
       time=1:24:144,
       amount=2,
       target=Ac
      }
[TREATMENT]
trt1={adm1, adm2}
[...]
```

Design administration

8.5 SETTINGS

8.5.1 GRAPHICS

- `nb_simulations`: number of simulations used to compute percentiles
- `iiv_parameter`: variability on individual parameters (checkbox *Random effect* is ticked when `iiv_parameter=true`.
- `iiv_covariate`: variability on covariate parameters (checkbox *Covariate* is ticked when `iiv_covariate=true`.
- `nb_band`: number of bands used to display the percentiles
- `pred_interval`: prediction interval