D-VASim

Dynamic Virtual Analyzer and Simulator

A quick start guide v1.0

February 25 2016

Department of Applied Mathematics and Computer Science
Section of Embedded Systems Engineering
Bio-Design Automation Group
Richard Petersens Plads, Lyngby 3200
Technical University of Denmark

http://bda.compute.dtu.dk

ese

BDA
Bio-Design
Automation

DTU

Contents

About D-VASim

D-VASim stands for Dynamic Virtual Analyzer and Simulator. It is developed to analyze and simulate the genetic logic circuit models developed in the Systems Biology Markup Language (SBML) [1]. Practically, D-VASim can be used to simulate any bio-model available in SBML L3v1 (Level 3 version 1) format; however, it is solely designed for the simulations of genetic logic circuits, where user can apply external signals on the model during runtime.

D-VASim is developed on graphical programming language platform called LabVIEW (Laboratory Virtual Instrument Engineering Workbench) [2]. Besides graphical programming, textual programming language, like JAVA, is also used to integrate JSBML [3] library in it.

In this short document, you will learn how to use D-VASim to perform virtual laboratory experiments. This requires you to have pre-synthesized bio-model, in the form of SBML file, you want to test. This SBML can be obtained by creating a bio-model using any other bio-tool and generating its SBML L3v1 format.

Stochastic simulation algorithm (SSA) has been implemented to perform stochastic simulations of SBML models. Furthermore, D-VASim is also capable of simulating the deterministic behavior of a bio model by solving ordinary differential equations.

Latest version of D-VASim can be downloaded from http://bda.compute.dtu.dk/downloads/. The example SBML models, included in the download package, are developed at University of Utah by Myer's group [4].

Conventions

The following conventions appear in this document:

💡 This icon denotes a tip, which notifies you to advisory information.

⚠ This icon denotes an alert, which notifies you to important information.

bold Bold text denotes items that you must select or click or enter the value in the software, such as open file option or running the simulation button or entering the value of simulation speed etc.

italic Italic text denotes the name of a folder or a file path.

bold and italic Bold and italic text denotes the name of a file.

1. Analyzing the SBML model using D-VASim

In this section, you will learn how to import and analyze the SBML model of a genetic circuit in D-VASim. In this document, we will use the SBML model of a genetic AND gate enclosed in the *Sample_SBML_Models* folder in the download package.

1.1. Launching D-VASim

The front interface of D-VASim, shown in Figure 1, will appear when you double-click on ***D-VASim.app*** (for MAC OS) or ***D-VASim.exe*** (for Windows OS).

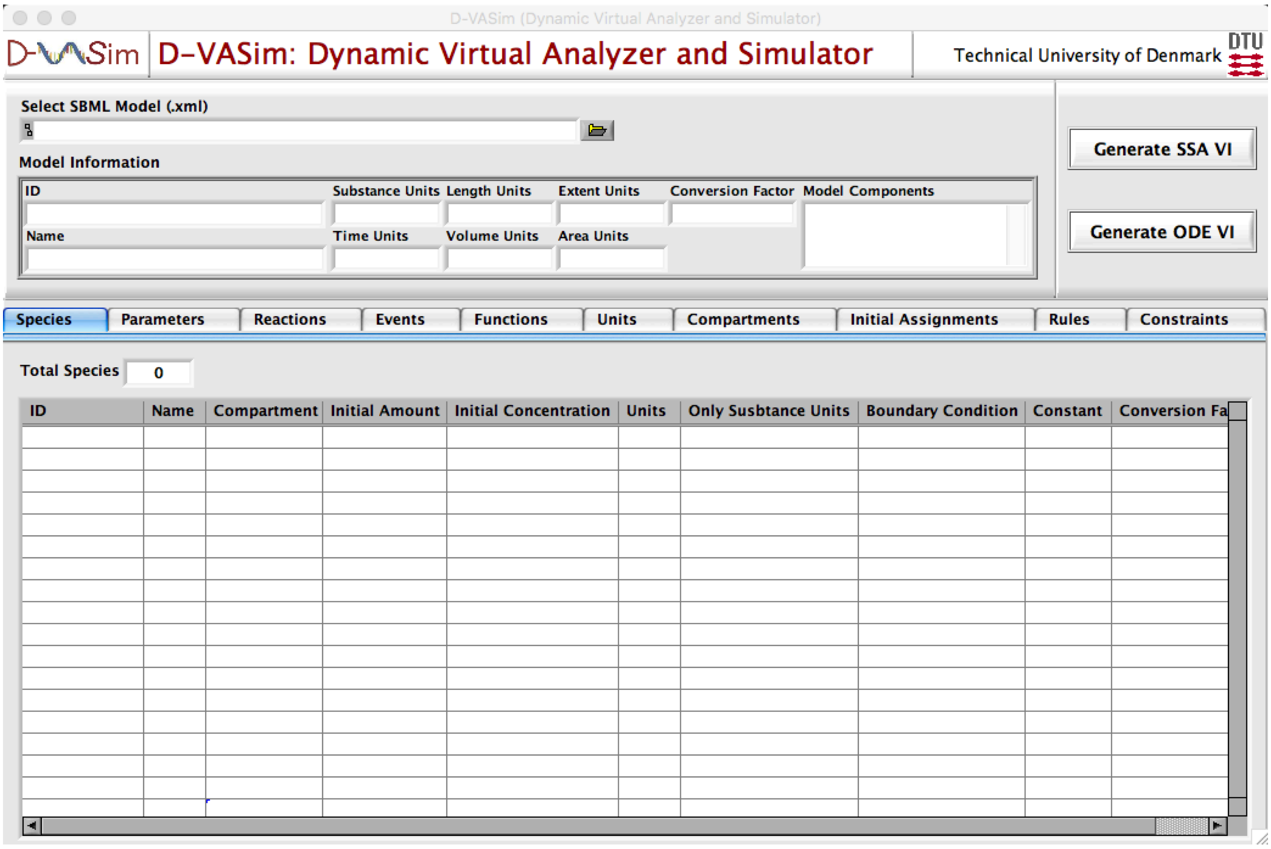
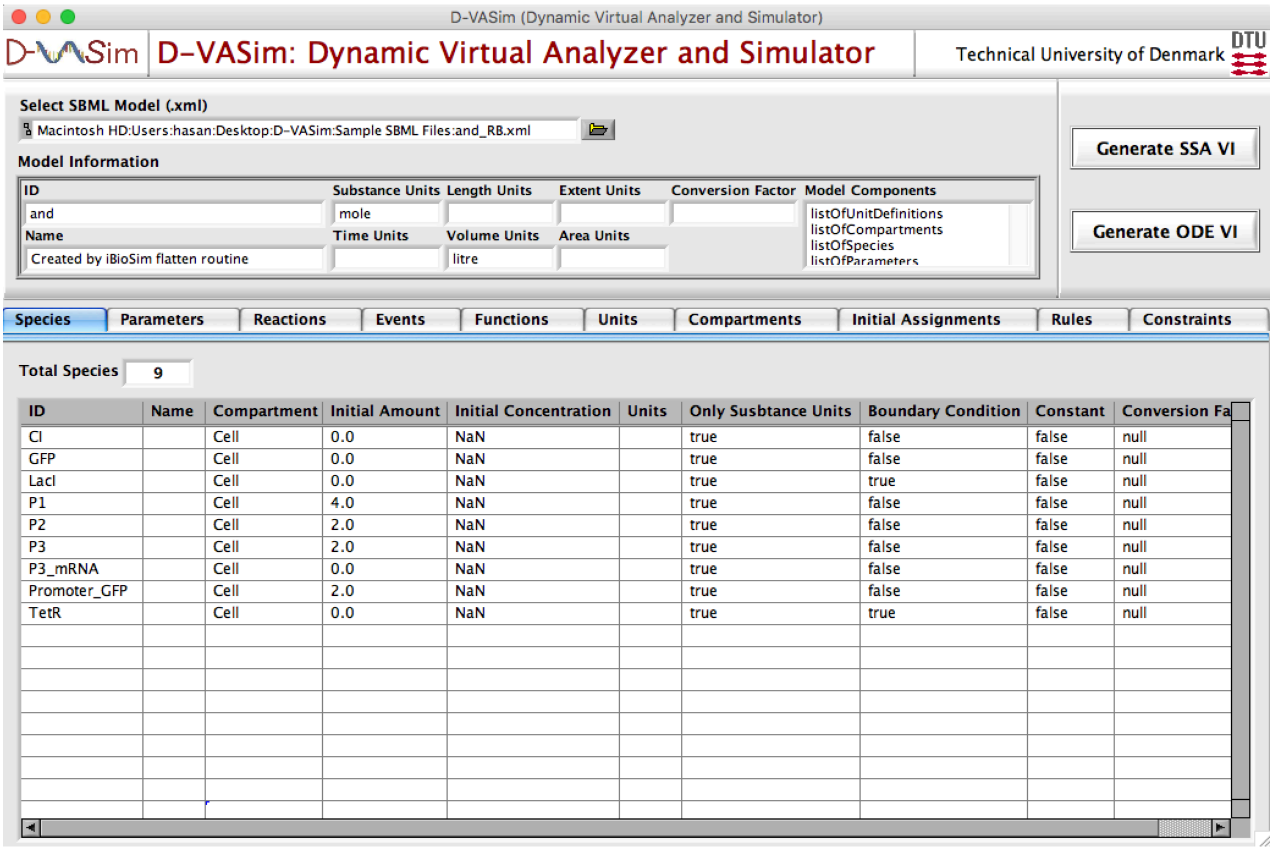


Figure 1. Front interface of D-VASim.

Complete the following steps to import the SBML model of a genetic AND gate circuit.

1. Click **open file** button on the option “Select SBML Model (.xml)”.
2. Navigate to the file ***and_RB.xml*** placed under the application directory*../D-VASim/Sample_SBML_Models/*.

This will run the JSBML library and display the SBML components as shown in Figure 2.



Select SBML Model (.xml)

Macintosh HD:Users:hasan:Desktop:D-VASim:Sample SBML Files:and_RB.xml

Model Information

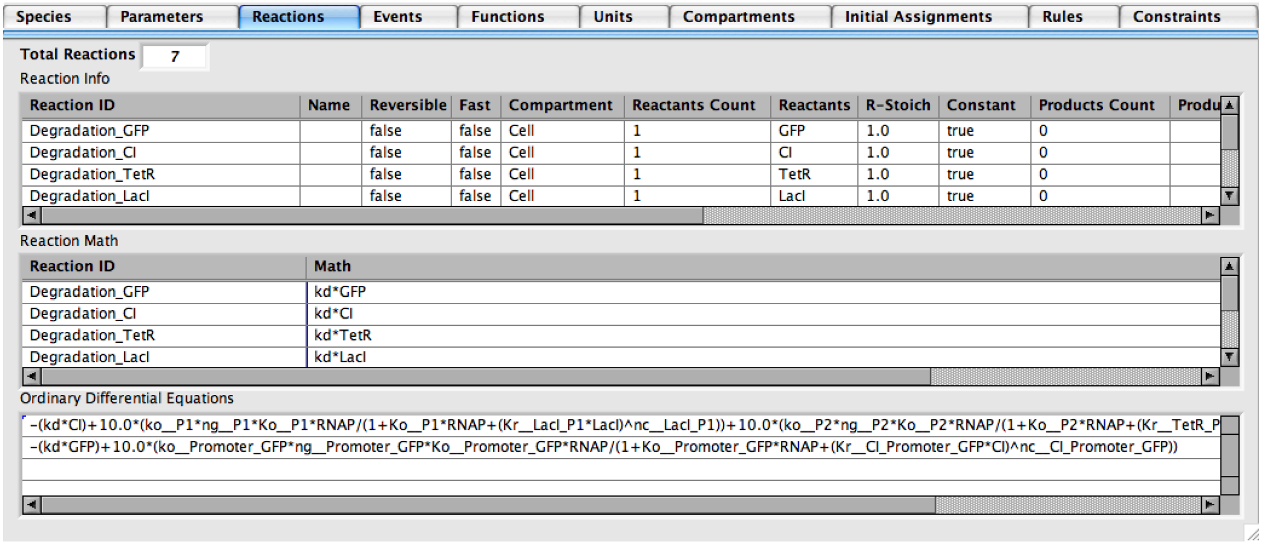
ID	Substance Units	Length Units	Extent Units	Conversion Factor	Model Components
and	mole				listOfUnitDefinitions listOfCompartments listOfSpecies listOfParameters
Name	**Time Units**	**Volume Units**	**Area Units**		
Created by iBioSim flatten routine		litre			

Total Species 9

ID	Name	Compartment	Initial Amount	Initial Concentration	Units	Only Susbtance Units	Boundary Condition	Constant	Conversion Fa
CI		Cell	0.0	NaN		true	false	false	null
GFP		Cell	0.0	NaN		true	false	false	null
LacI		Cell	0.0	NaN		true	true	false	null
P1		Cell	4.0	NaN		true	false	false	null
P2		Cell	2.0	NaN		true	false	false	null
P3		Cell	2.0	NaN		true	false	false	null
P3_mRNA		Cell	0.0	NaN		true	false	false	null
Promoter_GFP		Cell	2.0	NaN		true	false	false	null
TetR		Cell	0.0	NaN		true	true	false	null

Figure 2. D-VASim showing the SBML components of a genetic AND gate model.

All the SBML components of a model can be analyzed here in a clean tabular form. For example, The **Species** tab shown in Figure 2 depicts the total number of species used in the genetic AND gate model along with their corresponding details. Similarly, the **Reactions** tab shown in Figure 3 contains the information related to the reaction kinetics of a model. D-VASim also generates the ordinary differential equations of a model, which can be seen in Figure 3. These equations are used to simulate the deterministic behavior of SBML model.



Total Reactions 7

Reaction Info

Reaction ID	Name	Reversible	Fast	Compartment	Reactants Count	Reactants	R-Stoich	Constant	Products Count	Produ
Degradation_GFP		false	false	Cell	1	GFP	1.0	true	0	
Degradation_CI		false	false	Cell	1	CI	1.0	true	0	
Degradation_TetR		false	false	Cell	1	TetR	1.0	true	0	
Degradation_LacI		false	false	Cell	1	LacI	1.0	true	0	

Reaction Math

Reaction ID	Math
Degradation_GFP	kd*GFP
Degradation_CI	kd*CI
Degradation_TetR	kd*TetR
Degradation_LacI	kd*LacI

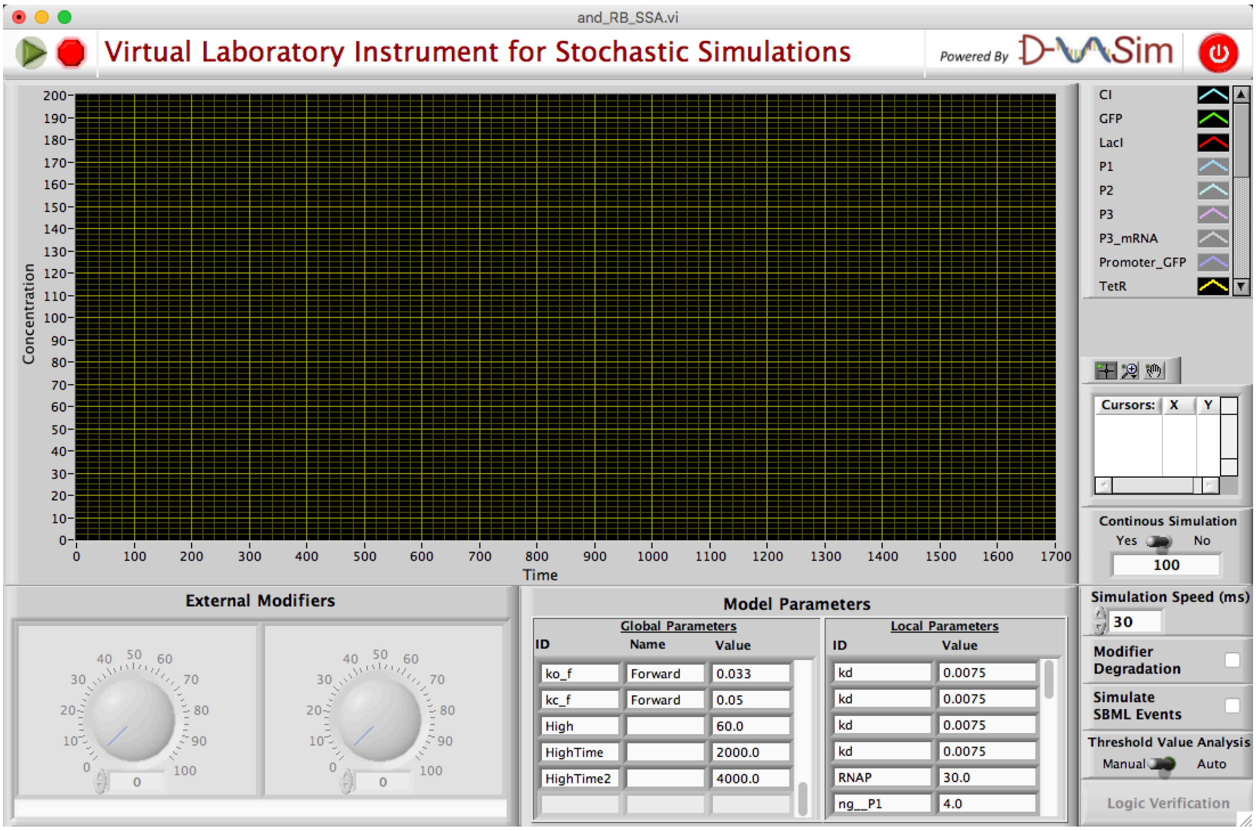
Ordinary Differential Equations

```
-(kd*CI)+10.0*(ko__P1*ng__P1*Ko__P1*RNAP/(1+Ko__P1*RNAP+(Kr__LacI_P1*LacI)^nc__LacI_P1))+10.0*(ko__P2*ng__P2*Ko__P2*RNAP/(1+Ko__P2*RNAP+(Kr__TetR_P
-(kd*GFP)+10.0*(ko__Promoter_GFP*ng__Promoter_GFP*Ko__Promoter_GFP*RNAP/(1+Ko__Promoter_GFP*RNAP+(Kr__CI_Promoter_GFP*CI)^nc__CI_Promoter_GFP))
```

Figure 3. Reactions tab showing the reaction kinetics of a genetic AND gate model.

1.2. Generating a Virtual Laboratory Instrument

Once the components of SBML model are analyzed, a virtual environment for the interactive simulation can be generated simply by clicking on any of the two options, **Generate SSA VI** or **Generate ODE VI** shown at the top right corner of a tool. Now click **Generate SSA VI** button to generate a virtual instrument (VI) for stochastic simulations of a genetic AND gate model. This will bring up a virtual instrument shown in Figure 4.



Continous Simulation
Yes [✓ No]
100

Simulation Speed (ms)
30

Modifier Degradation □

Simulate SBML Events □

Threshold Value Analysis
Manual [✓ Auto]

Model Parameters

Global Parameters			Local Parameters	
ID	Name	Value	ID	Value
ko_f	Forward	0.033	kd	0.0075
kc_f	Forward	0.05	kd	0.0075
High		60.0	kd	0.0075
HighTime		2000.0	kd	0.0075
HighTime2		4000.0	RNAP	30.0
			ng__P1	4.0

Figure 4. Virtual Instrument specifically for a genetic AND gate model.

Any instrument for stochastic simulation generated by D-VASim for the first time does neither reveal the names of graph legend nor the modifier controls, but only the parameters information. Pressing the **RUN** button (▶ shown at the top left corner) starts the simulation and makes the graph legends and modifiers visible.

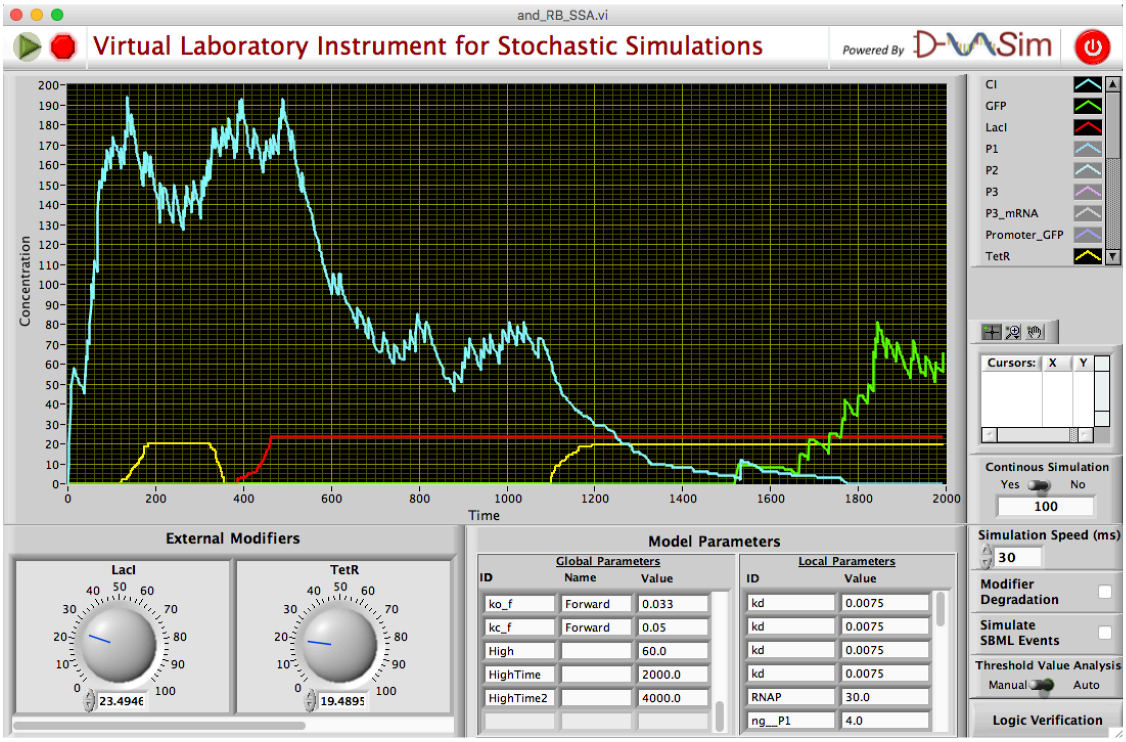
2. Virtual Experimentation for Stochastic Simulations

The virtual instrument (shown in Figure 4) looks similar to a physical instrument, which can be used to interact with the model by tuning the input concentrations with the control knobs and observing the effects graphically.

Once the simulation is started after pressing the ▶ button, user can increase or decrease the concentration of input (or external modifier) species and observe its effects on the model during run time. This runtime interaction with the model gives user an insight of being in the lab performing wet-lab experiments.

Figure 5 shows the screen shot of the stochastic simulation of a genetic AND gate taken randomly after ~2000 time units has lapsed. This figure signifies that unlike SBML events, which are predefined in the SBML file, user can interact with the model and change the concentration of input species to any level and at any instant of time. In case if the concentration of input species are required to be triggered to any specific level instantly, the numeric displays located beneath the control knobs can be used. As same as the user interact with the model by varying the input concentration using control knobs, the parameters can also be varied and their effects can be observed graphically during runtime.

When a **STOP** button (⬣ located besides the run button at the top left corner) is pressed, screen shot of a graphical window is captured and saved inside *Sim_Results* folder located in the default application directory.



LacI: 23.4946

TetR: 19.4895

Continous Simulation: Yes / No — 100

Simulation Speed (ms): 30

Modifier Degradation □

Simulate SBML Events □

Threshold Value Analysis: Manual / Auto

Logic Verification

Model Parameters

Global Parameters

ID	Name	Value
ko_f	Forward	0.033
kc_f	Forward	0.05
High		60.0
HighTime		2000.0
HighTime2		4000.0

Local Parameters

ID	Value
kd	0.0075
kd	0.0075
kd	0.0075
kd	0.0075
RNAP	30.0
ng__P1	4.0

Figure 5. Screen capture of the stochastic simulation of a genetic AND gate.

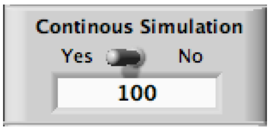
Each image is saved by the following naming convention; "<model-name>_SSA_Screenshot.bmp", where <model-name> is the name of a model. For example, the screenshot of a graphical window of a genetic AND gate model is saved as ***and_RB_SSA_Screenshot.bmp***. The simulation traces of the entire session are also saved in the *Sim_Results* folder with the convention "<model-name>_SSADatalog.xls". Pressing the **RUN** button again starts the simulation from the beginning.

Alert – When the **STOP** button is pressed, the captured screen shot and the data log file overwrites any previously saved files of the same model.

Tip – The properties of graphical plots, including colors, can be changed by a left mouse click on the color of graph legend and selecting the desired menu.

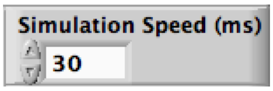
2.1. Simulation Settings

The simulation settings shown at the right bottom corner contains the following six simulation options.



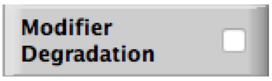
Continous Simulation
Yes No
100

1. **Continuous Simulation:** User may specify here if they want to perform continuous simulation or for a specific interval of time. If switched to "No", the box below can be used to specify the simulation time for which user wants to run the simulation. The default value of this switch is "Yes" meaning that the continuous simulation is always performed unless specified.



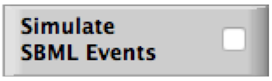
Simulation Speed (ms)
30

2. **Simulation Speed**: This option allows you to speed up or slow down the simulation by entering the numerical value in milliseconds.



Modifier Degradation □

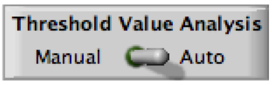
3. **Modifier Degradation**: When checked, perform more realistic simulation by allowing the input concentrations to degrade over time.



Simulate SBML Events □

4. **Simulate SBML Event**: When checked, simulate the events described in SBML file.

Tip – The triggering time and the assignment value of SBML events can be changed during run time until the time of event occurs. For example, if the triggering time of event is 1000 time units, the trigger time and the assignment value can be changed before 1000 time units has lapsed.

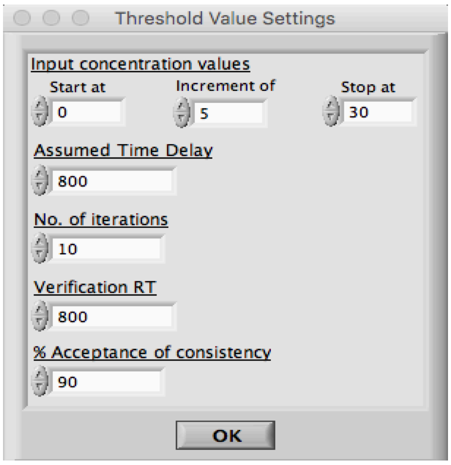


Threshold Value Analysis
Manual Auto

5. **Threshold Value Analysis**:
   Manual – In this mode, VI assume that the threshold value of input species and the propagation delays are already known.
   Auto – When this mode is selected, VI estimates the threshold value based on the setting tab shown in Figure 6 below.
   - **Start at** – Specify the initial concentration of input species from which the analysis is desired to be started. It is recommended to keep this value 0 for correct results.

- **Increment of** – This control allow you to set up the concentration value, which the tool increments during each iteration for checking threshold concentration.

  For example: If you set
  Start at: 0
  Increment of: 5
  Stop at: 15



Input concentration values

Start at: 0

Increment of: 5

Stop at: 30

Assumed Time Delay: 800

No. of iterations: 10

Verification RT: 800

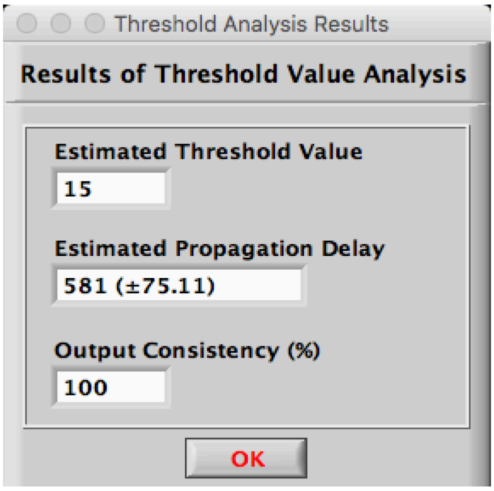
% Acceptance of consistency: 90

Figure 6. Settings dialogue for automatic threshold value analysis.

  Tool will check the threshold concentration for 4 different levels of concentration. i.e. 0 --> 5 --> 10 --> 15, for all possible input combinations.

- **Stop at** – Specify the concentration of inputs at which the threshold analysis should be stopped.
- **Assumed Time Delay** – Specify initially assumed time delay.
- **No. of iterations** – For how many times you want to verify if the estimated threshold value is consistent.
- Verification RT – For how long you want to verify the estimated threshold value for each no. of iterations mentioned above.
- **% Acceptance of consistency** – Specify the minimum percentage of threshold value consistency you want. If the chosen value is 90, the tool give the results only when the consistency of estimated threshold value is more than or equal to 90%.

When the tool finish estimating the threshold value with the desired minimum percentage of consistency, the results are displayed in the dialogue box shown in Figure 7. The value of estimated propagation delay is the average value over all “No. of iterations” specified in the settings dialogue (Figure 6). The value enclosed in braces after the average estimated propagation delay specifies the standard deviation.


Threshold Analysis Results
Results of Threshold Value Analysis
OK

Estimated Threshold Value: 15

Estimated Propagation Delay: 581 (±75.11)

Output Consistency (%): 100

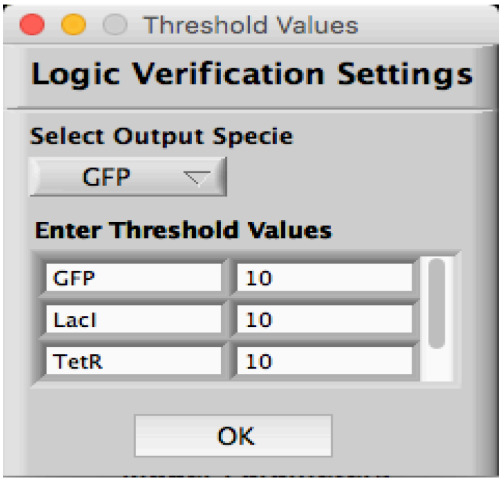
Figure 7. Dialogue box indicating the estimated results of threshold value and propagation delay analysis.

The results are stored in the current application directory with the following naming convention; “<model-name>_SSA-TP-Analysis.txt”, where <model-name> is the name of a model. For the case of genetic AND gate, the results will be saved as ***and_RB_SSA-TP-Analysis.txt***.

6. **Logic Verification** – Obtain the Boolean logic exist in the model. Clicking this button prompts the user to input the threshold values of input and output species, as shown in Figure 8. When automatic threshold value analysis is performed, this box is auto-populated with the estimated threshold value. This option is disabled when the simulation is stopped.

⚠ **Alert –** Make sure to apply all the possible input combinations each after the correct propagation delays. The results of boolean logic analysis will otherwise be incorrect.

⚠ **Alert –** D-VASim v1.0 supports the logic verification of up to 2 input genetic logic circuits.



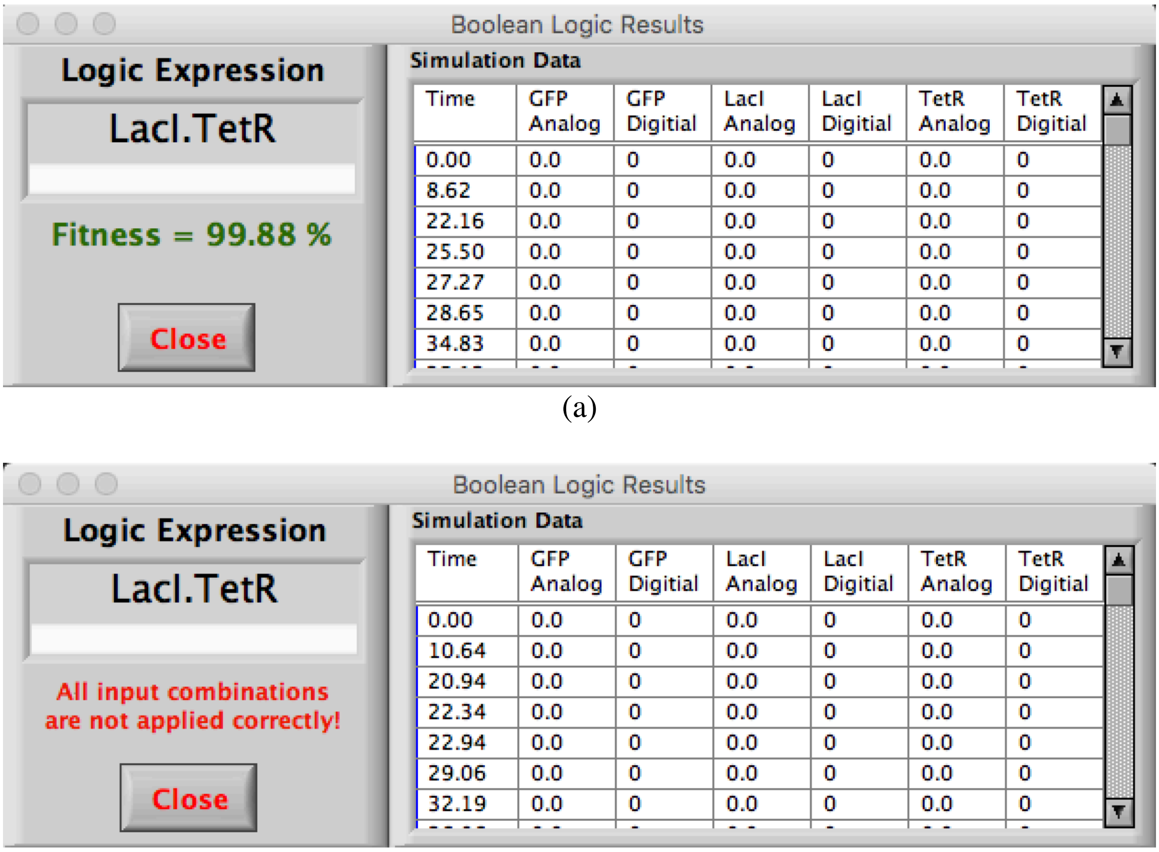
Select Output Specie: GFP

Enter Threshold Values

GFP	10
LacI	10
TetR	10

Figure 8. Dialogue box to set up the threshold values for Boolean logic analysis.

When all the threshold values are entered, a new window (Figure 9) pops up indicating the logic function, in terms of a Boolean expression, extracted from the genetic logic circuit model. This window also contains the entire simulation data as analog data points and their corresponding digital data according to the defined threshold values. If all the input combinations are applied correctly during the simulation, a message in green text (Figure 9 (a)) will appear showing the percentage fitness of the boolean expression in the acquired simulation data. If all the possible input combinations are not applied, a boolean expression may appear with the message in red text (Figure 9 (b)) indicating that all input combinations are not applied correctly. Boolean express box may also be left blank when all the input combinations are not applied correctly. A correct set of input combinations is the one in which all the possible input combinations are applied each after the minimum propagation delay.



Logic Expression

LacI.TetR

Fitness = 99.88 %

Simulation Data

Time	GFP Analog	GFP Digitial	LacI Analog	LacI Digitial	TetR Analog	TetR Digitial
0.00	0.0	0	0.0	0	0.0	0
8.62	0.0	0	0.0	0	0.0	0
22.16	0.0	0	0.0	0	0.0	0
25.50	0.0	0	0.0	0	0.0	0
27.27	0.0	0	0.0	0	0.0	0
28.65	0.0	0	0.0	0	0.0	0
34.83	0.0	0	0.0	0	0.0	0

(a)

Logic Expression

LacI.TetR

All input combinations are not applied correctly!

Simulation Data

Time	GFP Analog	GFP Digitial	LacI Analog	LacI Digitial	TetR Analog	TetR Digitial
0.00	0.0	0	0.0	0	0.0	0
10.64	0.0	0	0.0	0	0.0	0
20.94	0.0	0	0.0	0	0.0	0
22.34	0.0	0	0.0	0	0.0	0
22.94	0.0	0	0.0	0	0.0	0
29.06	0.0	0	0.0	0	0.0	0
32.19	0.0	0	0.0	0	0.0	0

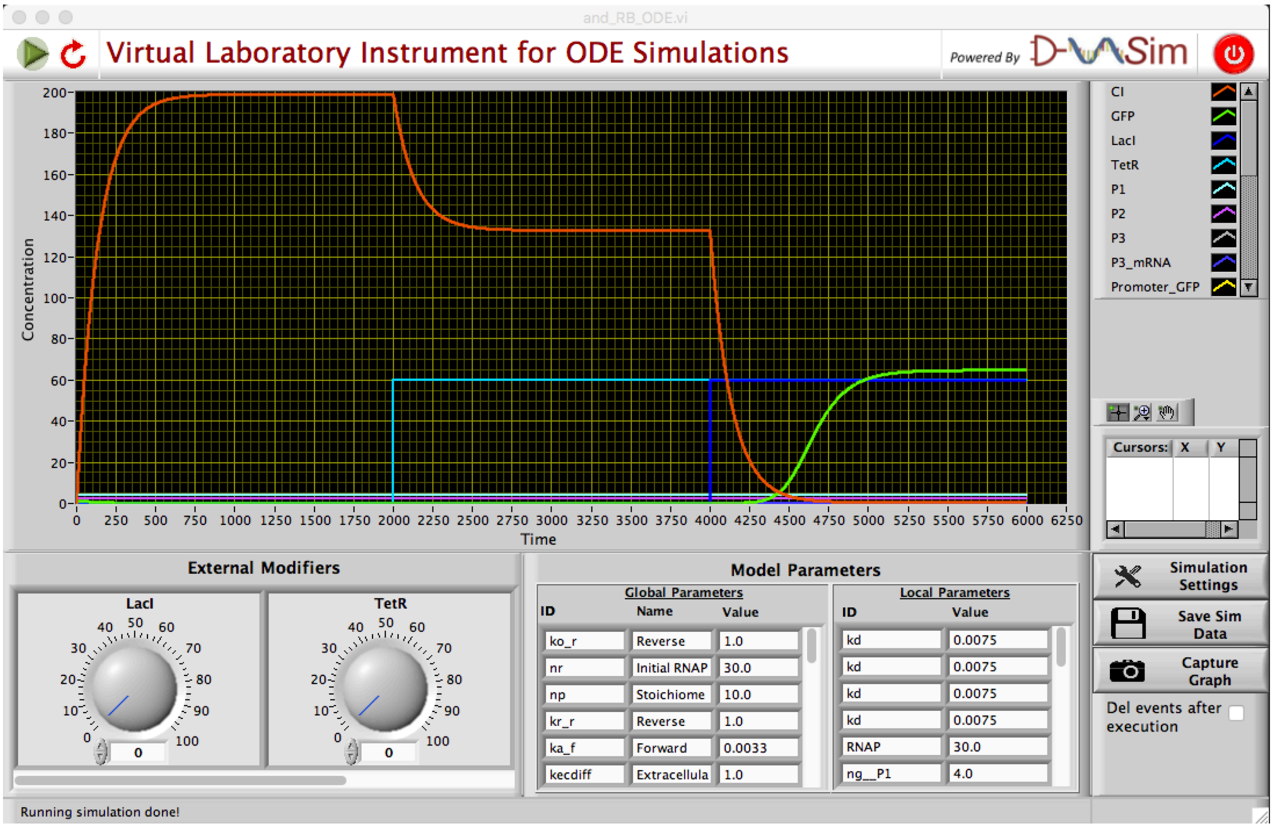
(b)

Figure 9. Boolean logic extracted from the analog simulation data along with its corresponding digital data.
(a) Percentage fitness appears when all the input combinations are properly applied.
(b) Error message appears when all the input combinations are not applied correctly before trying to estimate the boolean logic of a circuit.

3. Virtual Experimentation for ODE Simulations

Similar to generating virtual instrument for stochastic simulations, the virtual instrument for deterministic simulations can also be generated by clicking on **Generate ODE VI** button depicting on Figure 2.

When the ▶ button is pressed, the deterministic simulation results of a genetic AND gate for 6000 time units are appeared as shown in Figure 10. As same as in the case of stochastic simulation, user can change the concentration of external input species using control knobs for deterministic simulation also. However, every single change in the input concentration or parameters results in the execution of selected simulation algorithm for the defined interval of time and plots the new set of curves.



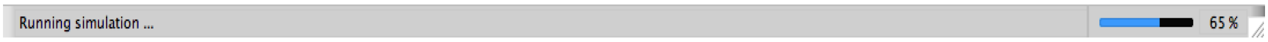
Model Parameters

Global Parameters			Local Parameters	
ID	Name	Value	ID	Value
ko_r	Reverse	1.0	kd	0.0075
nr	Initial RNAP	30.0	kd	0.0075
np	Stoichiome	10.0	kd	0.0075
kr_r	Reverse	1.0	kd	0.0075
ka_f	Forward	0.0033	RNAP	30.0
kecdiff	Extracellula	1.0	ng__P1	4.0

Running simulation done!

Figure 10. Screen capture of the deterministic simulation of a genetic AND gate.

The horizontal *status bar* is shown at the bottom in Figure 10. It shows the status of the actions being performed in this VI for ODE simulations. For example, pressing the run button shows the status of simulation as shown in Figure 11 below. The progress bar shown at right-hand side depicts the percentage of results currently being obtained. The plot is visible once the progress bar reaches to 100 %.


Running simulation ...

65 %

Figure 11. Screen capture of the status bar showing that the VI is running simulation. The progress bar shown at right-hand side indicates that the 65% of simulation data is obtained.

When a **RESET** button (↻) button is pressed, screen shot of a graph window and the simulation data is captured and saved under *Sim_Results* folder located in the default application directory. Each image is saved by the following naming convention; "<model-name>_ODE_Screenshot.bmp", where <model-name> is the name of a model. For example, the screenshot of a graphical window of a genetic AND gate model is saved as ***and_RB_ODE_Screenshot.bmp***. The simulation traces of the entire session are saved with the naming convention; "<model-name>_ODEDatalog.xls". Pressing the **RUN** button again starts the simulation from the beginning.

Alternatively, the simulation data and the graphical plots can be stored by pressing the buttons shown in Figure 12(a) and 12(b) respectively.

Save Sim Data
(a)
Capture Graph
(b)

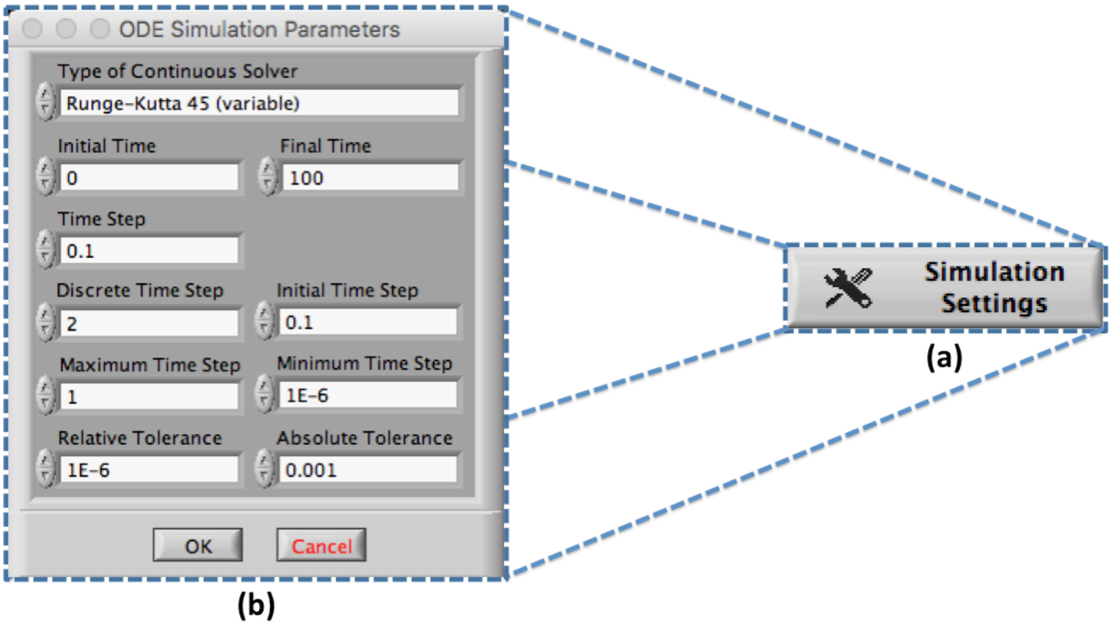
Figure 12. Available options in ODE VI to (a) Save simulation data. (b) Capture graph window.

Alert – When the **RESET** or **Save Sim Data** or **Capture Graph** button is pressed, the captured screen shot and the data log file overwrites any previously saved files of the same model.

Tip – The properties of graphical plots, including colors, can be changed by a left mouse click on the color of graph legend and selecting the desired menu.

3.1. Simulation Settings

The **Simulation Settings** button (shown in Figure 13(a)) helps user to set the desired simulation parameters. A settings window, shown in Figure 13(b), is popped-up when user clicks on the **Simulation Settings** button.


ODE Simulation Parameters

Type of Continuous Solver: Runge-Kutta 45 (variable)

Initial Time: 0

Final Time: 100

Time Step: 0.1

Discrete Time Step: 2

Initial Time Step: 0.1

Maximum Time Step: 1

Minimum Time Step: 1E-6

Relative Tolerance: 1E-6

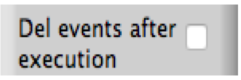
Absolute Tolerance: 0.001

Figure 13. ODE simulation properties (a) Simulation settings option. (b) Window of simulation settings.

D-VASim VIs for ODE simulation supports the following ten continuous solvers:

1. Runge-Kutta 1 (Euler)
2. Runge-Kutta 2
3. Runge-Kutta 3
4. Runge-Kutta 4
5. Runge-Kutta 23 (variable)
6. Runge-Kutta 45 (variable)
7. BDF (variable)
8. Adams-Moulton (variable)
9. Rosenbrock (variable)
10. Discreet States Only

The parameters shown in Figure 14(b) are set as the default parameters. There is one other option available at the right bottom of ODE VI, which is described below



Del events after execution: When marked checked, the events are deleted internally after executing once. This helps increasing the simulation speed by avoid checking the executed events repeatedly.

Tip – If the desired simulation results are not obtained, try simulating the model again by checking or unchecking the option “Del events after execution”.

4. General Software Options and Tips

Following are some general tips and options for both stochastic and ODE virtual instruments generated by D-VASim.

Tip – Hover mouse over simulation options to pop up the tip strip with a short description.

Tip – Pressing keys CMD+SHIFT+H (on MAC OS) and CTRL+H (on Windows OS) open the context help.

Tip – User can right click on the graph window to get the runtime shortcut menu shown in Figure 14 below.

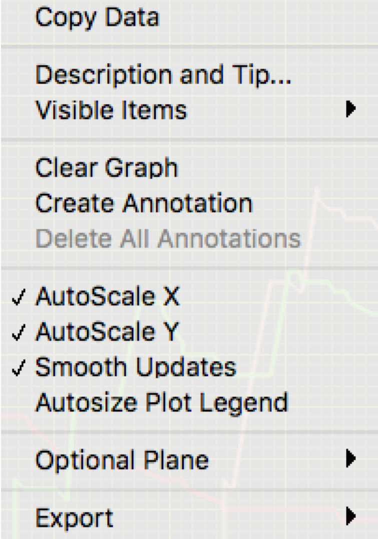
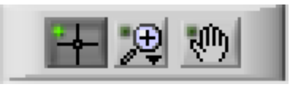
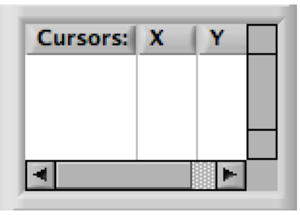


Figure 14. Runtime shortcut menu of graph window.



Graph Palette: This option can be used to zoom-in, zoom-out, expand graph etc.



Cursor Legend: With the help of this box, cursors can be set to track the values of desired species. For example, single-plot cursor can be set to monitor GFP specie by **right clicking** in the Cursors window and then selecting **Single-Plot** from **Create Cursors** option, as depicted in Figure 15(a). Once the cursor is created, it can be snap to GFP specie by **right clicking** on the created cursor and then go to **Snap To** option and select **GFP**. These steps are shown in Figure 15(b).

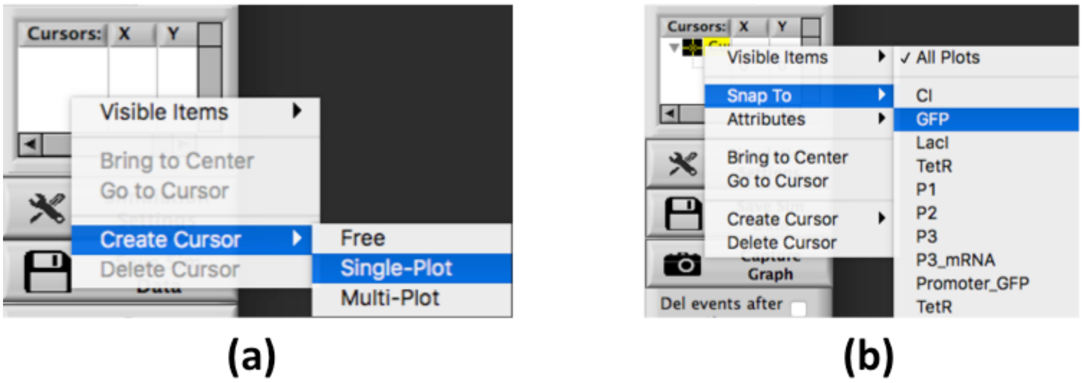


Figure 15. Steps of creating cursors on species.

Finally, similar to any physical instrument, virtual instruments powered by D-VASim also have emergency shut down button (⏻) shown at the top right corner of a virtual instrument. This button can be used to stop and shut down the instrument instantly.

References

[1]. Systems Biology Markup Language, http://sbml.org/Main_Page.

[2]. National Instruments LabVIEW, http://www.ni.com/labview/.

[3]. Dräger A, Rodriguez N, Dumousseau M, Dörr A, Wrzodek C, Le Novère N, Zell A, and Hucka M. **JSBML: a flexible Java library for working with SBML**. *Bioinformatics* (2011), 27(15): 2167–2168.

[4]. Chris J. Myers. Engineering Genetic Circuits. Chapman & Hall/CRC Press, July 2009.