AZRsim Documentation

AstraZeneca

Contents

User Guide **5**

1 Model Specification **7**
- 1.1 Model Name and Model Notes 7
- 1.2 Model States 8
- 1.3 Model Parameters 8
- 1.4 Model Variables 8
- 1.5 Model Reactions 9
- 1.6 Model Functions 9
- 1.7 Model Events 9

2 Modeling Functions **11**
- 2.1 create_model 11
- 2.2 simulate 11
- 2.3 plot 12
- 2.4 shiny_plot 12

3 Examples **13**
- 3.1 Michaelis–Menten 13
- 3.2 Two Compartment Model 13

User Guide

The user guide documents the functionality for the AZRsim R package which can be used to simulate ordinary differential equations in C and NONMEM.

In order to simulate the model in the **AZRsim** package we have to,

1. Create an appropriate text file representation of the model.
2. Compile the model with the `create_model()` function.
3. Simulate the compiled model with the `simulate()` function.

Chapter 1

Model Specification

In order to simulate the time varying variables of a system of ordinary differential equations you need to provide a specific representation of your mathematical model in a text file (a file with extension `.txt`). Every text file representation of a model must contain the following blocks presented exactly as defined below. Notice that each block name is preceded with ten `*`s. It is possible to comment out a line of code using `%`.

```
********** MODEL NAME

********** MODEL NOTES

********** MODEL STATES

********** MODEL PARAMETERS

********** MODEL VARIABLES

********** MODEL REACTIONS

********** MODEL FUNCTIONS

********** MODEL EVENTS
```

The remainder of this section deals with what each block is used for.

1.1 Model Name and Model Notes

`Model Name` and `Model Notes` are descriptive blocks that allow the user to specify a model name and any notes about the model. The name of the model will be presented when an `azrmod` object is printed to the console.

Below are the model name and model notes blocks for the simple harmonic oscillator example in the **AZRsim** package (`sho.txt`).

```
********** MODEL NAME

Harmonic Oscillator

********** MODEL NOTES
```

```
Harmonic Oscillator ODE example.
```

Here is what the associated `azrmod` object looks like after calling `create_model` on the text file.

```
sho <- create_model(system.file("examples/sho.txt", package="AZRsim"))
sho
AZRmodel
========
Name:                      Harmonic Oscillator
Number States:             2
Number Parameters:         1
Number Variables:          0
Number Reactions:          0
Number Functions:          0
```

1.2 Model States

The `Model States` block is where the user describes the system of differential equations followed by the initial conditions for each variable. The left hand side of each differential equation *must* be declared as `d/dt(x)` where `x` is a variable name chosen by the user.

The initial conditions *must* be defined, and they *must* be defined after the system of differential equations is defined. If variable `x` in the preceding paragraph had an initial condition value of 3 then we can declare `x(0) = 3` after defining the system of differential equations.

1.3 Model Parameters

The `Model Parameters` block is where the user defines the values of the parameters in the system of differential equations defined in `Model States`.

1.4 Model Variables

The `Model Variables` block provides a way to reparametrize parts of the model. Variables for intermediate calculations should be defined under `Model Variables`.

```
********** MODEL STATES

d/dt(A) = -R
d/dt(B) = R
A(0) = 1
B(0) = 0

********** MODEL PARAMETERS

k1 = 0.5

********** MODEL VARIABLES

R = k1*A
```

1.5 Model Reactions

The `Model Reactions` block provides a way to reparametrize parts of the model. Reaction rates should be defined under `Model Reactions`.

```
********** MODEL STATES

d/dt(A) = -R
d/dt(B) = R
A(0) = 1
B(0) = 0

********** MODEL PARAMETERS

k1 = 0.5

********** MODEL REACTIONS

R = k1*A
```

1.6 Model Functions

The `Model Functions` block can be used to define reoccuring calculations.

1.7 Model Events

The `Model Events` block can be used to define discrete state events.

There are essentially two parts to defining a discrete event in the model:

1. When does the event take place?
2. What is the event?

Below is an example of a event block.

```
********* MODEL EVENTS

event1 = eq(time, 1), y, y+40
event2 = eq(time, 2), y, y+40
```

The term `eq(time, 1)` pertains to (1). This says that the event occurs when the time step is strictly equal to 1. The term `y, y+40` pertains to (2). This is the event that is to take place. In this case the event involves variable `y` and we want to add 40 units of the drug this variable, `y+40`. (Note that events cannot take place at time 0.) In summary, the above block defines adding 40 to the value of `y` at time steps 1 and 2.

We can issue multiple events. The event defined below specifies resetting the value of `x` to 1 and `y` to 0 when `x` is equal to 0.1.

```
********* MODEL EVENTS

event = eq(x, 0.1), x, 1, y, 0
```

Below is a summary of the options that can be used to declare when an event takes place.

- `eq(a, 0)` implies variable a is strictly equal to 0.

- `lq(a, 0)` implies variable a is less than or equal to 0.
- `gq(a, 0)` implies variable a is greater than or equal to 0.
- `lt(a, 0)` implies variable a is strictly less than 0.
- `gt(a, 0)` implies variable a is strictly greater than 0.

Chapter 2

Modeling Functions

This section specifies the user-facing functions required to simulate the system of ordinary differential equations.

2.1 create_model

The `create_model` function is used to create the C code based on the text file representation of the model. This function returns and object of class `azrmod`. Once you create a model you can print a summary of the model in R. Below is an example of printing the simple harmonic oscillator, which is part of the **AZRsim** package.

```
sho <- create_model(system.file("examples/sho.txt", package="AZRsim"))
sho
AZRmodel
========
Name:                         Harmonic Oscillator
Number States:                2
Number Parameters:            1
Number Variables:             0
Number Reactions:             0
Number Functions:             0
```

In this model, the system of differential equations involves two states (y_1 and y_2) and one parameter (θ).

2.2 simulate

Once you have created an `azrmod` object with `create_model` you can use the `simulate` generic to simulate the state variables over a specified number of time steps. Note that the `simulate` function requires an object of class `azrmod`, and will return an object of class `azrsim` and `data.frame`.

In the simple harmonic oscillator example above we have two states, so calling `simulate(sho, seq(1, 100, by=0.1))` will provide a simulation of these two states over the time sequence 1 to 100 at 0.1 intervals.

2.3 plot

The `plot` generic can be called on an `azrsim` object and, by default, will construct a lattice plot of the first 9 parameters over time.

2.4 shiny_plot

The `shiny_plot` function must be called on an `azsim` object and will launch a shiny app that will allow the user to interactively include/exclude any state variable estimated.

Chapter 3

Examples

3.1 Michaelis–Menten

3.2 Two Compartment Model