HSimulator

User Manual

Contents

1 What you need to know before starting

HSimulator is an optimized, multithread Java simulator compatible with Java 6.0 or higher versions. The software works both as a standalone simulator and as a Javadoc documented Java library.

1.1 Introduction

HSimulator is compiled by means of Java 6.0. As any Java software, it is completely platform independent and it can be executed wherever it is installed a Java 6.0 runtime environment. This software has been tested on Windows XP, Windows 7, Windows 10, Linux and OS X.

1.1.1 Dependences of the software

In order to implement all the facilities provided by the software, a big part of the code is based on a set of functions that are distributed by other Java projects. The imported libraries are distributed together with HSimulator in the folder `lib` which needs to be placed in the same folder of the JAR archive of HSimulator. The imported Java libraries are:

- `JFreeChart`: a chart library to display professional quality charts.
- `Jxl`: a mature java API enabling developers to read, write, and modify Excel spreadsheets dynamically.
- `RSyntaxTextArea`: a syntax highlighting, code folding text editor for Java Swing applications.

1.2 How to launch

As HSimulator is a portable application there is no need to install it. The software works both in terminal and in a graphical environment. In the latter case a graphical user interface is automatically displayed.

- To launch HSimulator on Windows and OS X Environments:
  1. Double-click on the archive `HSimulator.jar`
  2. If, for any reason, this does not work, open the command prompt, move to the main project folder and type this command:

     ```
     java -jar HSimulator.jar
     ```

- To launch the application on Linux Environments:
  1. Open the terminal, move to the main project folder and type this command:

     ```
     java -jar HSimulator.jar
     ```

- To force the execution in terminal:
  1. Open a command prompt, move to the main project folder and type this command to display the help:

     ```
     java -jar HSimulator.jar -help
     ```

- To use HSimulator as a Java library we refer to the attached Javadoc documentation and to the example code `Gemcitabine.java`

1.3 Graphical user interface (GUI)

After having launched the application in a graphical environment, a graphical user interface will be automatically loaded (see Figure 1). The window contains, on the left, a syntax highlighting text editor, where you can type a new model specification or open an existing one. The right side contains, on the top, the compulsory options needed to run the simulation and on the bottom a space area where simulation charts will be displayed.

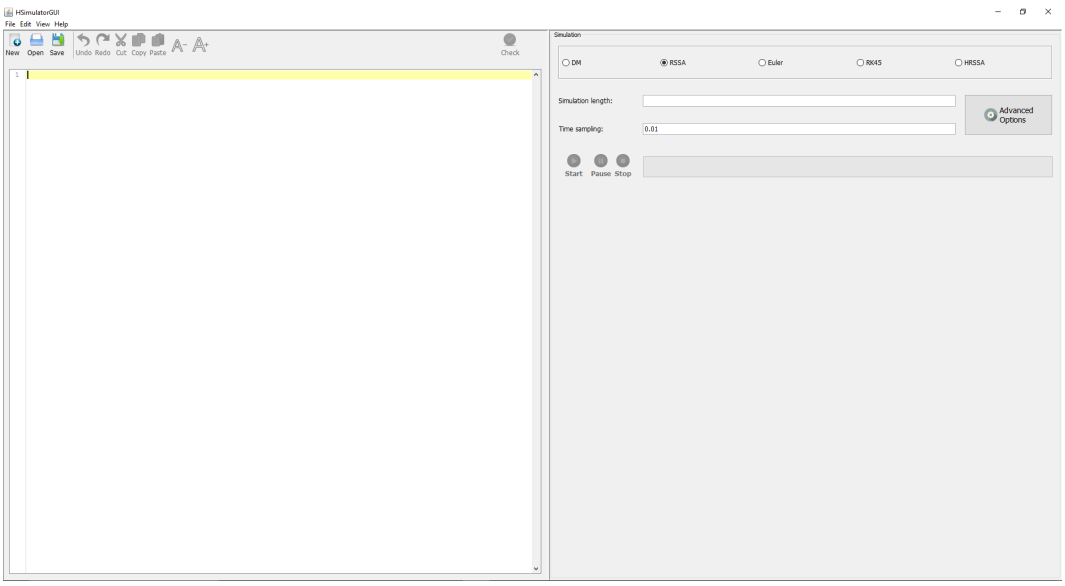

Figure 1: The HSimulator GUI.

The interface includes also a progress bar that indicates the status of the current simulation process and three buttons:

i `Start/Resume Simulation`: it starts or resumes the simulation. Enabled only when all parameters are set.

ii `Pause Simulation`: it pauses a simulation process before its completion. If pressed again, the process can be resumed.

iii `Stop Simulation`: stops a simulation before its completion.

1.4 How to run your first simulation

The software simulates biochemical reaction networks defined in a `txt` by means of a very simple syntax (see section 2 for details). Therefore, the first thing to do is to have a model specification loaded in the text editor. You can do this by typing the model description in the editor by opening an existing one stored as a `txt` file.

In this demonstrative example, we are going to pick up an example model by clicking on `File` → `Open model`. From here select the file `MAPK.txt`, which is included with the software.

Once you have the model specification loaded, in order to run the simulation you will only need to select a simulation algorithm (default value = HRSSA, see Section 3) and to provide a value for the parameter `simulation length` (default value = 1). All other parameters are already set by default, hence there is no need to worry about them for a first trial. Now that all the options are given, the `Start` button turns enabled and you can start your first simulation with HSimulator. Once the process is completed, a simulation chart will appear on the right-bottom side of the GUI (see Figure 2).

When the simulator is working in a terminal, you can obatin the same simulation of Figure 2 by the following command:

```
java -jar HSimulator.jar -alg=rssa -tMax=2 MAPK.txt
```

Simulation results will be automatically collected in an excel worksheet. If possible, the plot of the simulation can be generated by adding the argument `-doPlot`.

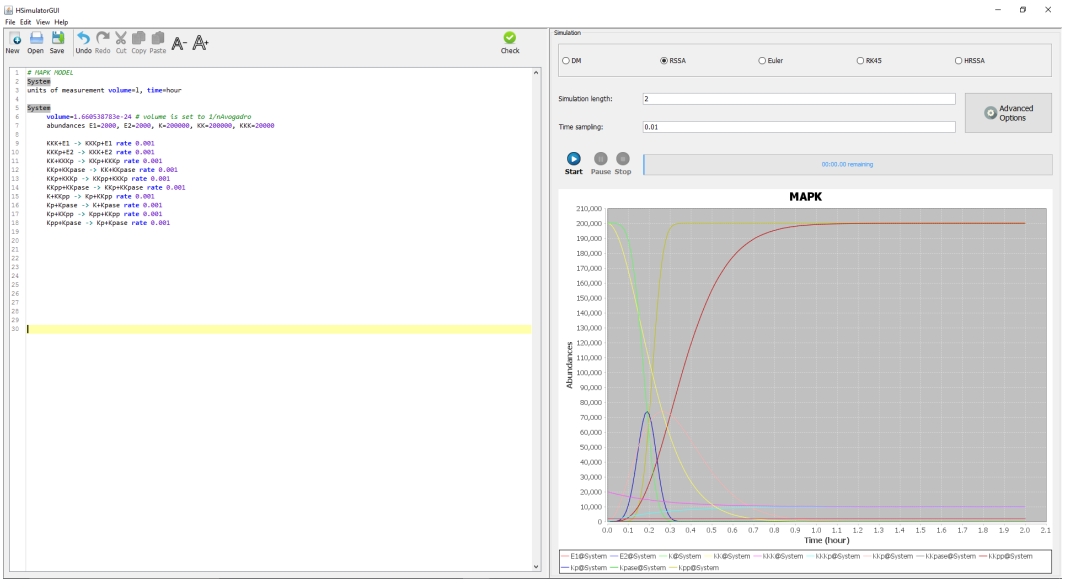

Figure 2: The HSimulator GUI showing a simulation.

2 Model specification

HSimulator accepts a textual description of the biochemical network to simulate, which can be also partitioned in several reaction volumes (multi-compartmental modeling). Each model reaction is defined according to standard arrow based notation and must be equipped with a mass-action stochastic reaction constant according to [2]. In Figure 3 we provide an example of biochemical reaction network with 5 volumes. The same specification is available in the model file `ModelSpecification.txt`.

The textual specification of the model starts with the definition of the reaction volumes (the first line is a comment, you can add comments with the hashtag character), which are indicated using round brackets to represent compartment inclusion (see also Figure 4). Immediately after the specification of the compartment hierarchy, the units of measurement of volume and time have to be specified:

```
units of measurement volume=ml (cm^3), time=min
```

These are global units, that is, all compartment volumes and simulation length have to be specified according to these units of measurement.

```
# This is an example of biochemical reaction network
System(PlasmaMembrane(Cytosol(Organelle,Nucleous)))
units of measurement volume=ml (cm^3), time=min

System
    volume=100
    abundances A=20, B=100

    B + A  -> B:A              rate 100.0
    B:A    -> B + A            rate 100.0
    2C + B -> D + 3B           rate 100.0
    E + F  -> G@Cytosol + F    rate 100.0
           -> H                rate 100.0
    D      ->                  rate 100.0

PlasmaMembrane
    volume=0.2

Cytosol
    volume=0.1

Nucleous
    volume=0.05
    abundances I=5, L=8

    I -> L:M                   rate 100.0

Organelle
    volume=0.05
```

Figure 3: An example of biochemical reaction network with 5 volumes. The same specification is available in the model file `ModelSpecification.txt`.

The textual specification continues with the definition of each single compartment. For each compartment we have to specify the volume size, the abundances of available species (not declared species are assumed to have zero abundance) and finally the list of reactions acting in the compartment.

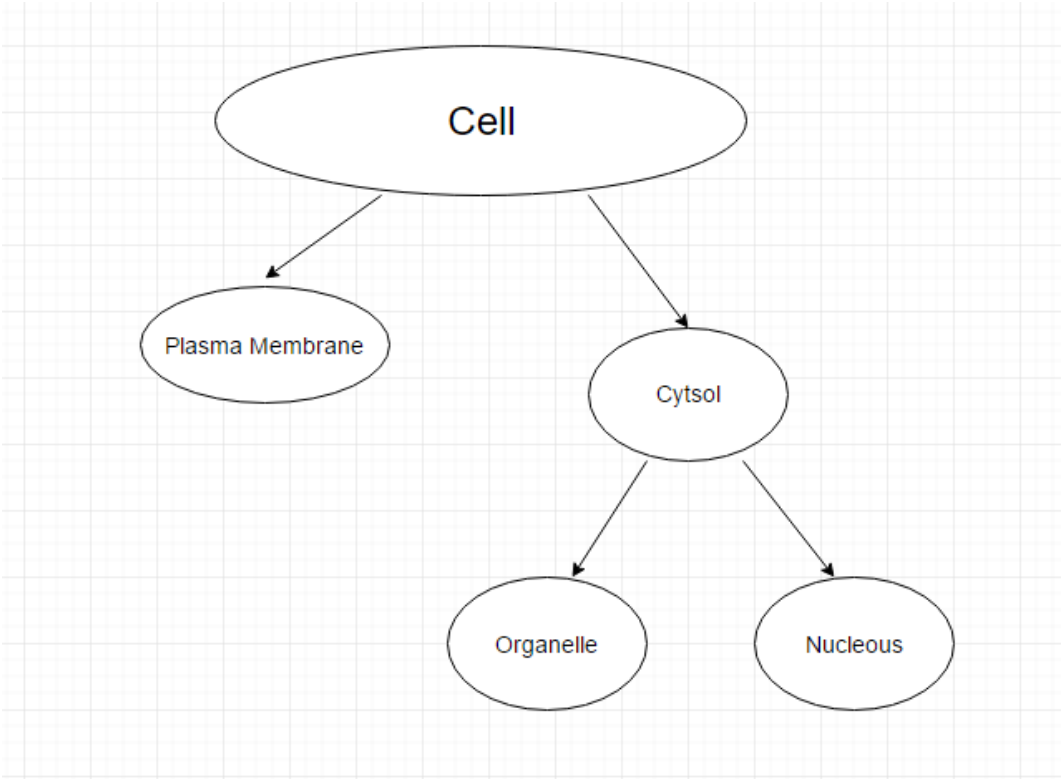

Figure 4: The hierarchy of compartments specified by the line of code `System(PlasmaMembrane(Cytosol(Organelle,Nucleous)))`.

```
System
    volume=100
    abundances A=20, B=100

    B + A  -> B:A              rate 100.0
    B:A    -> B + A            rate 100.0
    2C + B -> D + 3B           rate 100.0
    E + F  -> G@Cytosol + F    rate 100.0
           -> H                rate 100.0
    D      ->                  rate 100.0
```

Each reaction has to be equipped with a mass-action stochastic reaction constant according to [2]. In the definition of a reaction, you can use : to define complexes, `@` to indicate species location (`G@Cytosol` means species G in

compartment Cytosol) and implicit multiplication to indicate stoichiometric coefficients (`2C` means two molecules of species C).

Please note that you can check whether your model specification syntax is correct by clicking on the `Check` button of the GUI toolbar, or by clicking on `View` → `Check Model`. If an error occurs, an error dialog will be prompted displaying the message error.

2.1 Examples of model specification

To exemplify the way in which a reaction network can be textually specified in the simulator, in the following we will shortly present some biochemical models that are distributed with the software. We refer to [3] for a complete description of models and for simulation benchmarks.

2.1.1 Oregonator

Oregonator is a simple biochemical oscillator inspired by the phenomenon initially discovered by Belousov and Zhabotinsky. The model is important because it exhibits oscillating phenomena even if the system is far from the thermodynamic equilibrium. The Oregonator is of interest for both theoretical research and practical simulation.

```
# OREGONATOR MODEL
System
units of measurement volume=l, time=min

System
     volume=1.660538783e-24 # volume is set to 1/nAvogadro
     abundances x=500, y=1000, z=2100

     x + y -> rate 0.1
     y -> x rate 2
     x -> 2x + z rate 104
     2x -> rate 0.016
     z -> y rate 26


```

Figure 5: Oregonator model.

2.1.2 MAPK

The mitogen-activated protein kinase cascade (MAPK) describes a chain of proteins that cascade a signal from the cell receptor to its nucleus and result with a cellular response, i.e., cell proliferation, division and apoptosis. The MAPK pathway is stimulated when ligands, e.g., growth factors, bind to the receptor on the cell surface. The process is controlled through three main proteins kinases: MAPKKK, MAPKK and MAPK. The propagation of the stimulated signal is cascaded by sequential phosphorylation.

```
# MAPK MODEL
System
units of measurement volume=l, time=hour

System
     volume=1.660538783e-24 # volume is set to 1/nAvogadro
     abundances E1=2000, E2=2000, K=200000, KK=200000, KKK=20000

     KKK+E1 -> KKKp+E1 rate 0.001
     KKKp+E2 -> KKK+E2 rate 0.001
     KK+KKKp -> KKp+KKKp rate 0.001
     KKp+KKpase -> KK+KKpase rate 0.001
     KKp+KKKp -> KKpp+KKKp rate 0.001
     KKpp+KKpase -> KKp+KKpase rate 0.001
     K+KKpp -> Kp+KKpp rate 0.001
     Kp+Kpase -> K+Kpase rate 0.001
     Kp+KKpp -> Kpp+KKpp rate 0.001
     Kpp+Kpase -> Kp+Kpase rate 0.001


```

Figure 6: MAPK model.

2.1.3 Gemcitabine

Gemcitabine (2,2-difluorodeoxycytidine, dFdC) is an anti-cancer chemotherapy drug. It has been used to treat different types of cancer including non-small-cell lung cancer, pancreatic cancer, bladder cancer, and breast cancer. Gemcitabine produces clinical effects by incorporating its triphosphatemetabolite (dFdCTP) into DNA leading to the inhibition and blocking of DNA synthesis. The gemcitabine model has been developed to understand the mechanisms of resistance to gemcitabine efficacy. The model details mechanisms for the race between gemcitabine and natural nucleoside

triphosphate dCTP for DNA incorporation. It also includes the mechanisms of resistance by considering the role of ribonucleotide reductase (RR), deoxycytidine kinase (dCK) and human equilibrative nucleoside transporter1 (hENT1).

```
# GEMCITABINE MODEL
System
units of measurement volume=l, time=hour

System
     volume=1.660538783e-24 # volume is set to 1/nAvogadro
     abundances dFdC_out=100000, dCK=1000, RR=1000, dCMPD=1000, CDP=2000

     dFdC_out -> dFdC rate 9.97234
     dFdC -> dFdC_out rate 2.61675E-4
     dFdC_out -> dFdU rate 4.72336E-6
     dFdU -> dFdU_out rate 0.0508194
     dFdC -> dFdU rate 0.0
     dFdC+dCK -> dFdC_MP+dCK rate 0.00104994
     dFdC_MP -> dFdC rate 0.0875208
     dFdC_MP -> dFdC_DP rate 2.37162
     dFdC_DP -> dFdC_MP rate 0.212216
     dFdC_DP -> dFdC_TP rate 2.52037
     dFdC_TP -> dFdC_DP rate 1.44908
     dFdU+dCK -> dFdU_MP+dCK rate 9.68E-4
     dFdU_MP -> dFdU rate 5.60415E-5
     dFdU_MP -> dFdU_DP rate 0.07844
     dFdU_DP -> dFdU_MP rate 0.00420541
     dFdU_DP -> dFdU_TP rate 0.164322
     dFdU_TP -> dFdU_DP rate 9.05139E-5
     dFdC -> dFdU rate 4.76746E-4
     dFdC_MP+dCMPD -> dFdU_MP+dCMPD rate 4.559E-6
     -> CDP rate 1000.0
     CDP+RR -> dCDP+RR rate 0.05
     dCDP -> dCTP rate 25.2037
     dFdC_DP+RR -> dFdC_DP__RR rate 1.0E-5
     dFdC_DP__RR -> dFdC_DP+RR rate 0.1
     dCTP+dCK -> dCTP__dCK rate 1.0E-5
     dCTP__dCK -> dCTP+dCK rate 0.1
     dFdC_TP+dCMPD -> dFdC_TP__dCMPD rate 1.0E-7
     dFdC_TP__dCMPD -> dFdC_TP+dCMPD rate 0.1
     dFdC_TP -> dFdC_TP_DNA rate 0.0544456
     dFdU_TP -> dFdU_TP_DNA rate 7.37496E-4
     dCTP -> dCTP_DNA rate 0.5

```

Figure 7: Gemcitabine model.

3 Simulation

HSimulator provides an implementation of five state of the art simulation strategies covering exact stochastic simulation (DM and RSSA), deterministic simulation (Forward Euler and the Runge-Kutta-Fehlberg RK45 adaptive algorithm) and hybrid simulation (HRSSA). When deterministic or hybrid simulation is selected, the simulator automatically translates the reaction network into a set of mass-action ODEs.

The simulator GUI offers on the right side a panel devoted to the simulation process. From here you can decide the simulation algorithm and the simulation parameters (when the simulator is launched in terminal, we refer to the help printed with the argument `-help`).

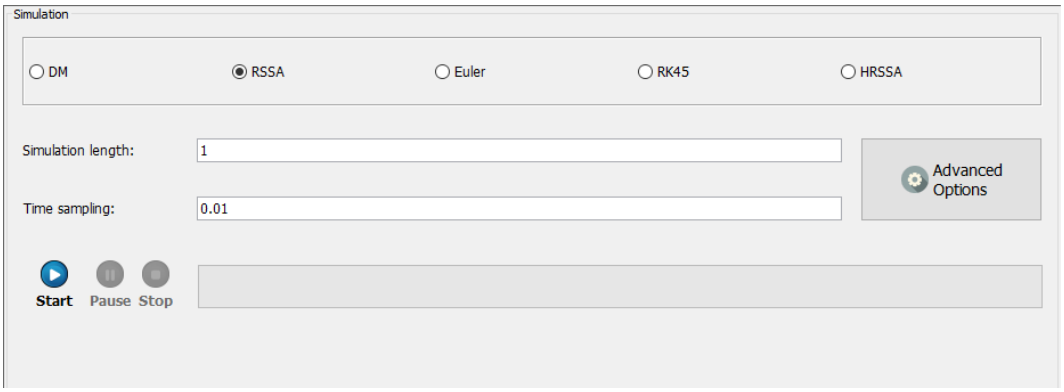

Figure 8: Panel dedicated to the simulation process.

The mandatory parameter `simulation length` indicates the temporal duration of the simulation. Figure 9 provides the dynamics of the Gemcitabine model with with simulation length of 24 hours. The parameter `time sampling` indicates the discretization time interval used to store the dynamics of the model. For instance, with a time sampling of 0.01, HSimulator will store the model dynamics at time steps of 0.01 units of time (instants 0, 0.01, 0.02, ...).

In Figures 10 and 11 it is possible to evaluate how plot changes in relation to the time sampling: larger values decrease the number of saved points and the plot will become less accurate. Please note that you can also decide

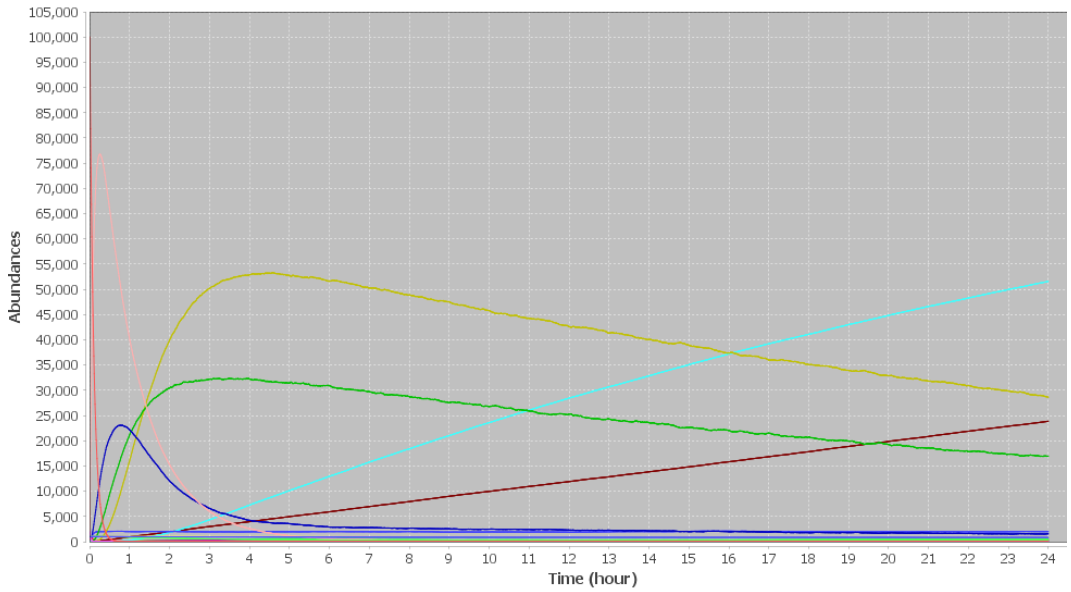

Figure 9: Gemcitabine dynamics with simulation length of 24 hours.

to view the plot in logarithmic scale by means of the `View` menu. It is important to stress that the time sampling does not affect the computation of the dynamics, but only the way in which the simulator stores computed time series.

3.1 Algorithms

The software allows to simulate a model according to different simulation algorithms. When you select an algorithm in the simulator GUI, you can have access to its settings by clicking the `Advanced Options` button (when the simulator is launched in terminal, we refer to the help printed with the argument `-help`).

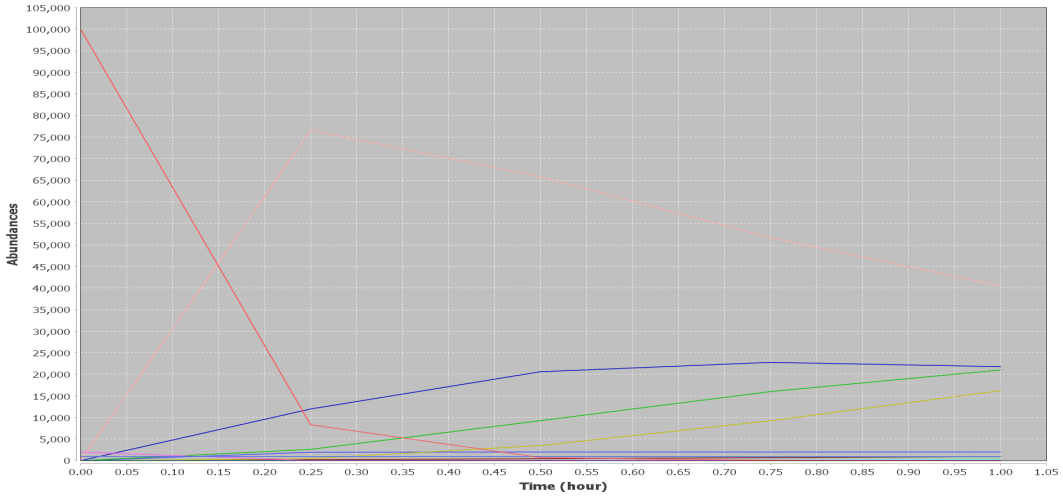

Figure 10: Gemcitabine dynamics with `simulation length` = 1 hour and `time sampling` = 0.25.

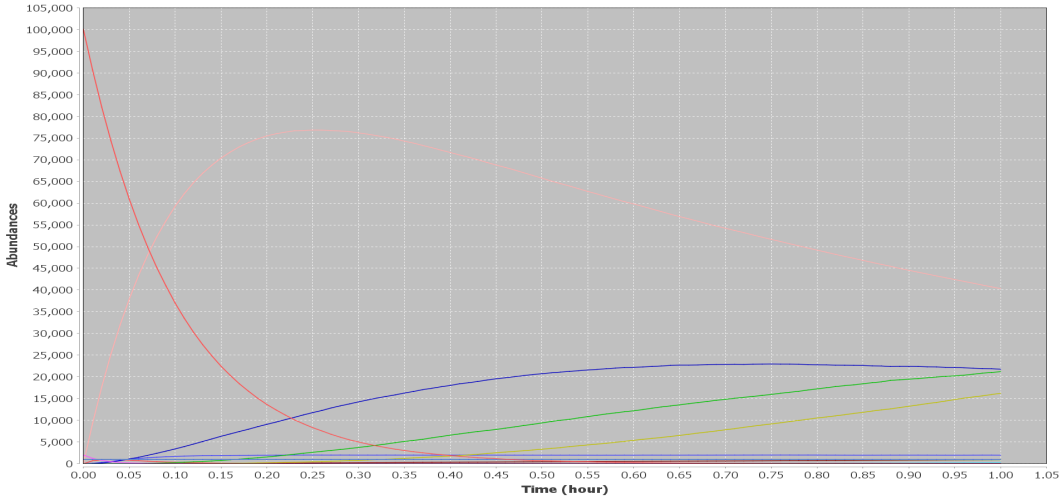

Figure 11: Gemcitabine dynamics with `simulation length` = 1 hour and `time sampling` = 0.01.

3.1.1 Direct Method (DM)

This is an implementation of the Gillespie Direct Method [2]. You do not need to specify any setting for running the simulation, however, the software

allows to compute more than one simulation in parallel for computing an averaged dynamics. The number of simulations is indicated by the parameter `number of simulations to compute`.

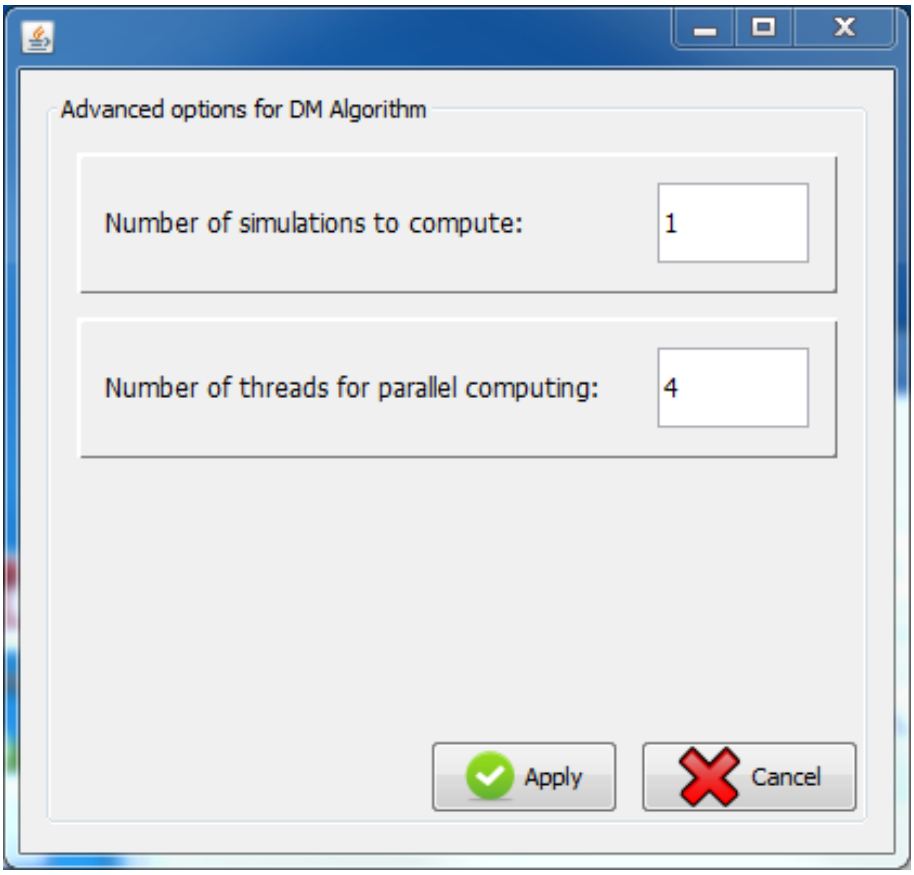

Figure 12: Advanced Options for Gillespie's DM

3.1.2 Rejection-based Stochastic Simulation Algorithm (RSSA)

RSSA [5, 6, 4, 7] is an exact stochastic algorithm equivalent to the Direct Method, which has an efficient way of computing reaction propensities. RSSA is currently the fastest algorithm available in literature for exact stochastic simulation. It has three parameters that can be set:

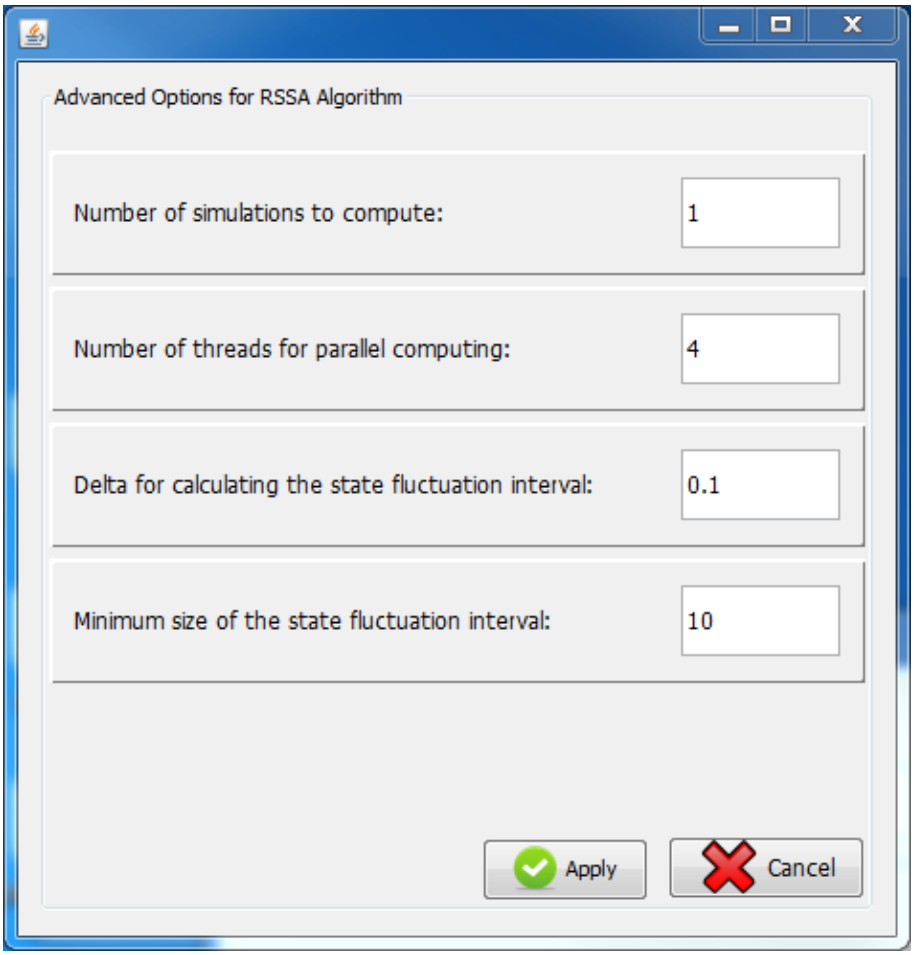

Figure 13: Advanced Options for RSSA.

`Number of simulation to compute`: number of simulations to compute in parallel to obtain the averaged dynamics of the system.

`Delta for calculating the state fluctuation interval`: this is the

value used to compute the fluctuation interval of the current system state X. Such an interval is computed by:

$$[X - \delta X, X + \delta X].$$

The size of the fluctuation interval affects the runtime of RSSA, but not the accuracy of the computation (RSSA is an exact stochastic algorithm). We refer to [5] for any insight into the matter.

`Minimum size of the state fluctuation interval`: this parameter will set the minimum admitted size of the fluctuation interval. If during the simulation that interval will be tighter that the size set by this parameter, the algorithm will enlarge the interval to fulfill the requirement. This parameter does not affect the accuracy of the simulation, but, like δ parameter, changes the runtime.

3.1.3 Euler Algorithm

This is a simple implementation of the forward Euler method for deterministic simulation [1]. This numerical algorithm is a first order method provided for didactic purposes. We recommend to use the RK45 algorithm for having more accurate simulations (see next section). It has one parameter that can be set:

`Discretization step`: this parameter specify the discretization step-size of the algorithm.

3.1.4 Runge-Kutta-Fehlberg algorithm (RK45)

RK45 is an adaptive numerical method that provides efficient deterministic simulation [1]. It has two parameters that can be set:

`Initial discretization step`: this parameter specify the discretization step-size adopted by the algorithm at the beginning of the simulation. This value will be then automatically adapted during the simulation to preserve the accuracy of the computation.

`Maximum error admitted`: threshold for the maximum truncation error admitted at each simulation step.

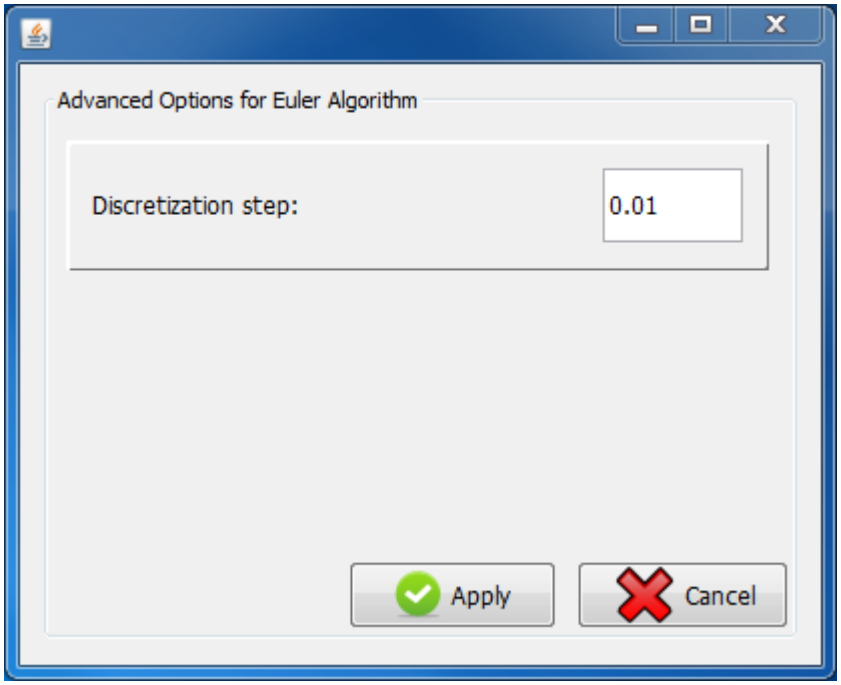

Figure 14: Advanced options for Euler Algorithm

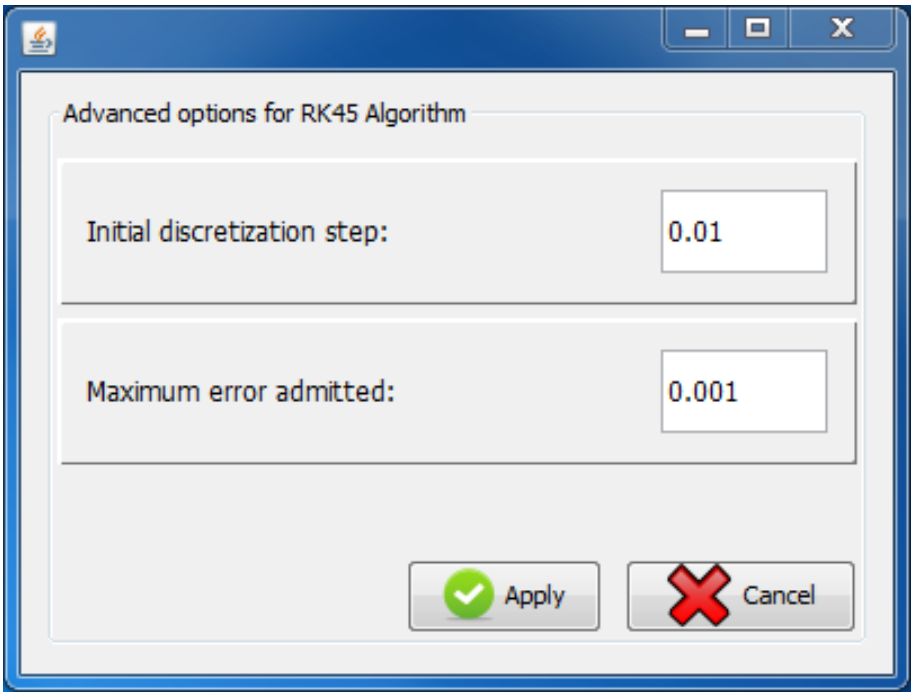

Figure 15: Advanced Options for RK45 Algorithm.

3.1.5 Hybrid Rejection-based Stochastic Simulation Algorithm (HRSSA)

HRSSA [3] is a new efficient hybrid stochastic/deterministic simulation algorithm for spatially homogeneous biochemical reaction networks. HRSSA allows fast simulation of large models by ensuring at same time the exactness of the simulation of a subset of the reaction network modeling slow reactions. This algorithm is built on top of RSSA and therefore some of its parameters are inherited from it:

`Number of simulation to compute`: number of simulations to compute in parallel to obtain the averaged dynamics of the system.

`Delta for calculating the state fluctuation interval`: this is the value used to compute the fluctuation interval of the current system state X. Such an interval is computed by:

$$[X - \delta X, X + \delta X].$$

The size of the fluctuation interval affects the algorithm runtime, we refer to [3] for any insight into the matter.

`Minimum size of the state fluctuation interval`: this parameter will set the minimum admitted size of the fluctuation interval. If during the simulation that interval will be tighter that the size set by this parameter, the algorithm will enlarge the interval to fulfill the requirement.

`Discretization step for simulating fast reactions`: this parameter specify the discretization step-size used for the deterministic simulation of fast reactions.

`Minimum molecule number of fast reactions`: the minimum amount of molecules that has to be available for fast reactions. This value is usually greater than 100.

`Minimum number of reaction firing of fast reactions`: the minimum number of times that a fast reaction has to be applied, in average, within the time range of size indicated by the parameter `discretization step for simulating fast reactions`. This value usually ranges between 5 and 10.

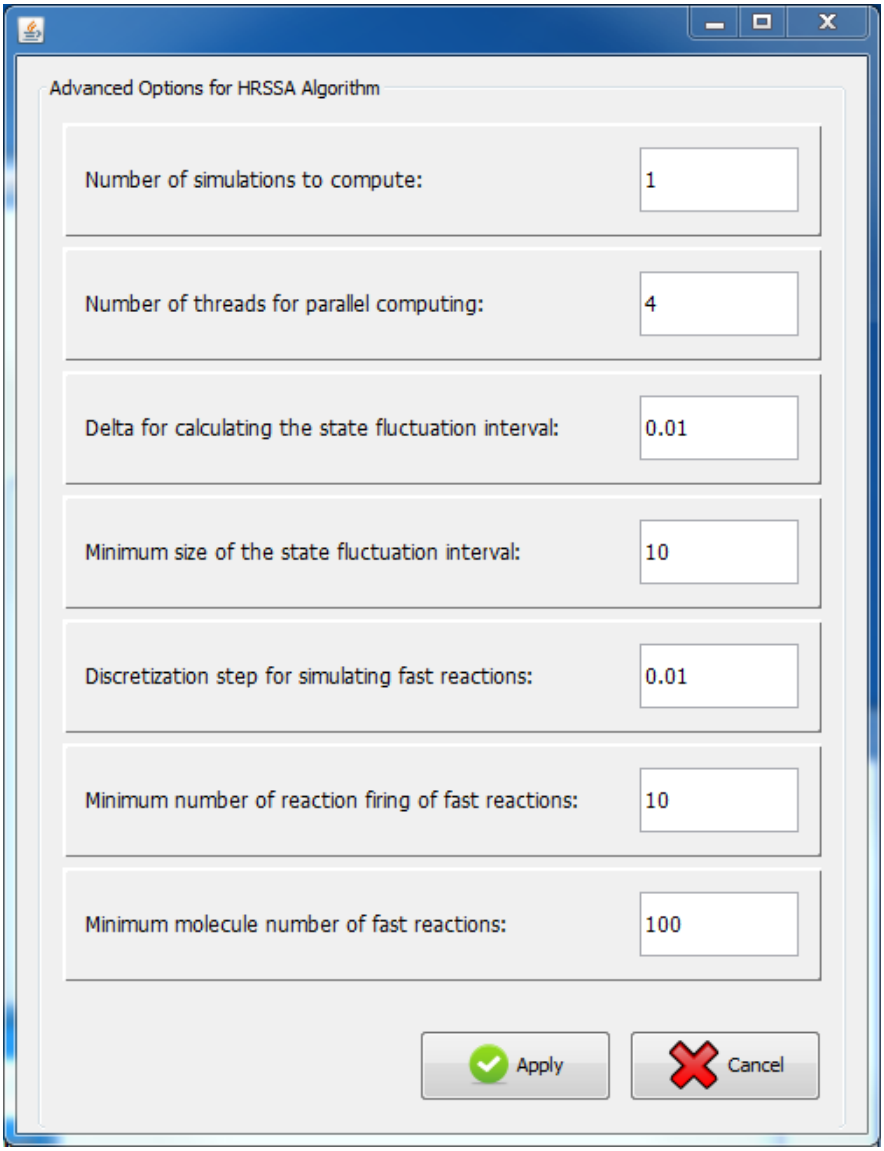

Figure 16: Advanced options for HRSSA

3.2 Evaluation of simulation results

Once a simulation process is completed, the simulator will automatically create an excel worksheet storing simulated time series when the simulation is executed in terminal. When the GUI is used, instead, a plot will automat-

ically appear in the bottom right part of the window. Moreover, there are some addtional actions you can do to evaluate simulation results:

Save Simulation Results: by clicking on `File` → `Save Simulation Results...` you can save the time series of the simulation as a *Microsoft Excel* .xls file or as a tab-separated file (.txt file). Note that if the time series is very long you will only be able to save it as .txt file.

Logarithmic scale plot: the software allows to plot time series in logarithmic scale. This action can be done by clicking on `View` → `Logarithmic Scale Plot`.

Print Model Simulation Plot: by clicking on `File` → `Print Model Simulation Plot`, you can print the simulation plot. The same action can be done by right-clicking on the chart and selecting `Print...`

Show plot in a separate window: in order to analyze better the chart, the software allows to create a new one in a separate window. This action can be done by double-clicking on the plot, or by selecting `View` → `Show Plot`.

References

[1] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations, 2nd Edition*. John Wiley & Sons, 2008.

[2] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[3] Luca Marchetti, Corrado Priami, and Vo Hong Thanh. HRSSA – Efficient hybrid stochastic simulation for spatially homogeneous biochemical reaction networks. *Journal of Computational Physics*, 317:301–317, 2016.

[4] V. H. Thanh and C. Priami. Simulation of biochemical reactions with time-dependent rates by the rejection-based algorithm. *The Journal of Chemical Physics*, 143:054104, 2015.

[5] V. H. Thanh, C. Priami, and R. Zunino. Efficient rejection-based simulation of biochemical reactions with stochastic noise and delays. *The Journal of Chemical Physics*, 141:134116, 2014.

[6] V. H. Thanh, R. Zunino, and C. Priami. On the rejection-based algorithm for simulation and analysis of large-scale reaction networks. *The Journal of Chemical Physics*, 142:244106, 2015.

[7] V. H. Thanh, R. Zunino, and C. Priami. Efficient constant-time complexity algorithm for stochastic simulation of large reaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, page (in press), 2016.