Table of Contents

Chapter 1: BMSS Guidelines

BMSS, an automated BioModel Selection System for gene circuit designs, supports Bio-model automated fitting and selection processes, providing a means to efficiently derive the best model candidate that could capture the ***transient (time-series)*** dynamic profiles of a Bio-part or device using characterization data. This system is an open source platform which is implemented in Python. The developed Python Package BMSSlib is available for downloading from GitHub.

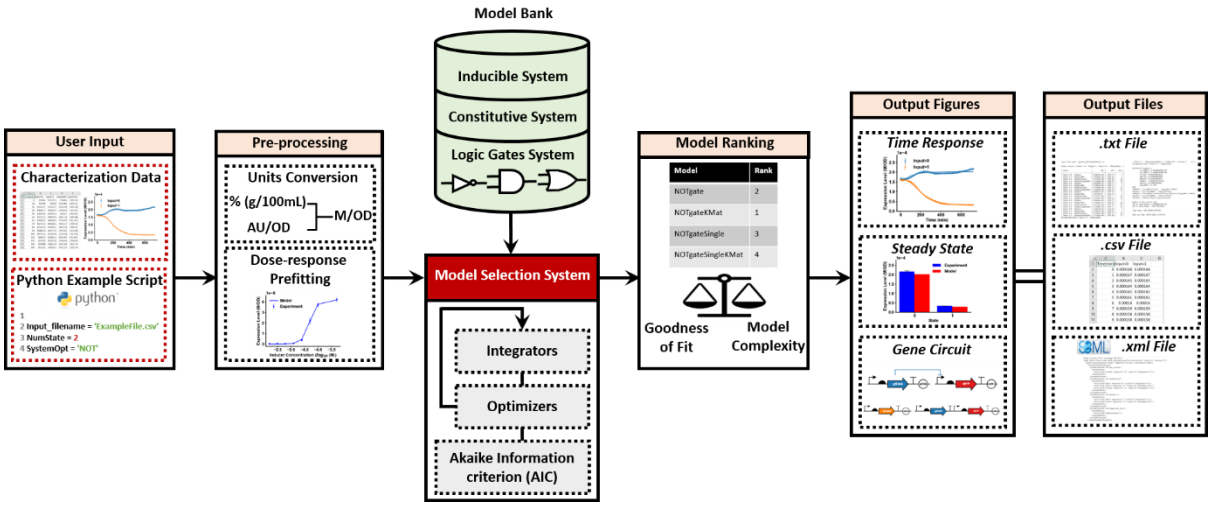
The package supports ***three*** routinely used gene regulatory systems:

- Inducible System
- Constitutive System (Single Dataset or Multiple Datasets)
- Logic Gate System (NOT, AND, and OR gates)

1.1. System Overview

The BMSS supports the following features:

- Read in User Input Characterization data in .csv file
- Model Fitting and Model Selection from a library of pre-established models
- Exhibit the graphical figures of processed experimental data, experimental & Simulation data, and/or steady state results.
- Export the best model candidate in Systems Biology Markup Language (SBML) file (.xml)
- Generate Synthetic Biology Open Language (SBOL) visual-compliant gene circuit diagrams for the best model candidate
- Export Model Data in .csv file

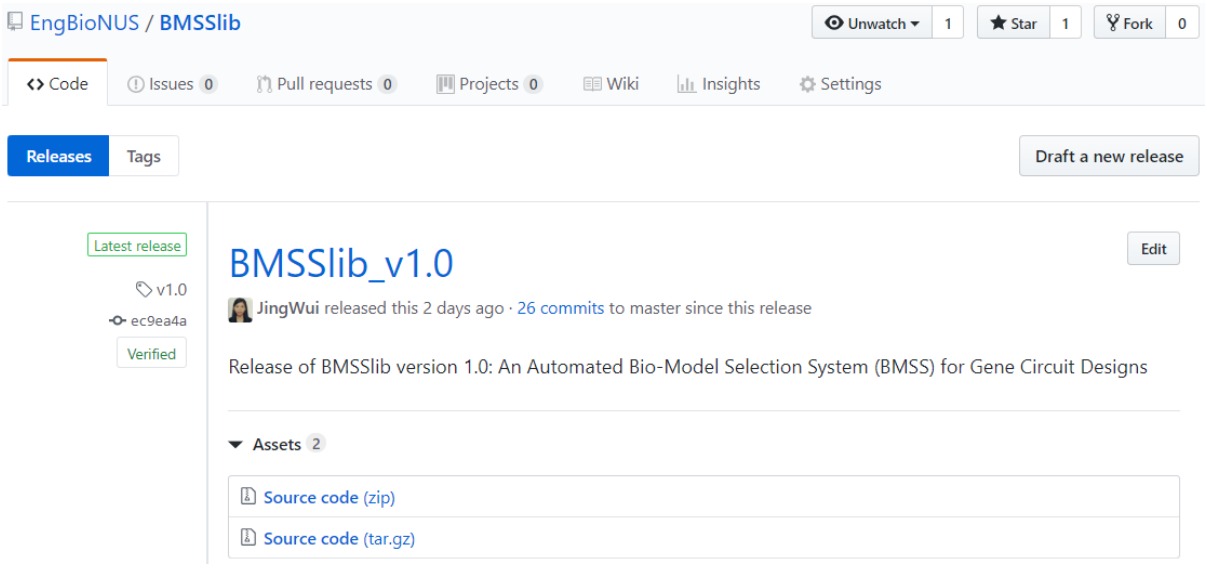


Descriptions: The user input characterization data in csv file (including time, fluorescence level relative to cell growth, and their corresponding standard deviations) is read by the system data reader for unit conversions and dose-response prefitting (only applicable for inducible system). Numerical integration, optimization, and plotting packages are used to solve the ordinary differential equations, iteratively fit models retrieved from the model bank to experimental data through minimizing the sum squared residuals and plot the graphical results for data visualizations. The model selection algorithm is based on the AIC statistical inference criterion. The output figures display the time-response and steady-state response behaviors in conjunction with the imported measured experimental data and their corresponding standard deviations represented in error bars. User-defined customizable SBOL-

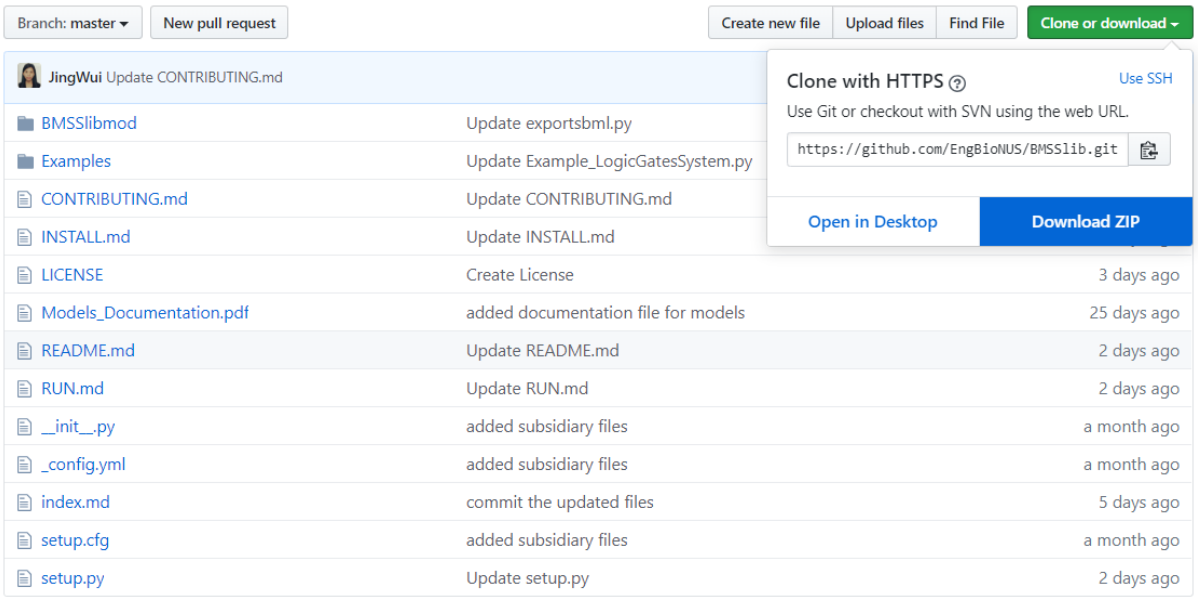
compliant gene circuit graphics rendering is also exhibited by the BMSS platform. The details of the model ranking and the best model candidate are listed in the output text file with the model simulation data exported in a separate csv file. The SBML representation of the chosen model is also programmed and exported in XML format.

1.2. Installation

A stable BMSSlib_v1.0 has been released and can be downloaded at GitHub Release page.



Users can also download the developmental version of BMSS python package BMSSlib zip file from GitHub, and unzip to get the BMSSlib-master folder.



There are two subfolders inside the BMSSlib-master folder:

- ***BMSSlibmod***: contains all the main source files

- ***Examples***: contains the example main files for the three gene regulatory systems (Inducible System, Constitutive System, and Logic Gate System), and the example input files (.csv) containing the characterization data stored in the **InputData** folder. The following output files will be exported into the **Results** folder:
    - .txt – output text file listing the model ranking table, the model formulation and its corresponding estimated parameters of the recommended best model candidate.
    - .csv – model simulation results for post-processing
    - .xml—the SBML file encoding the details of the selected best model candidate, which allows the reproducibility and transferability for further post-processing processes.

BMSSlib is implemented in *Python 3.6.5* environment and can be *soon* installed using *pip* in Anaconda Prompt/terminal as shown below:

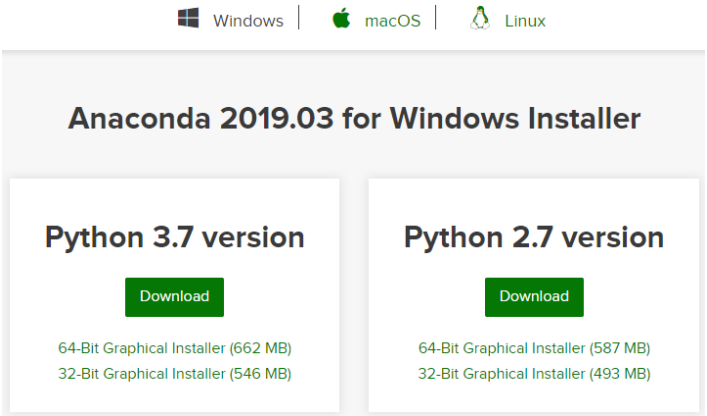
```
(BMSSenv) C:\Users\bchyjw>pip install BMSSlib
```

The system can be run on different OS systems:

New users can first download and install the open-source Anaconda Distribution which comes with all the fundamental packages and GUI applications (Spyder, Jupyter Notebook etc.) of Python for scientific computing. The open source packages can be individually installed from the Anaconda repository with the `conda install` command or using the `pip install` command that is installed with Anaconda.

The default installation of Anaconda at the page includes Python 3.7 version and Python 2.7.

- Download Python 3.7, choosing based on your Operating System (Windows, MacOS, Linux) as displayed on top of the Python versions.



Note: In view of the potential compatibility issues for different versions of dependencies, it is highly recommended to create a virtual environment of Python 3.6.5 in Anaconda.

- Open Anaconda Prompt (functions like command prompt) or Anaconda Navigator (Environments → base (root) → Open Terminal) to create a virtual environment as BMSSenv (or any other name) by typing the command below:

  ```
  conda create -n BMSSenv python=3.6.5 anaconda
  ```

- When prompted with a message to ask whether to proceed with the packages installation or update, just insert y, refers to yes to proceed with the installation.
- After all the installations have finished, insert the command below to activate the newly created environment
  conda activate BMSSenv

In additional to those fundamental packages available in Anaconda, users are required to install two additional packages using the terminal at Anaconda Navigator or Anaconda Prompt:

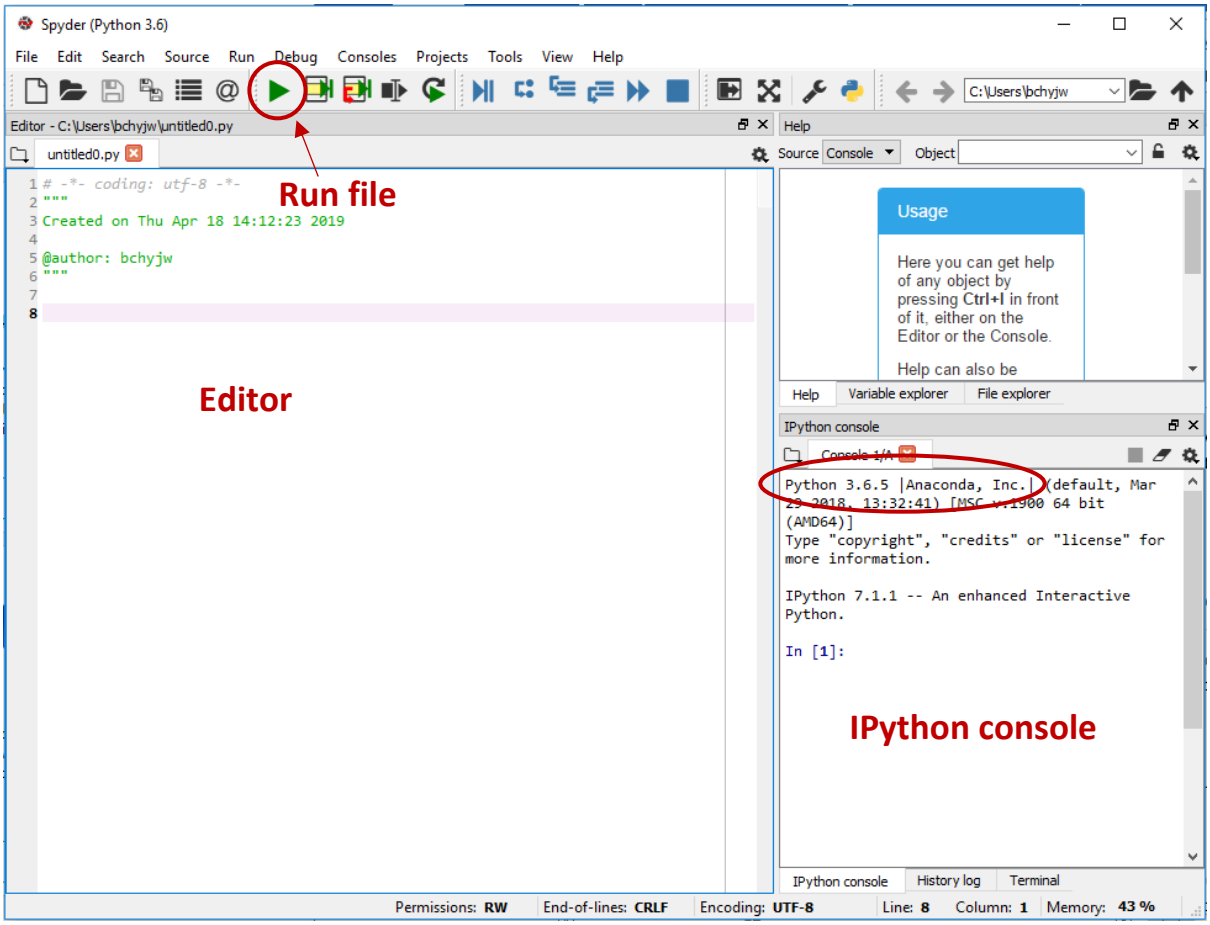
pip install tabulate

pip install tesbml

1.3. Quick Start

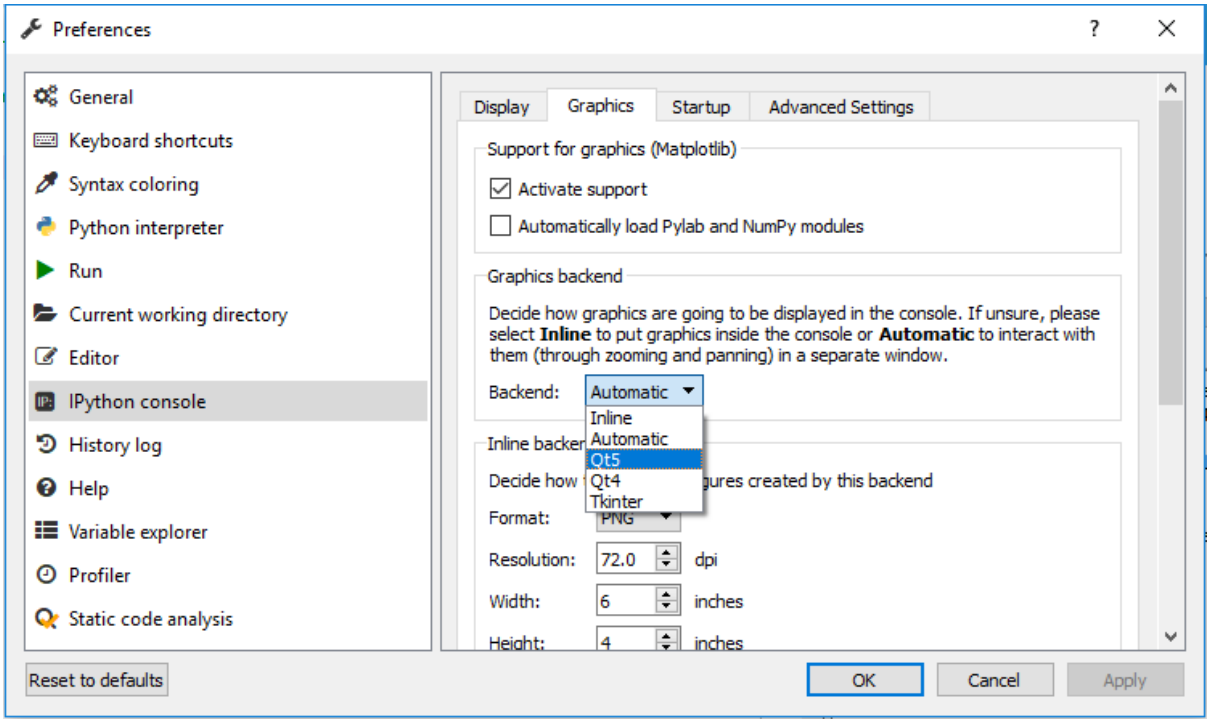
Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included in Anaconda. The IDE includes editing, interactive testing, debugging and introspection features. Refer to the Anaconda Documentation for more details.

- Launch Spyder (BMSSenv) from your start menu or Anaconda Navigator.



- Check if the right Python version (3.6.5) is used in the IPython console shown at the right bottom.
- IPython console allows users to run code by line, cell, or file, and render plots right inline.
- Editor is where users can input all their codes for running.

- Click on the Run button to execute the file.
- To display the output Graphics in a separate window, select **Tools → Preferences → IPython console → Graphics**, change the option of **Backend** from **Inline** to **Qt5**.
- Click **Apply** then **OK**
- Close the Spyder IDE and reopen the IDE from Start menu to apply the changed setting.



1.4. Curating Input Data

The raw characterization data are to be obtained from microplate reader. The fluorescence data are normalized to the cell growth OD_{600} after deducting each of the technical triplicate readings from the negative control cultures at that particular time point. The average and standard deviation (S.D.) can then be calculated.

The standard of the user input file for the three gene regulatory systems:

Inducible System:

[Time(min), computed Fluorescence/OD readings in the order of descending inducer concentration, corresponding S.D. values following the same order]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Time(min)	0.12500%	0.06250%	0.03125%	0.01563%	0.00781%	0.00391%	0.00195%	0.00098%	0.00000%	0.125%std	0.0625%st	0.03125%s	0.01563%s	0.00781st	0.00391st	0.00195%s	0.00098%s	0%std
2	0	27.59477	22.55971	23.91481	26.60527	25.50572	34.21403	24.08986	33.93663	24.56543	2.962317	5.86328	7.096696	5.954864	2.863218	11.09499	6.370971	1.451179	2.50505
3	10	35.62949	33.55519	28.80788	27.87137	28.34344	31.20062	23.65252	34.0669	25.09609	9.568997	0.322885	9.197267	4.708114	3.743319	7.335467	5.319813	1.317361	2.71064
4	20	69.56017	57.14021	43.87688	38.63086	29.85434	29.81356	22.50326	26.47797	22.13242	13.97596	11.07401	8.643246	6.649645	5.773839	9.969546	2.333331	5.696152	2.581827
5	30	162.1974	127.5279	93.03068	69.71074	48.67284	37.55033	25.1556	26.69219	19.85452	27.35657	22.5646	16.50681	13.36463	10.12064	13.50402	2.569114	1.003066	1.60225
6	40	302.0339	239.8071	180.6934	126.1931	79.67726	52.92511	28.66026	24.6349	20.60537	46.71979	40.34133	26.82483	20.52171	9.6426	8.618934	5.308358	2.506009	2.55374
7	50	485.8443	395.5596	300.7903	210.2365	126.7234	76.41285	35.09319	26.46132	16.19526	57.39949	50.01254	39.95316	31.56051	16.35006	9.827706	3.789592	5.28647	1.23587
8	60	685.8187	572.2312	441.4452	305.8692	181.1294	102.7457	41.7344	28.03575	15.35404	71.28649	59.48433	41.98884	34.17429	21.56356	18.98299	5.038742	3.663006	1.366119
9	70	888.764	753.0005	591.5833	413.2538	244.0499	136.787	54.22656	29.67224	15.23023	78.80328	64.76577	55.72852	45.12623	21.02845	19.14032	4.379934	1.901005	1.032475
10	80	1085.508	935.5087	748.3811	529.9757	326.2495	182.2103	69.73601	36.17173	16.11524	75.5577	66.17849	57.65701	50.78266	25.99334	27.07108	7.351197	2.527552	1.795508
11	90	1258.146	1117.843	910.967	659.7759	400.9412	220.3237	80.39061	38.03905	13.13829	94.79006	76.53162	74.68878	56.85507	31.87792	21.33732	6.089192	3.592683	1.912994
12	100	1485.847	1334.05	1093.518	779.4632	460.6503	252.3593	93.96227	42.05653	13.1206	102.0934	78.68588	71.56373	53.17159	26.363	31.17237	6.975086	2.589773	0.700081
13	110	1687.077	1518.674	1230.821	861.2854	511.4432	284.6447	106.4225	47.08271	11.27738	94.94382	75.66161	73.76796	50.7346	34.98583	25.81949	7.172345	3.178813	2.095103

Constitutive System:

Single Dataset: [Time(min), Single Fluorescence/OD data set, Single S.D. data set values]

Multiple Datasets: [Time(min), Multiple Fluorescence/OD data sets, Multiple S.D. data sets values following the same order]

	A	B	C	D	E	F	G	H	I
1	Time(min)	P66	P51	J23108	J23102	P66std	P51std	J23108std	J23102std
2	0	793.4041	2443.6	339.6817	1129.792	9.864046	10.46622	2.31366	32.51676
3	10	824.1178	2374.27	386.023	1176.516	7.916155	12.25735	2.9365	36.61407
4	20	888.1874	2445.674	448.2703	1271.903	8.286637	5.156355	1.406577	35.04243
5	30	918.3077	2410.069	488.4508	1347.289	8.358063	7.372048	4.188244	36.42289
6	40	945.5188	2342.78	521.496	1421.442	9.064196	31.8452	4.376119	35.46303
7	50	960.1947	2327.45	568.2599	1490.074	9.345212	21.98356	2.653389	40.06487
8	60	967.4937	2306.384	601.7069	1534.067	8.036808	34.1196	2.86511	25.37681

Logic Gate System:

NOT gate (one-state): [Time(min), Fluorescence/OD at state=0, Fluorescence/OD at state=1, S.D. of the Fluorescence/OD at state=0, S.D. of the Fluorescence/OD at state=1]

AND, OR gates (two-state): [Time(min), Fluorescence/OD at state=00, Fluorescence/OD at state=01, Fluorescence/OD at state=10, Fluorescence/OD at state=11, S.D data for each of the different states following the same order]

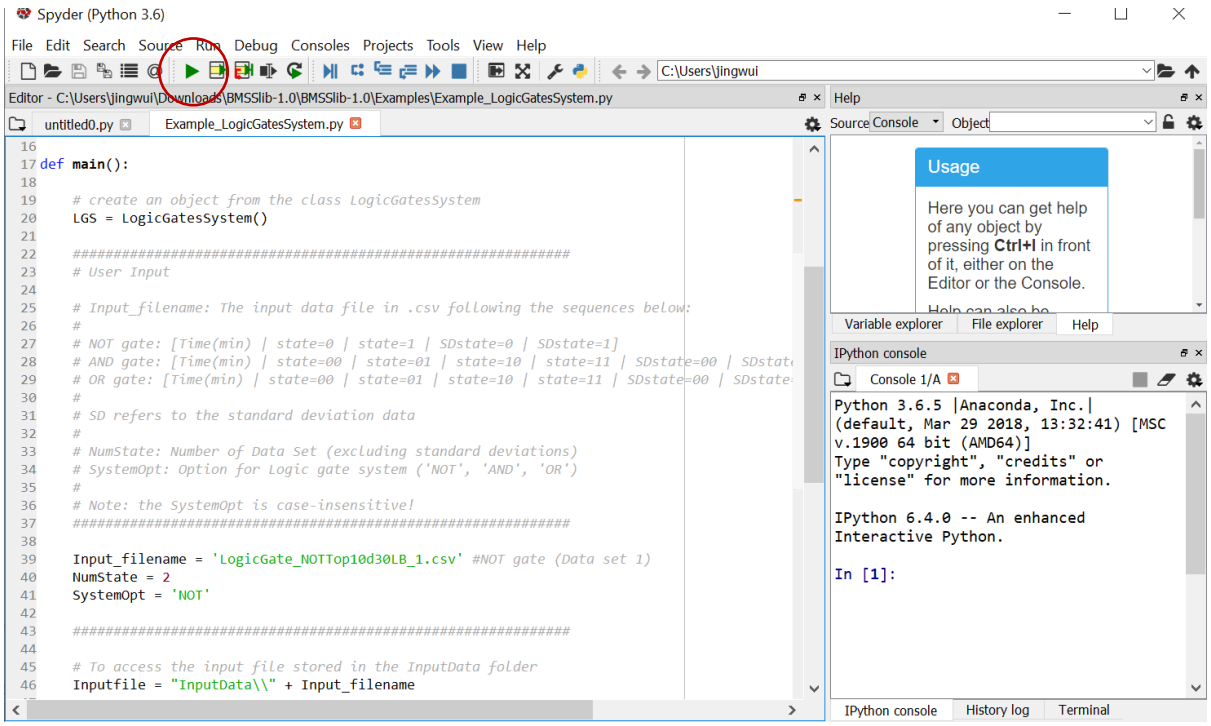
	A	B	C	D	E
1	Time(min)	Input=0	Input=1	Input0std	Input1std
2	0	95081.97	91211.53	1369.597	3192.237
3	10	94306	91050.04	529.0017	1659.306
4	20	93154.74	91322.66	519.1491	1922.176
5	30	93081.67	90204.55	1558.416	2313.033

Note: The input characterization data file must be saved in comma-separated values (CSV) format from Excel. No spacing is allowed for the naming put in the first row of the .csv file. Apply the naming rules: starting with letters then followed by alphabets, letters or underscore, with no spacing, when setting the filename of the input .csv file.

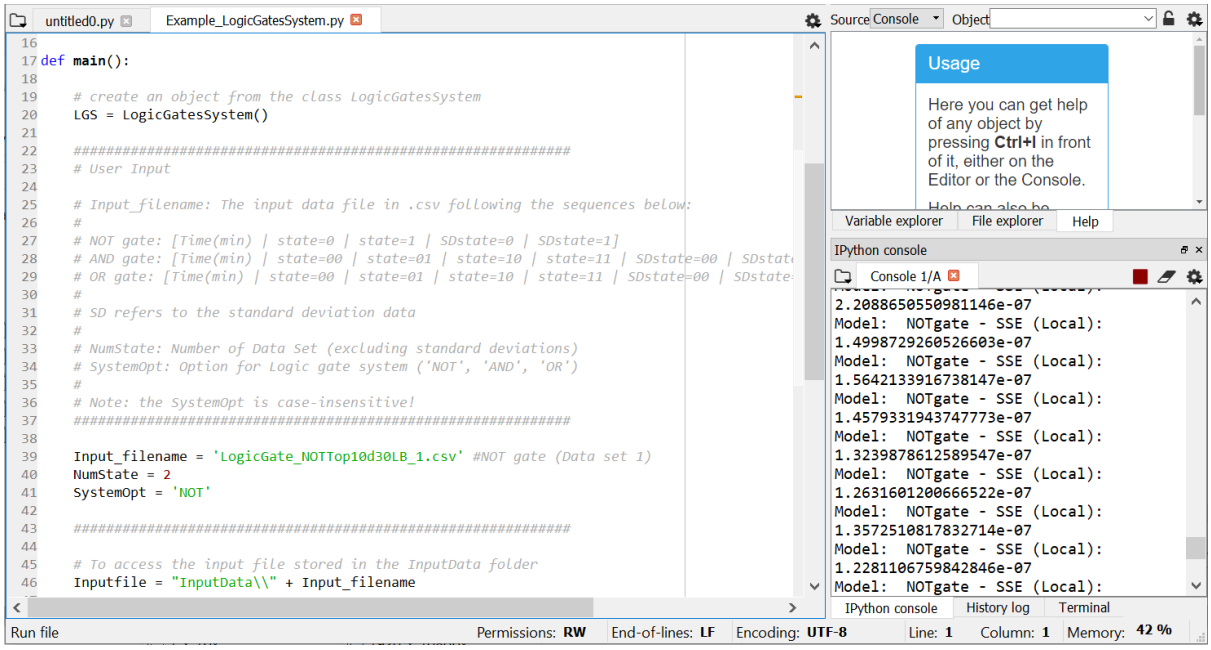
1.5. Running Example file

- To get started, at Spyder, **File → Open** to browse and open the example file (Inducible, Constitutive, or Logic gate system) in .py located in the **BMSSlib-1.0/Examples** (stable release) or **BMSSlib-master/Examples** (developmental version) folder. Select the file based on your system of interest.
- Below is the example file for the Logic gate system. This Logic gate system exemplifies a NOT gate circuit of CRISPR/dCas9 as follows:

The dCas9 gene is expressed constitutively, whereas the gRNA is controlled by an inducible promoter which can be turned ON (state = 1) or OFF (state = 0).

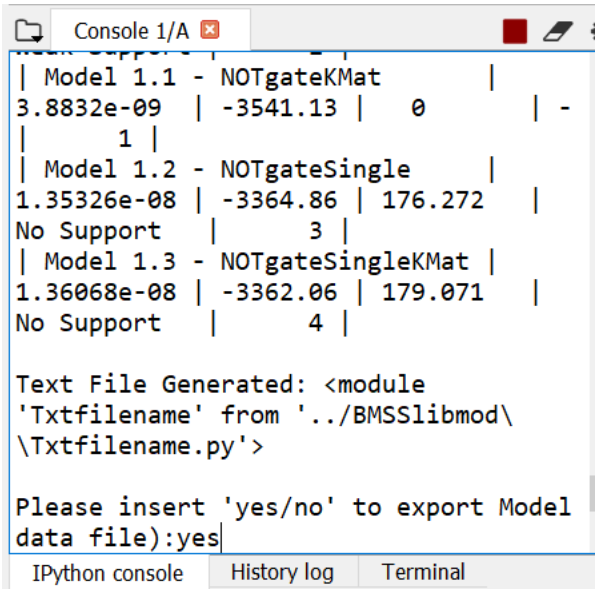


- Click on the  Run button at the toolbar.

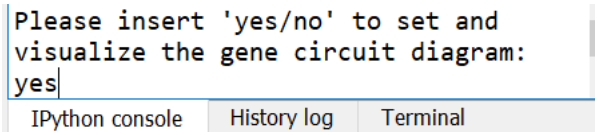


The program is running with the running model followed by the sum squared error (SSE) and the type of the optimizers (global or local) are displayed at the IPython console.

Note: if running the Example file for the inducible system, after clicking on the **Run** button, users will be prompted to insert the *Molar Mass of the inducer* at the IPython console. Please insert **150.13** to represent the Molar Mass of Arabinose inducer in g/mol.



- When the run has finished, users will be prompted to insert either yes or no to export the model data file (in .csv).



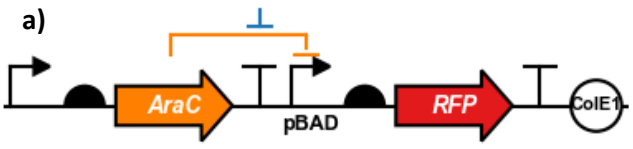
- Next, please insert either yes or no to visualize the SBOL-compliant gene circuit diagram.

1.6. Setting Output Specifications

To visualize the SBOL-compliant gene circuit diagram, users are required to insert the gene circuit specifications to the BMSS system for plotting.

1.6.1. Inducible System Example

With the given sample characterization data file, users are required to insert the following details (shown in Italic bold fonts) to generate the SBOL-compliant visual graphics below (e.g. Arabinose-inducible pBAD promoter system):



Note: Insert only the Italic bold information

- Reporter type: ***RFP***
- Number of plasmids: ***1***
- Name of origin: ***ColE1***
- Number of parts: ***2***
- Name of gene 1: ***AraC***
- Name of gene 2: ***RFP***
- Name of Inducible Promoter: ***pBAD***

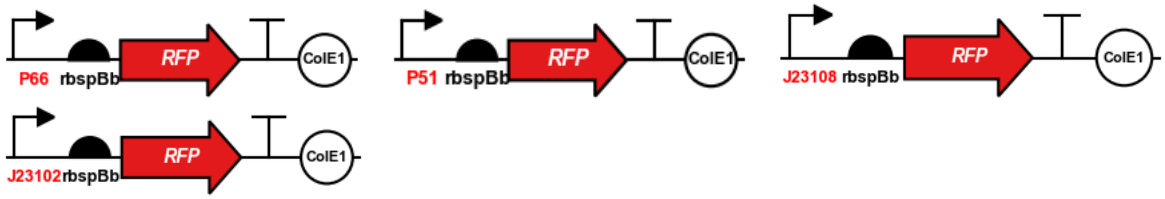
- Reporter type: ***RFP***
- Number of plasmids: ***1***
- Name of origin: ***ColE1***
- Number of parts: ***1***
- Name of gene 1: ***RFP***
- Name of Inducible Promoter: ***pBAD***

1.6.2. Constitutive System Example

The sample characterization data file given consists of multiple independent data sets from varied constitutive promoters with same ribosome binding sites (RBSs).

	A	B	C	D	E	F	G	H	I
1	Time(min)	P66	P51	J23108	J23102	P66std	P51std	J23108std	J23102std
2	0	793.4041	2443.6	339.6817	1129.792	9.864046	10.46622	2.31366	32.51676
3	10	824.1178	2374.27	386.023	1176.516	7.916155	12.25735	2.9365	36.61407
4	20	888.1874	2445.674	448.2703	1271.903	8.286637	5.156355	1.406577	35.04243
5	30	918.3077	2410.069	488.4508	1347.289	8.358063	7.372048	4.188244	36.42289
6	40	945.5188	2342.78	521.496	1421.442	9.064196	31.8452	4.376119	35.46303
7	50	960.1947	2327.45	568.2599	1490.074	9.345212	21.98356	2.653389	40.06487
8	60	967.4937	2306.384	601.7069	1534.067	8.036808	34.1196	2.86511	25.37681

To plot the SBOL-compliant gene circuit diagrams for the example file as shown below, please insert the following information when prompted at the IPython console



Note: Insert only the Italic bold information

- Reporter type: ***RFP***
- Name of Origin: ***ColE1***
- Name of RBS: ***rbspBb***

The different promoter names will be taken from the names provided in the input characterization file.

1.6.3. Logic Gate System Example

This sample logic gate system exemplifies a NOT gate circuit of CRISPR/dCas9 as follows:

Please insert the following information when prompted at the IPython console:

Note: Insert only the Italic bold information

- Reporter type: ***GFP***
- Number of plasmids: ***2***
- Name of origin 1: ***ColE1***

- Name of origin 2: ***p15A***
- Number of parts: ***3***
- Name of gene 1: ***gRNA***
- Name of gene 2: ***dCas9***
- Name of gene 3: ***GFP***

1.7. Running New Characterization Data File

Users can run their own characterization data file measured from microplate reader in BMSS by following the steps given below:

- Prepare the average and calculated standard deviation of the growth normalized data following format stated in Section 1.4.
- Save the input file in .csv and paste the file in **BMSSlib-1.0\Examples\InputData** (stable release)
- In Spyder, **File → Open** and browse to the BMSSlib-1.0\Examples folder and click on the corresponding Example .py file of interest and click **open**
- For each of the different regulatory systems, modify the highlight sections as shown below:

Inducible System:

```
IS = InducibleSystem()


############################################################
# User Input

# Input_filename: The input data file in .csv following the sequences below:
#
# e.g.: [Time(min) | 100uM | 50uM | 10uM | 1uM | 0uM | SD100uM | SD50uM | SD10uM | SD1uM | SD0uM ]
#
# The inducer concentration in the input file must be in units of %, uM, or nM;
# Note: No spacing in word between concentration value and unit
# Note: The concentrations of inducer shall be in descending order.
#
# SD refers to the standard deviation data
#
# NumDataSet: Number of Data Set (excluding standard deviations)
# Inducer_unit: The unit for the inducer concentration
# OptInhibition: True if there is Inhibitory effect at high concentrations,
#           else None if there is only inducible trend
# SystemOpt: Option for Inducible system ('ALL', 'ConstInd', 'DegradeInd', 'DelayInd', 'Inhibition')
#
# 'ALL': (default setting) run through all the models available in the library
# 'ConstInd': run models under the assumption of constant inducer concentration
# 'DegradeInd': run models under the assumption of inducer with fast degradation behavior
# 'DelayInd': run models under the assumption of significant initial delayed response
# 'Inhibition': run models under the assumption of inhibition at high inducer concentrations
#
# Note: When the Inducer_unit is in '%', which refers to mass concentration in (g/100 mL),
# users will be promted to insert the Inducer Molar Mass (in g/mol) at the console the unit
# to convert to the unit of Molar.
#
# Note: the SystemOpt is case-insensitive!
############################################################

# Example file
Input_filename = 'Inducible_pBADAra_rbspBb_MG.csv' # Arabinose (Data set 1)
NumDataSet = 9
Inducer_unit = '%' # Enter only (10**-6, 10**-9 or '%')
OptInhibition = None
SystemOpt = 'ConstInd'
```

Constitutive System:

```
22    CS = ConstitutiveSystem()
23
24    ###############################################################
25    # User Input
26
27    # Input_filename: The input data file in .csv following the sequences below:
28    #
29    # Single Constitutive Dataset: [Time(min) | Promoter1 | SDPromoter1 ]
30    # Multiple Constitutive Dataset with fixed Promoters ( <= 6 datasets):
31    # [Time(min) | RBS1 | RBS2 | RBS3 | SDRBS1 | SDRBS2 | SDRBS3 ]
32    # Multiple Constitutive Dataset with fixed RBSs ( <= 6 datasets):
33    # [Time(min) | Promoter1 | Promoter2 | Promoter3 | SDPromoter1 | SDPromoter2 | S
34    #
35    # SD refers to the standard deviation data
36    #
37    # NumState: Number of Data Set (excluding standard deviations)
38    # SystemOpt: Options for Constitutive system ('SingleConst', 'MultiFixedRBS', 'M
39    #
40    # Note: the SystemOpt is case-insensitive!
41    ###############################################################
42
43    ### User Input (Data FileName and Number of Data Set in the file)
44#    Input_filename = 'Constitutive_p66.csv'
45#    NumDataSet = 1
46#    SystemOpt = 'SingleConst'
47
48    ### multi data set (varied Promoters with fixed RBS)
49    Input_filename = 'Constitutive_4xmultiFixRBS.csv'
50    NumDataSet = 4
51    SystemOpt = 'MultiFixedRBS'
```

Logic Gate System:

```
20    LGS = LogicGatesSystem()
21
22    ###############################################################
23    # User Input
24
25    # Input_filename: The input data file in .csv following the sequences below:
26    #
27    # NOT gate: [Time(min) | state=0 | state=1 | SDstate=0 | SDstate=1]
28    # AND gate: [Time(min) | state=00 | state=01 | state=10 | state=11 | SDstate=00
29    # OR gate: [Time(min) | state=00 | state=01 | state=10 | state=11 | SDstate=00 |
30    #
31    # SD refers to the standard deviation data
32    #
33    # NumState: Number of Data Set (excluding standard deviations)
34    # SystemOpt: Option for Logic gate system ('NOT', 'AND', 'OR')
35    #
36    # Note: the SystemOpt is case-insensitive!
37    ###############################################################
38
39    Input_filename = 'LogicGate_NOTTop10d30LB_1.csv' #NOT gate (Data set 1)
40    NumState = 2
41    SystemOpt = 'NOT'
```

- Click on  Run button at the toolbar
- Refer to Section 1.6 to set the graphic specification for plotting the gene circuit.

Chapter 2: Cross-platform Implementations: Model Analysis Tools

To verify the interoperability and functional extensibility of the BMSS platform, the generated SBML file encoding the computational model for genetic parts/circuits can be loaded into other existing model-based tools such as iBioSim and COPASI. These open-source software offer modelling and simulation environments coupling with design and analysis features.

2.1. iBioSim: A tool for modelling, analysis, and design of genetic circuits

iBioSim is a computer-aided design (CAD) tool for modelling (users input parameter values or part details from other externally linked databases), design and analysis of genetic circuits.

2.1.1. Installation

Users may install the release beta version of iBioSim-3.0.0 developed by Myers Research Group for Windows, Mac or Linux Operating system. More guidelines for installing iBioSim can be found in the link below:

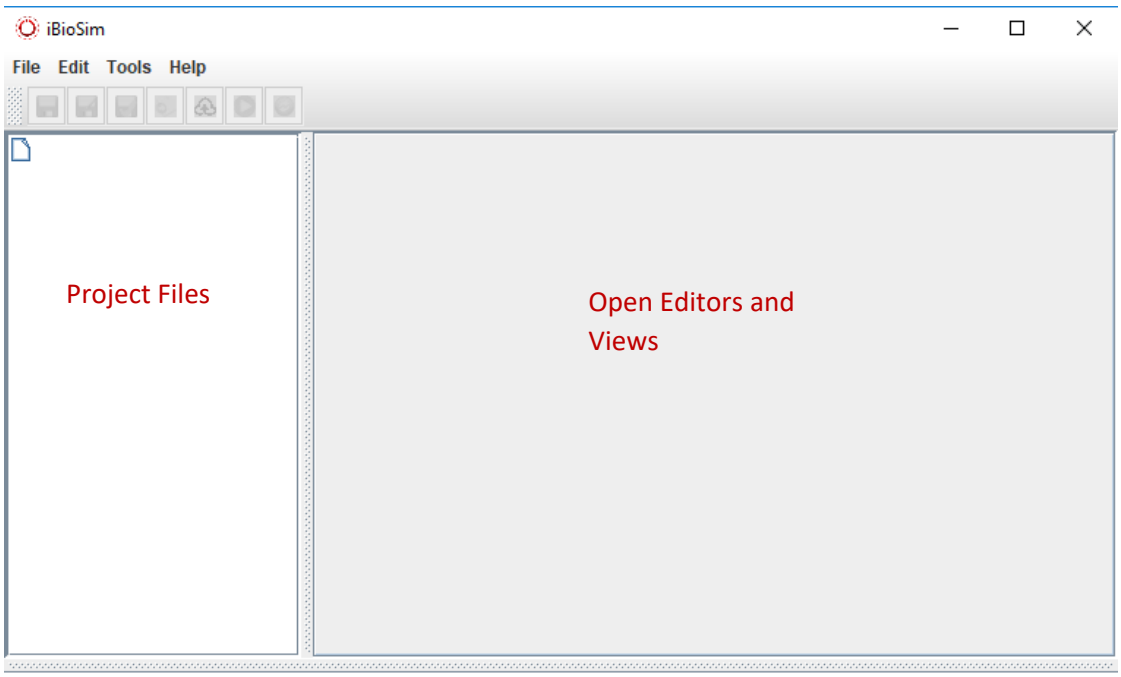
https://github.com/MyersResearchGroup/iBioSim

For windows users:

In order to run the iBioSim.bat file, users are required to install Eclipse IDE for Java, and also Java Runtime Environment (JRE) to run the Java program. The same applies for Mac OS and Linux OS to run their corresponding iBioSim scripts in mac64 or linux64.

In the downloaded iBioSim folder, users may refer to all documentation or tutorial files available in the docs subfolder.

To begin, from the downloaded iBioSim folder, double click on the iBioSim executable file, which is the iBioSim.bat file for Windows OS. A command prompt will pop out and iBioSim Graphical User Interface (GUI) will be displayed as shown below. Close other unnecessary pop out windows or just follow the default settings and click ok.

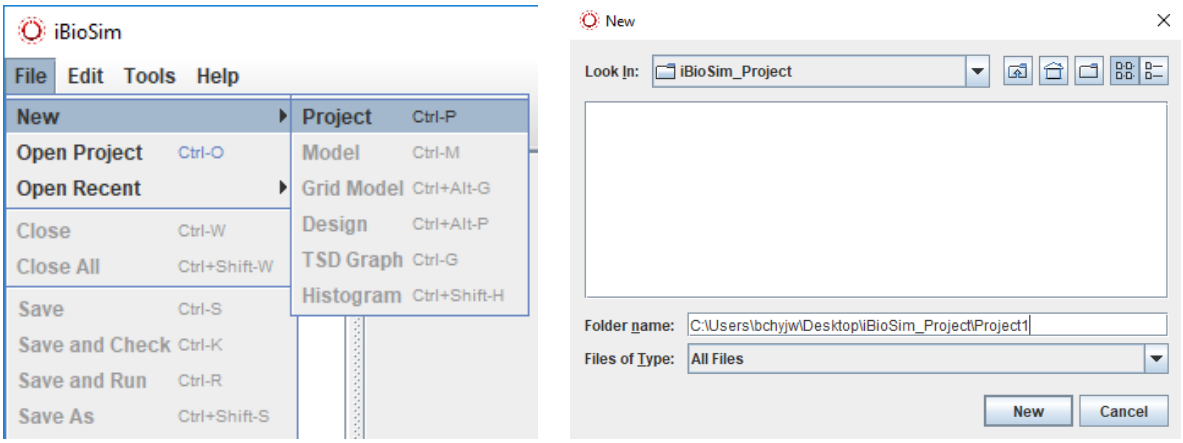


Note: If the iBioSim GUI never appears after a while. Please restart your computer once and retry again. The changes might only be applied after restarted.

2.1.2. Creating a New Project

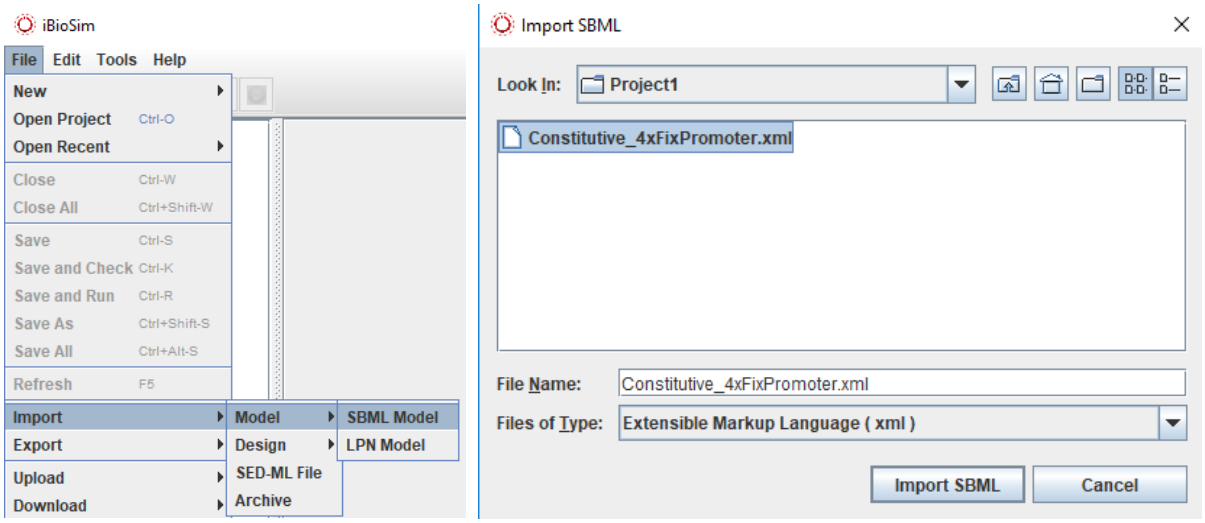
Before starting a new project,

- Create an iBioSim_Project folder in your working directory (e.g. Desktop, Document, etc.)
- To create a new project, click on **File** menu, select **New → Project**.
- You will then be prompted to browse to a project directory. At the **Look In** menu, double-click on the iBioSim_Project folder that you have created.
- At the **Folder name**, add a new subfolder known as Project1 (e.g. absolute path…\iBioSim_Project\Project1) inside the iBioSim_Project folder
- Then click on the **New** button to create the new project as shown below.

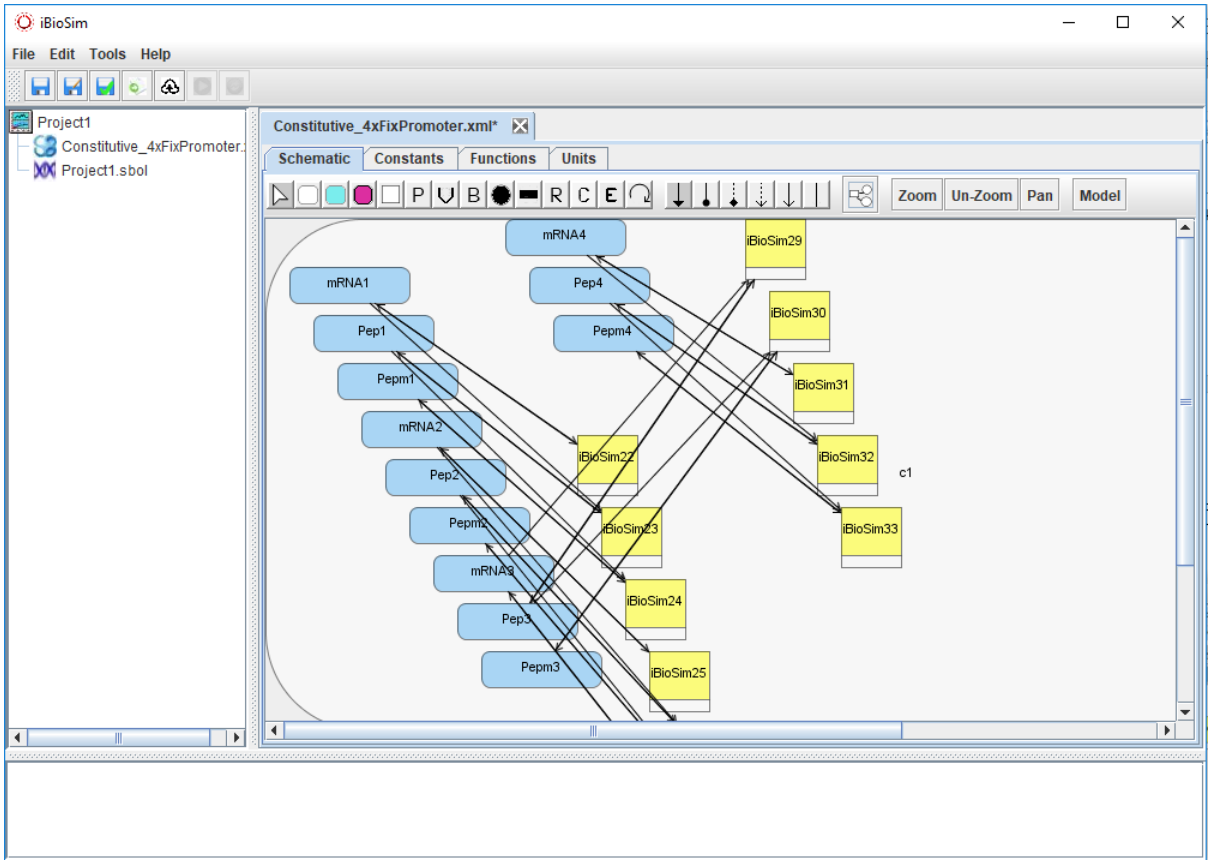


2.1.3. Importing SBML Model File

- Copy and paste the BMSS-generated SBML file (e.g. Constitutive_4xFixPromoter.xml) into the newly generated Project1 subfolder.
- Under **File** menu, select **Import → Model → SBML Model**
- In the Import SBML window, browse to the Project1 directory and select the .xml file
- Click on the Import SBML button to load the SBML into iBioSim

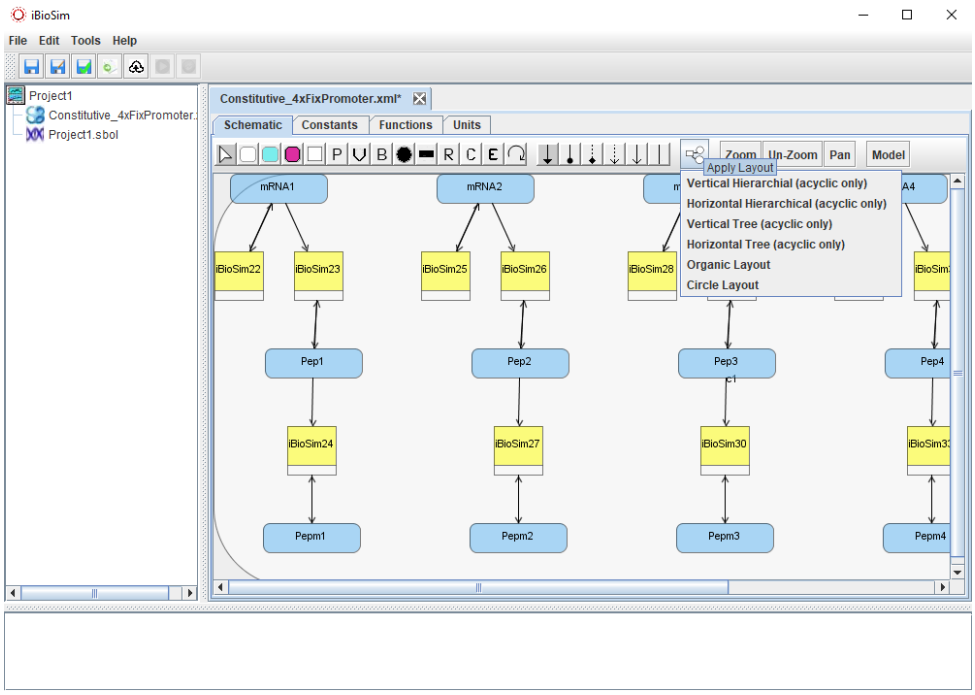


Note: Click OK for any validation problems found unless there is an error. The schematic diagram for the model is illustrated as below:

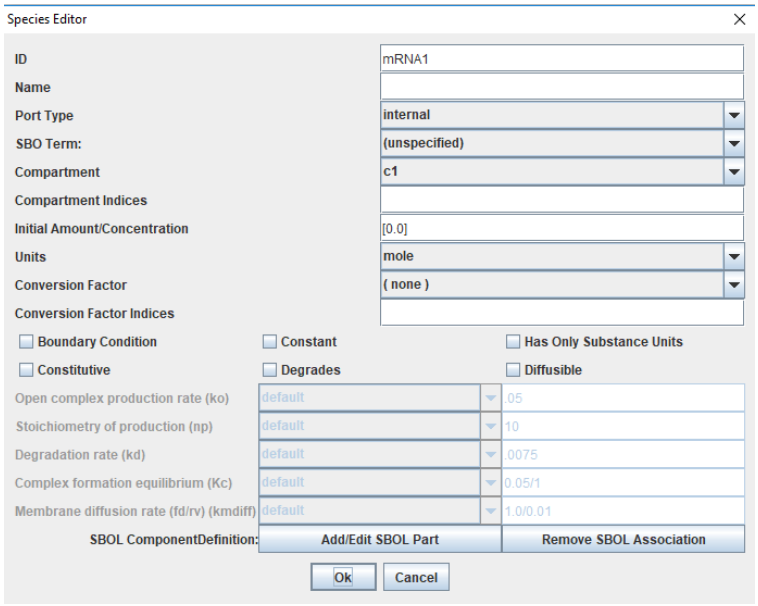


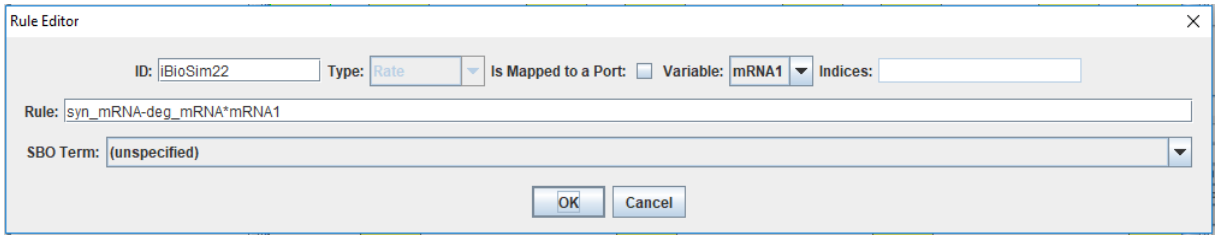
To organize the network layout:

- Click on the **Apply Layout** and select **Vertical Hierarchial (acyclic only)** to reorganize the displayed layout
- Click **File → Save** to save the modified layout. The original imported SBML (.xml) file will be updated to include the model layout, which could be useful to be used later in COPASI software.



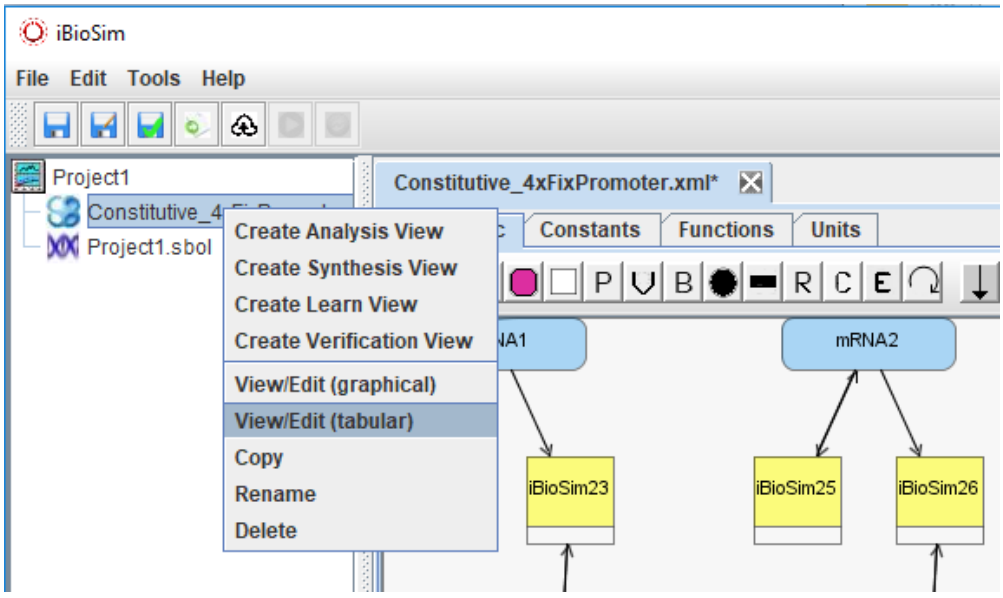
- mRNAX, PepX, and PepmX in this example model refer to the species ID for mRNA, Peptide/Protein, and matured Peptide/Protein. The rate rules linked to the species or between the species describe the corresponding ordinary differential equation (ODE).
- Double click on the species or rate rule to open the editor for viewing or modification as shown below.



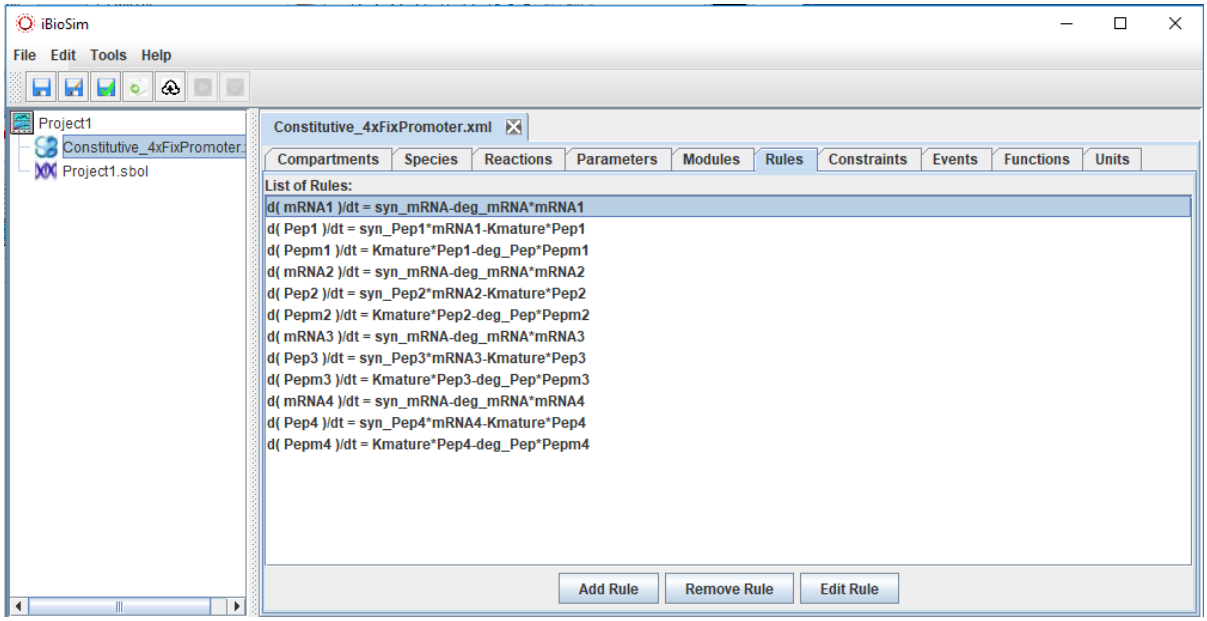


To view or edit the model details,

- right click on the .xml file at the left panel (project tree)
- select View/Edit (tabular)



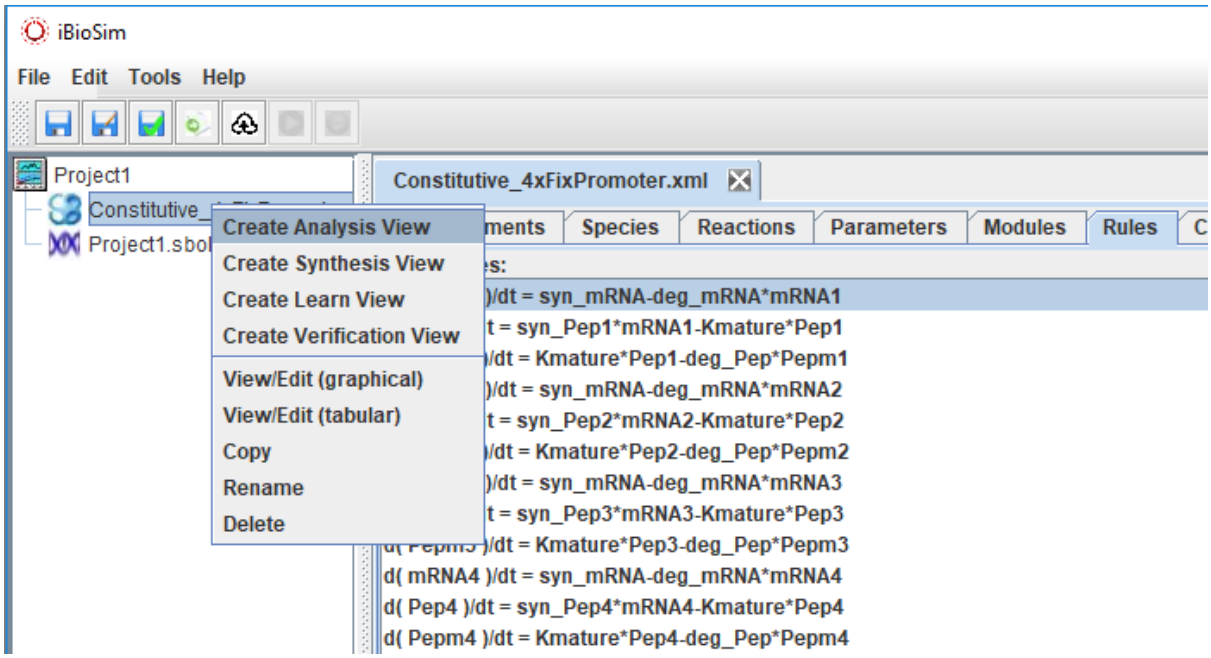
- The initial conditions for the different species are listed under **Species** Tab; all the model parameters are available under the **Parameters** Tab; and a full list of the Rate rules (ODEs) could be found under the **Rules** Tab. Double click to view and/or edit the component of interest



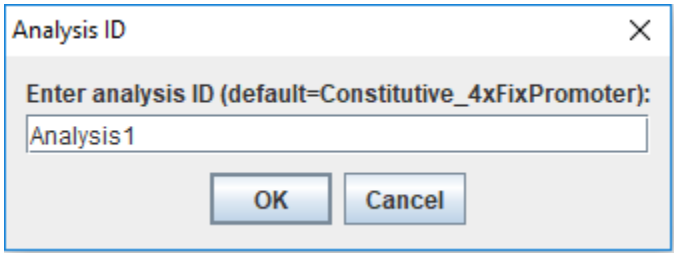
2.1.4. Running Model Simulations

To run the model simulation:

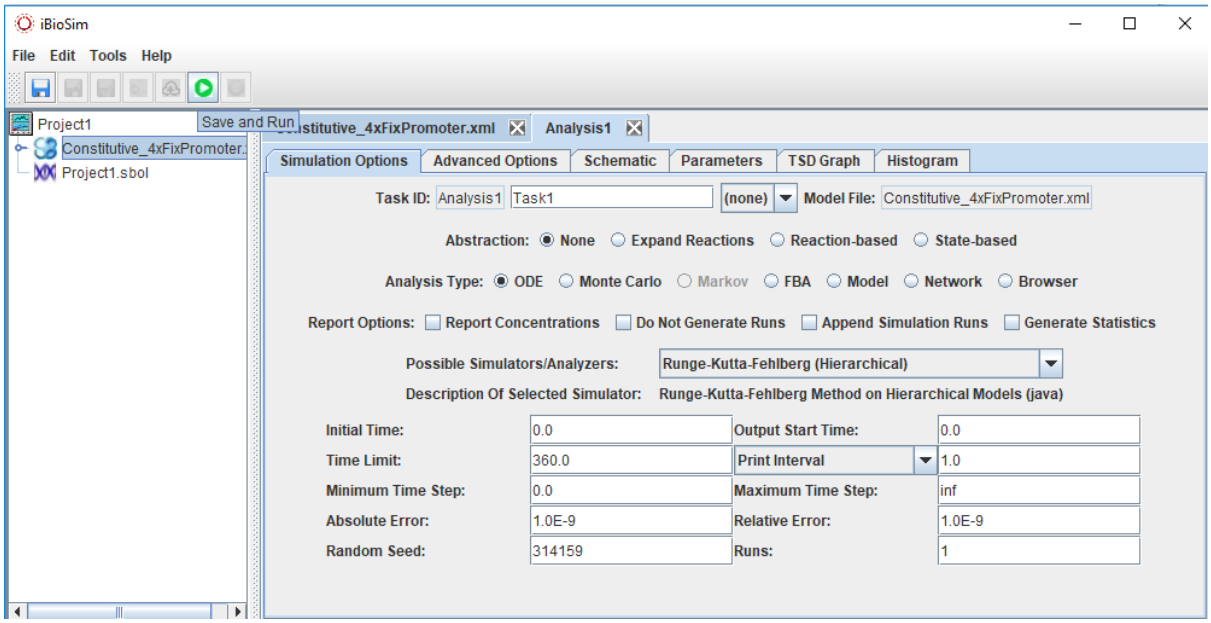
- Right click on the .xml file at the left panel
- Select Create Analysis View
- Enter Analysis 1 on the pop-up Analysis ID window
- Click OK



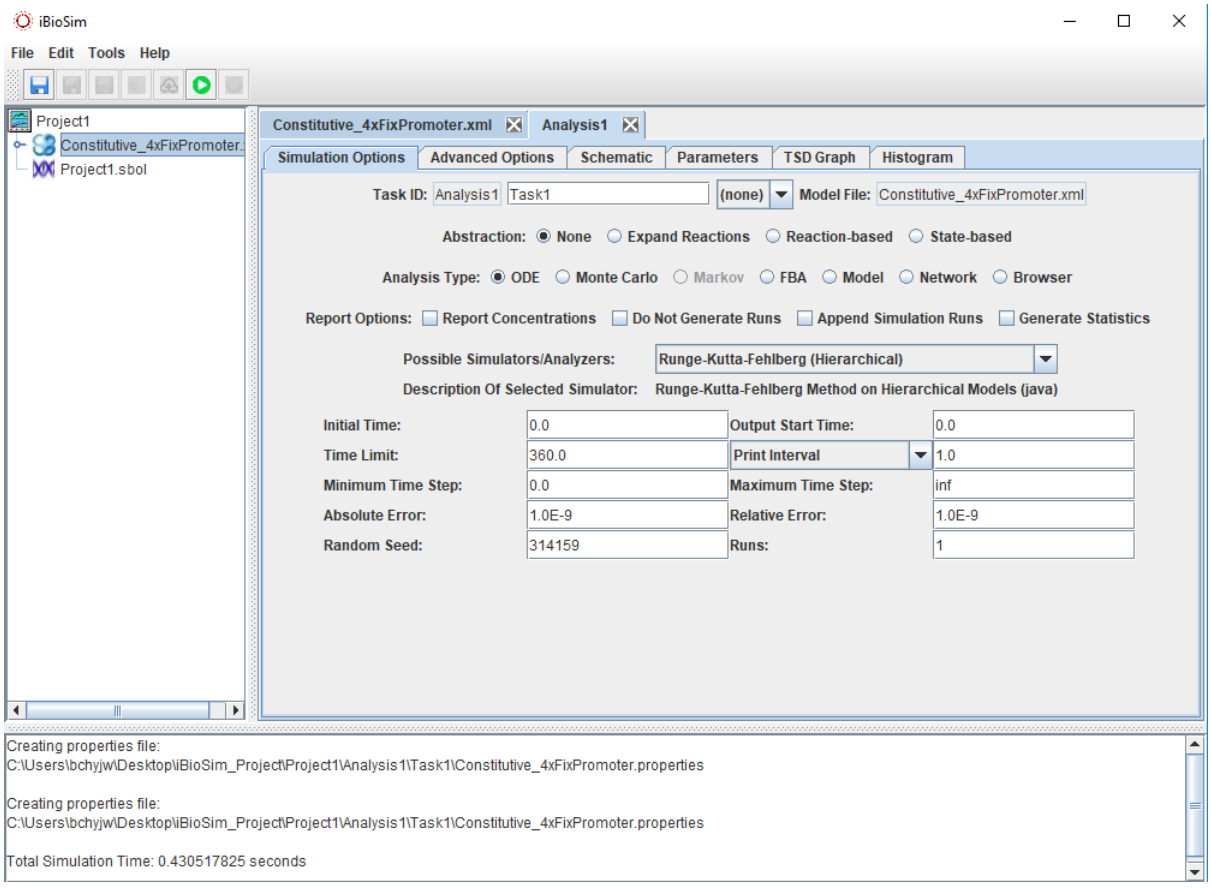
- Enter an Analysis ID (e.g. Analysis1)



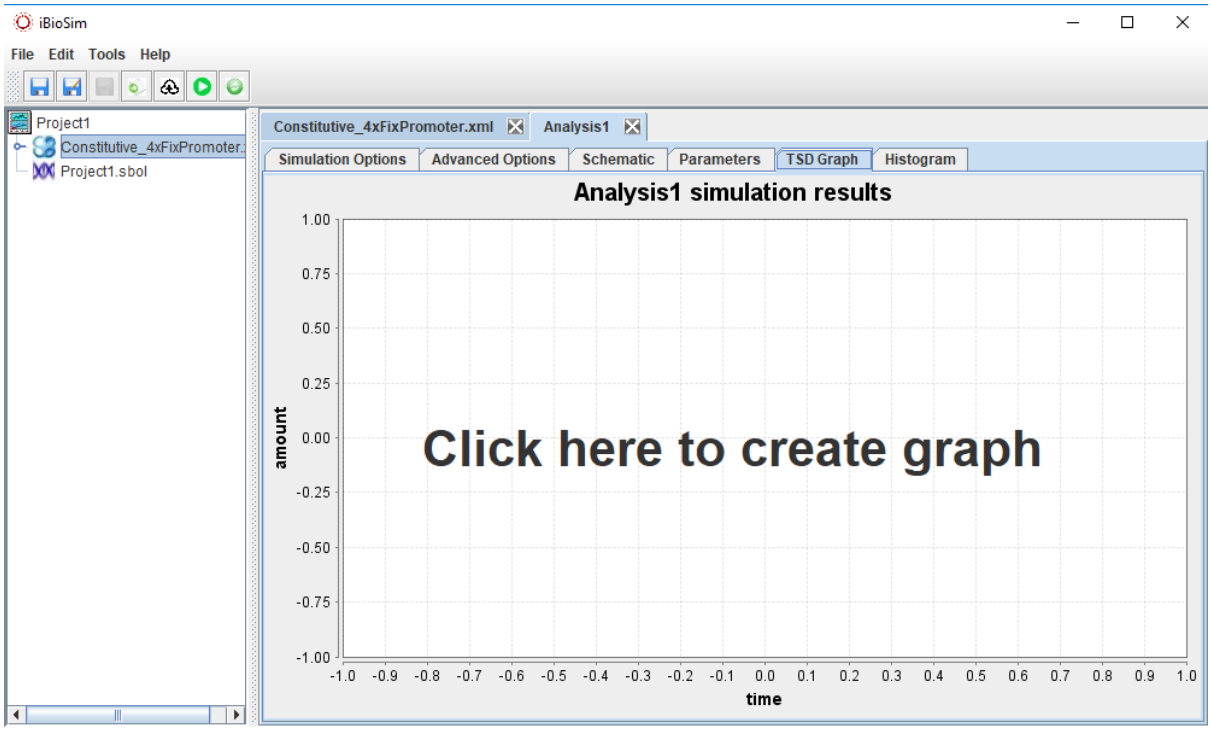
- Under the **Simulation Options** Tab, Enter the **Task ID** (e.g. Task1) under Analysis1
- Select **Analysis Type** of **ODE** to perform continuous-deterministic simulations.
- Choose the appropriate **Possible Simulators/Analyzers**, which are the numerical methods to solve the ODEs. Use the Default setting of **Runge-Kutta-Fehlberg (Hierarchical)**.
- Adjust the **Time Limit** to match your simulation duration. For the SBML file generated by BMSS, the unit of time is in minute.
- Click on the **Save and Run** green button on top to execute the simulation.



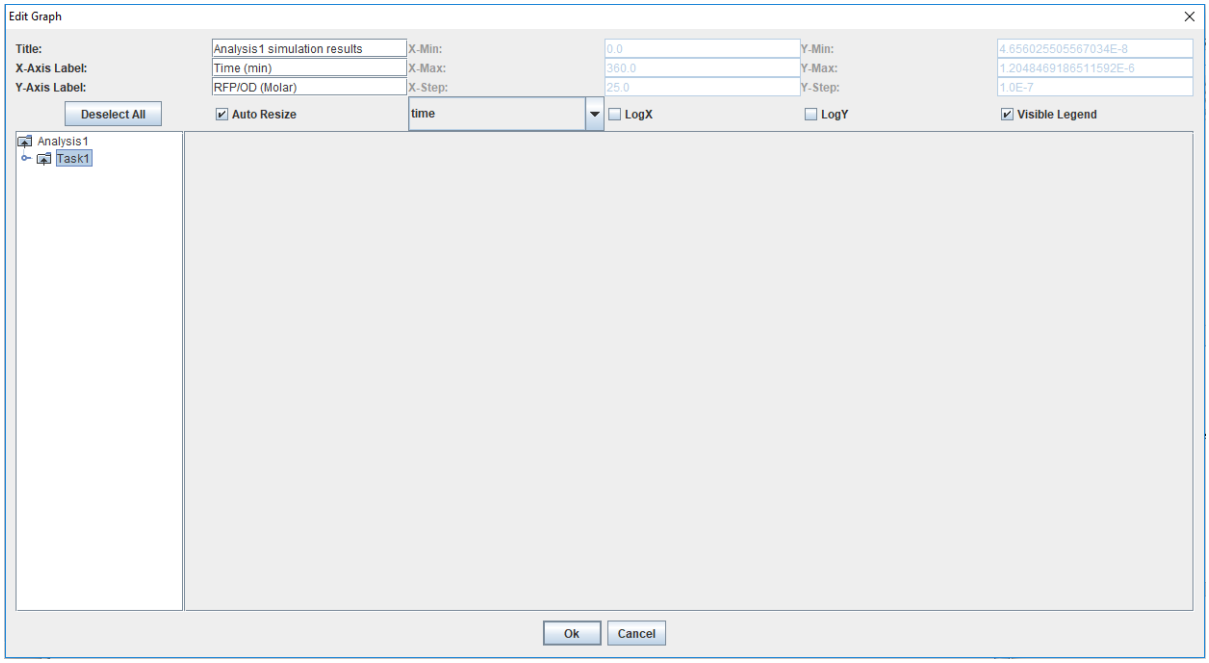
- If the simulation runs successfully, there will be no error message displayed on the Command Log at the bottom, and the Total Simulation Time will be shown.



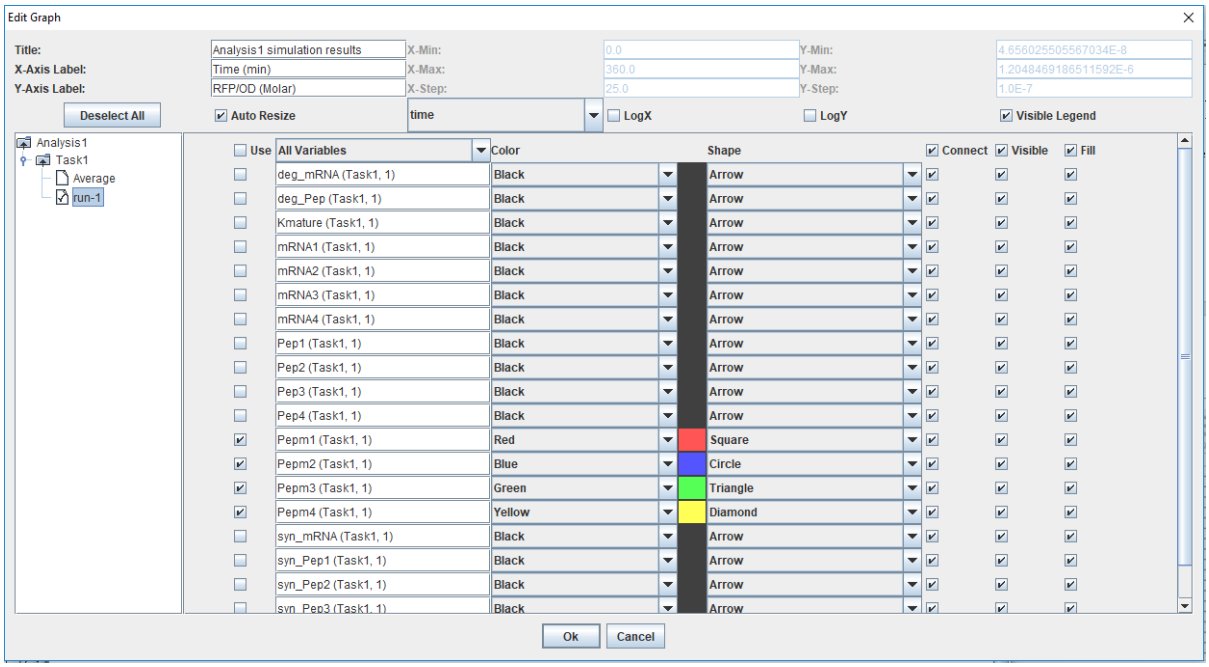
- Under the **TSD Graph** Tab, which refers to Time-Series Data, click on the centre or the graph to open the **Edit Graph** window



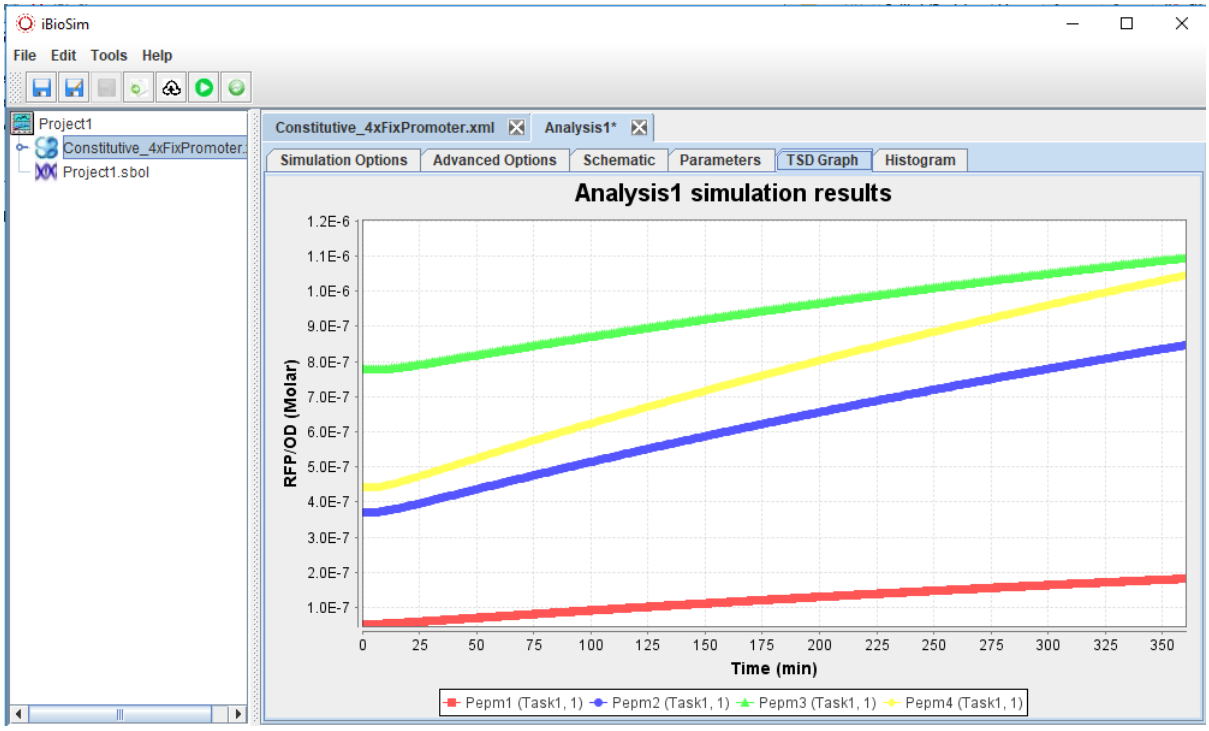
- Users can modify the Figure features such as Title, X-Axis and Y-Axis Labels as displayed below:



- Double click on the **Task1** on the left panel under **Analysis1**
- Click on **run-1** to display all the variables and parameters of the model
- Click on the tick-box to select the variables to be plotted on the TSD graph (e.g. Pepm1-Pepm4)
- Users may modify the **Color** of the line and the **Shape** of the plotted symbols.
- Then click **OK**

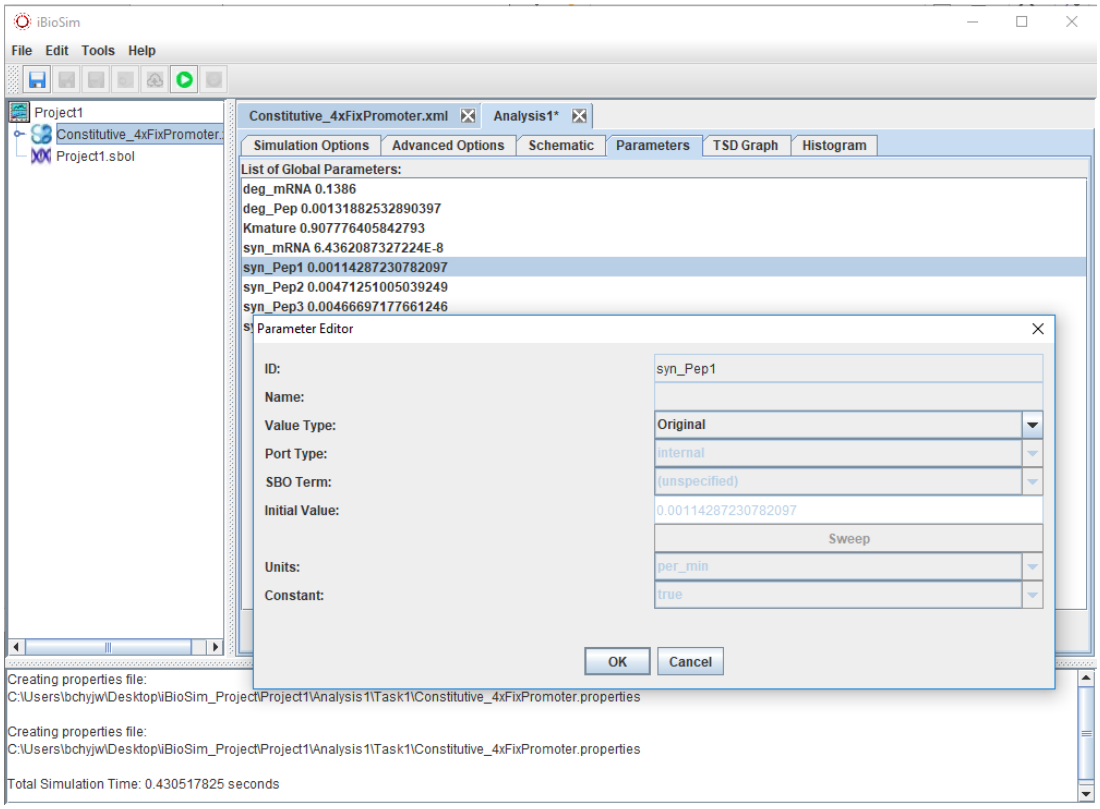


- The time-series simulation results are shown as follows for the four different constitutive promoters-driven RFP/OD expression profiles:



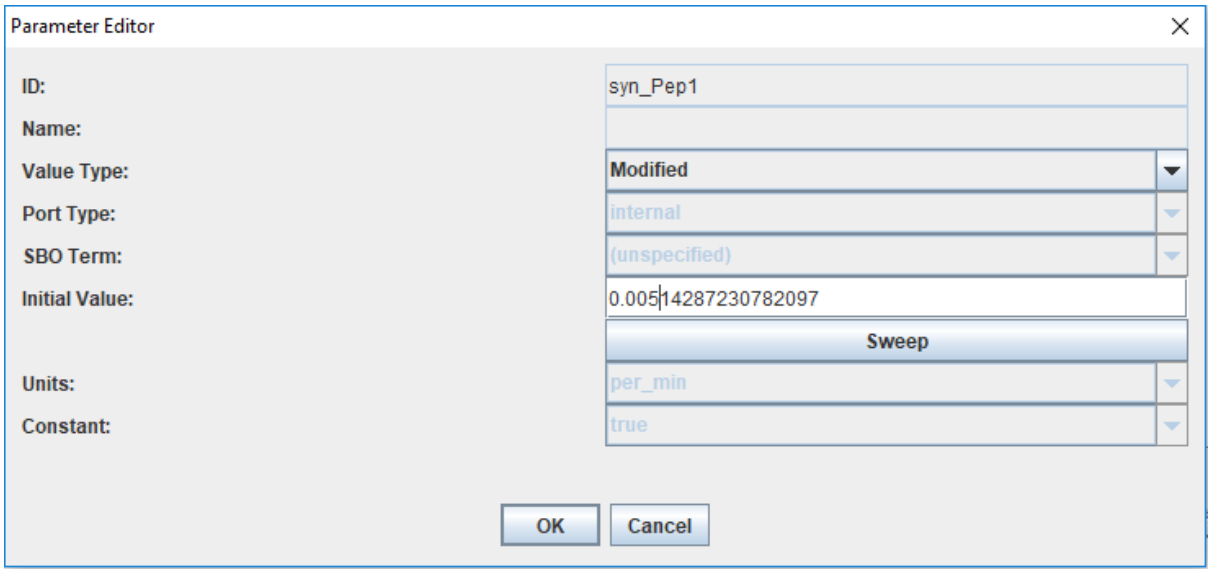
2.1.5. Modifying Model Simulations/Parameters

- To modify the model parameters, shift to the **Parameters** tab,
- Double click on the parameter to be changed (e.g. syn_Pep1) to open the **Parameter Editor**

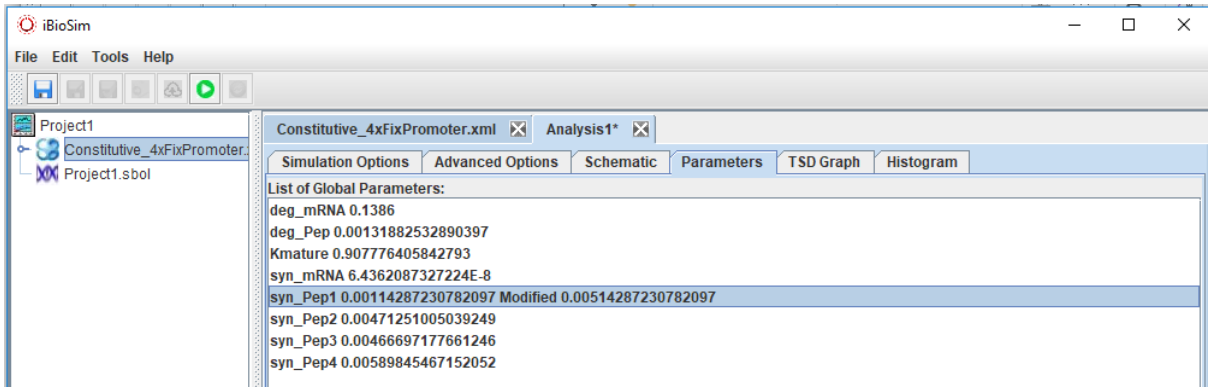


- Change the **Value Type** from **Original** to **Modified**

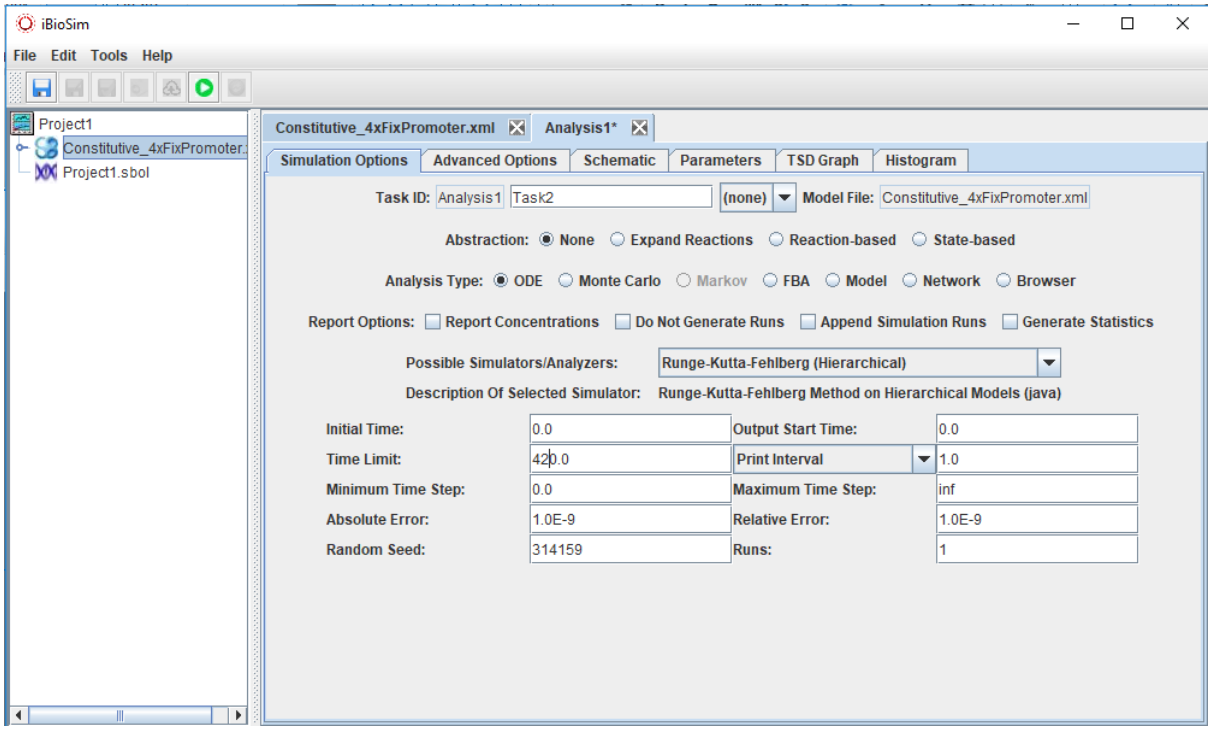
- Modify the **Initial Value** to the value of interest (e.g. increase the peptide synthesis rate from 0.00114 min-1 to 0.00514 min-1)
- Click **OK**



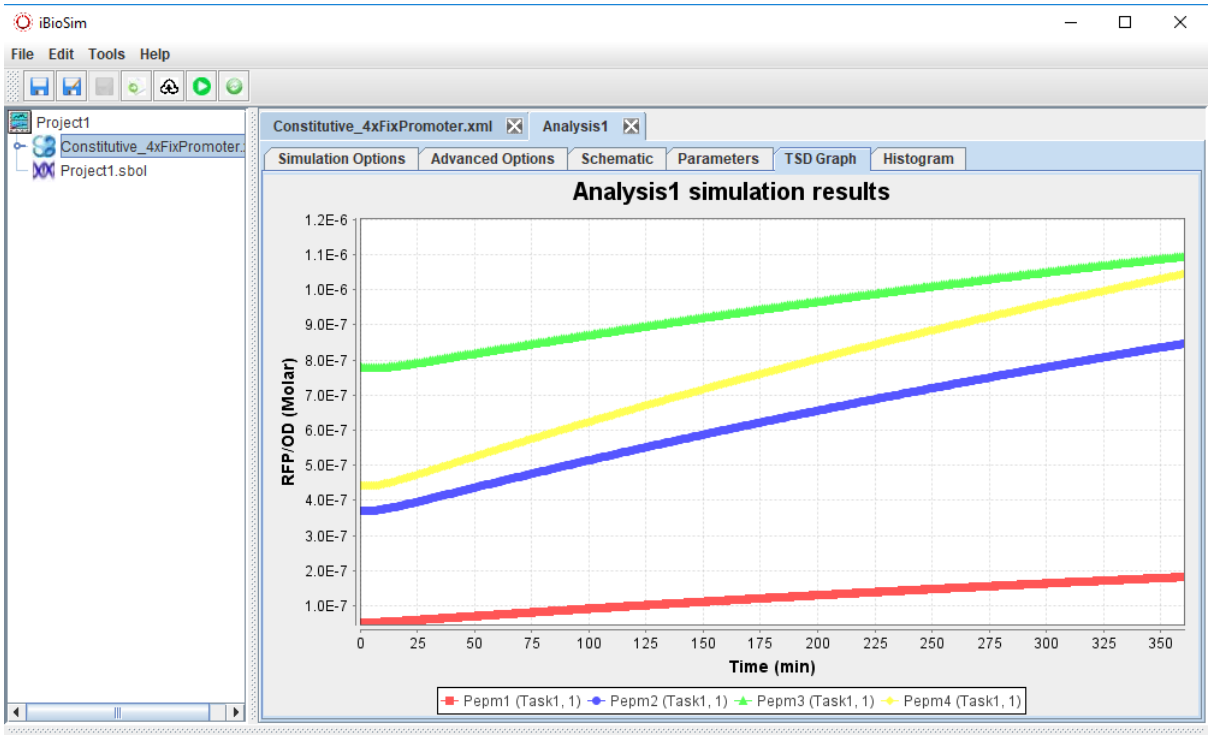
- The original and the modified values for the parameter syn_Pep1 will then be listed in the **Parameters** tab.



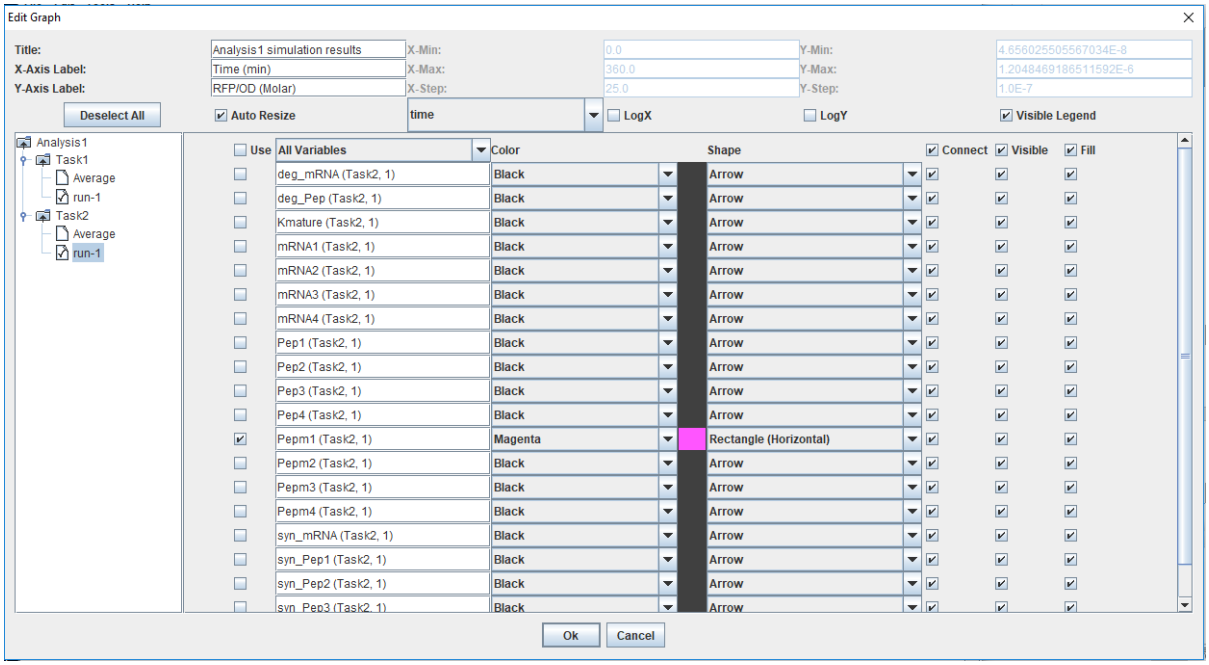
- Go to the **Simulation Options** tab
- Change the **Task ID** to Task2
- Users can also modify the **Time Limit** (e.g. from 360 min to 420 min) for the new simulation
- Click on the **Save and Run** green button at the top to execute the simulation process



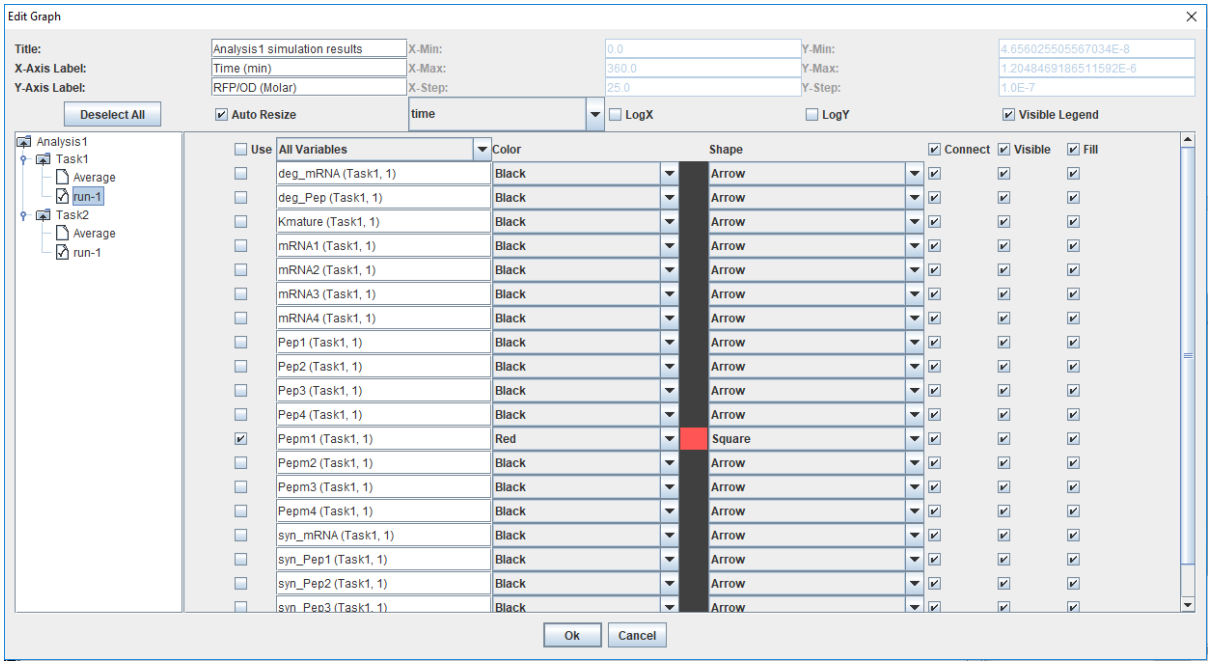
- When there is no error displayed at the **command Log**, the run is successful
- Go to the **TSD Graph** tab, the results from previous run are still on display.
- To modify the graph, click on the centre of the figure to open the **Edit Graph**



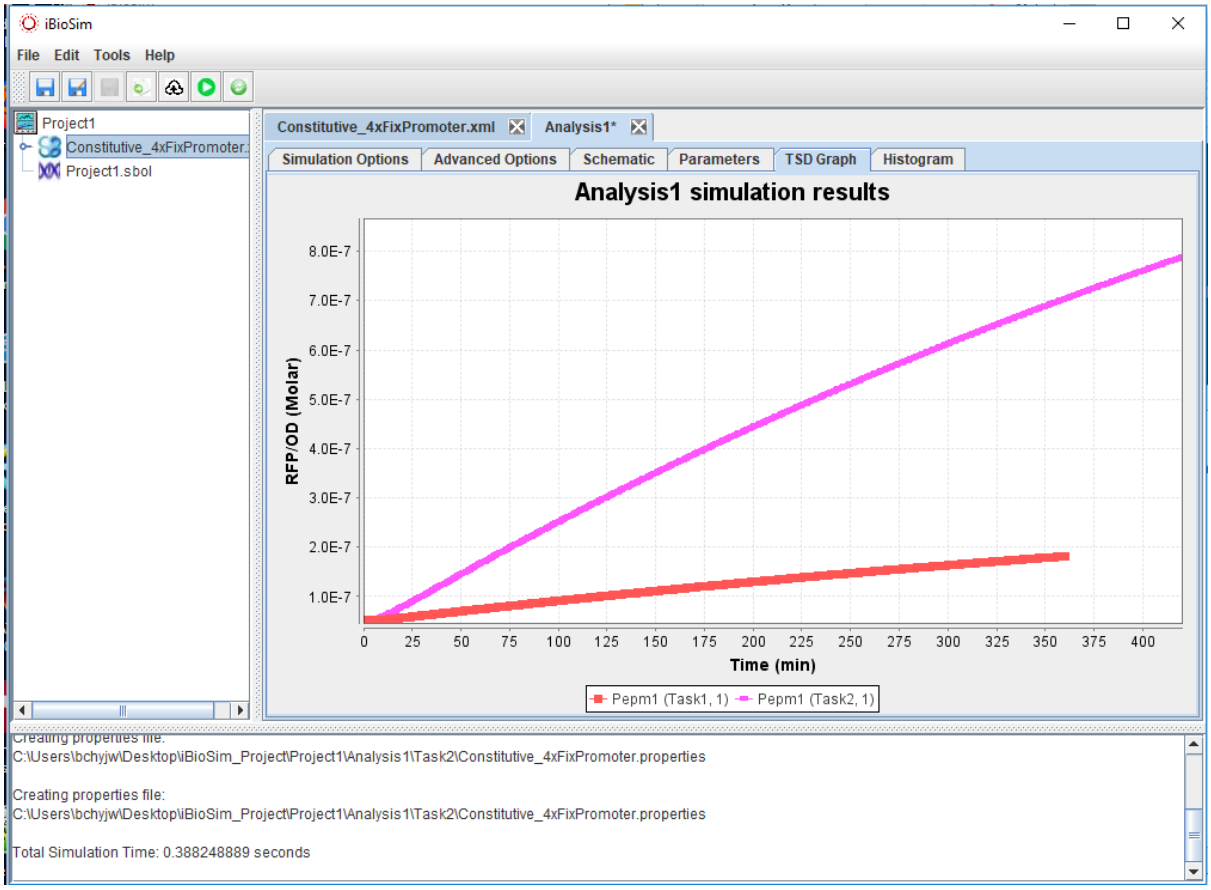
- There will be two tasks: Task1 and Task2, listed under the Analysis1
- Double click on the Task2 and click on run-1
- Click on the tick-box for Pepm1 to plot the modified simulation result after changing the parameter syn_Pep1.



- Users can also click on the **Task1 → run-1** to check the selected variables to be plotted. The variables selected from previous Task1 run will be included by default.
- Users can untick the components other than Pepm1 from Task1 to plot only the Pepm1 variable from Task1 and Pepm1 variable from Task2 for comparison as shown below.
- Click **OK**



- The Pepm1 simulation results from Task1 and Task2 are demonstrated below:
- As can be seen, after increasing the value of syn_Pep1 (protein synthesis rate), there is an increase in the final RFP/OD expression level in comparison with the original parameter value.
- More modifications can be done by creating new **Task ID** under **Simulation Options** Tab, and the same steps can be repeated to get the final simulation results.



- In the Project1 subfolder under iBioSim_Project folder, there is a Project1.sedml file, which encodes information or setup to run the simulations for Task1 and Task2, in conjunction with the results plotting. SED-ML refers to Simulation Experiment Description Markup Language, which is a standard to ensure exchangeability and reproducibility of simulation experiments.

2.2. COPASI: A tool for model simulations and analyses

COPASI is an open-source software application for simulation and analysis of biochemical networks and their dynamics. This standalone program can support models encoded in the SBML standard and can simulate their behaviours using ODEs or Gillespie's stochastic algorithm.

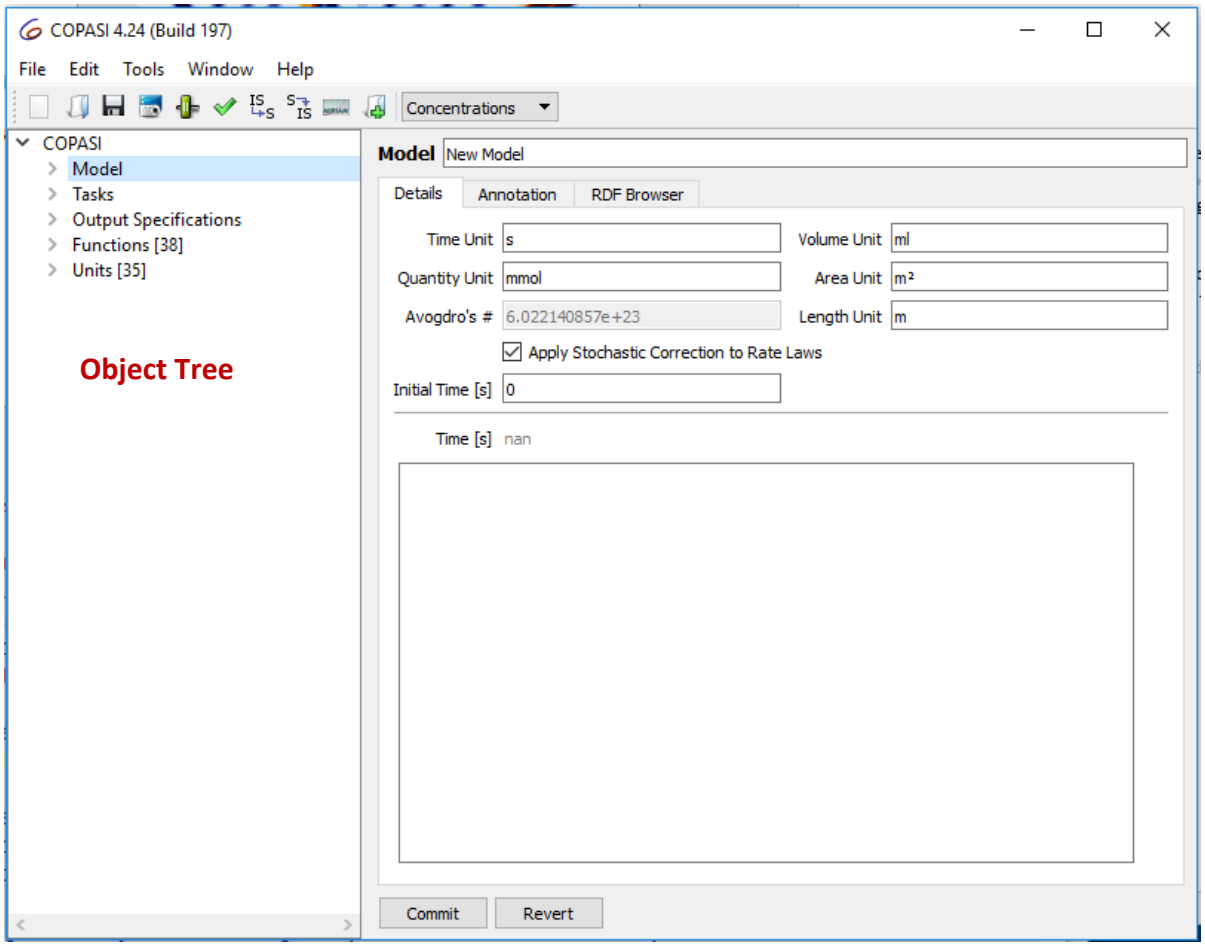
2.2.1. Installation

Download and install COPASI based on your OS platform. Refer to the COPASI Online User Manual for the detailed documentation of the different features available.

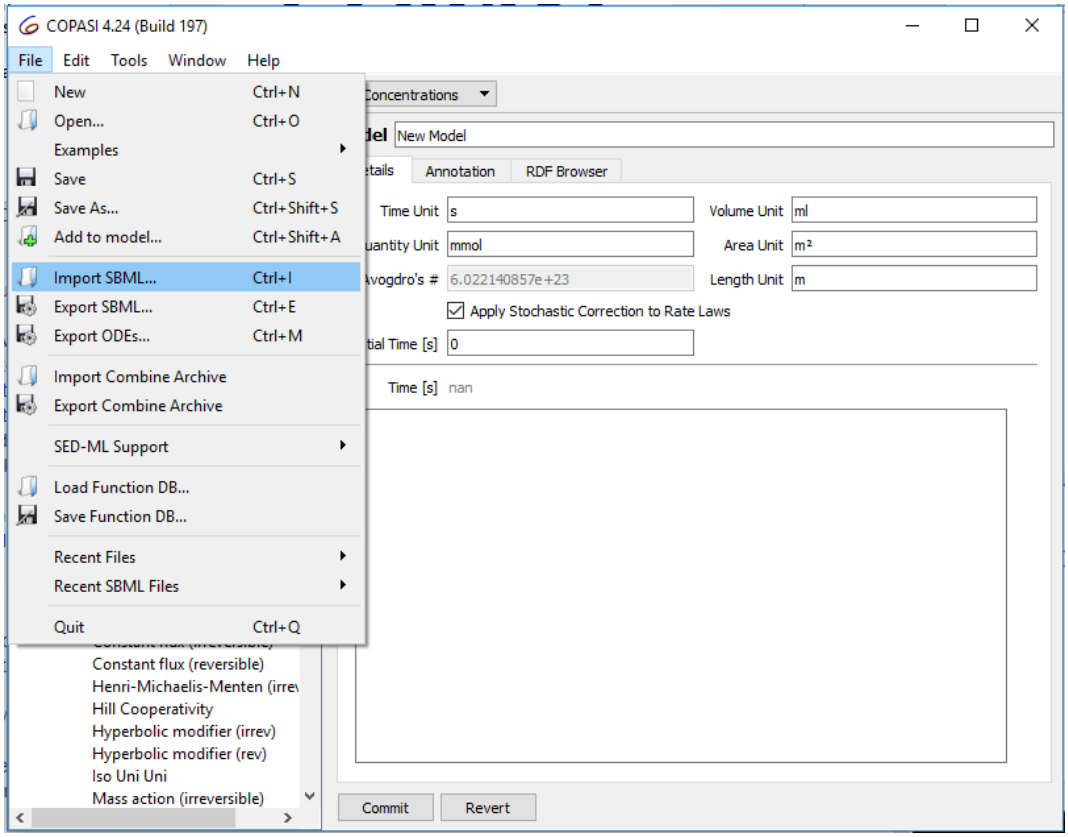
2.2.2. Opening New Project/Loading Model

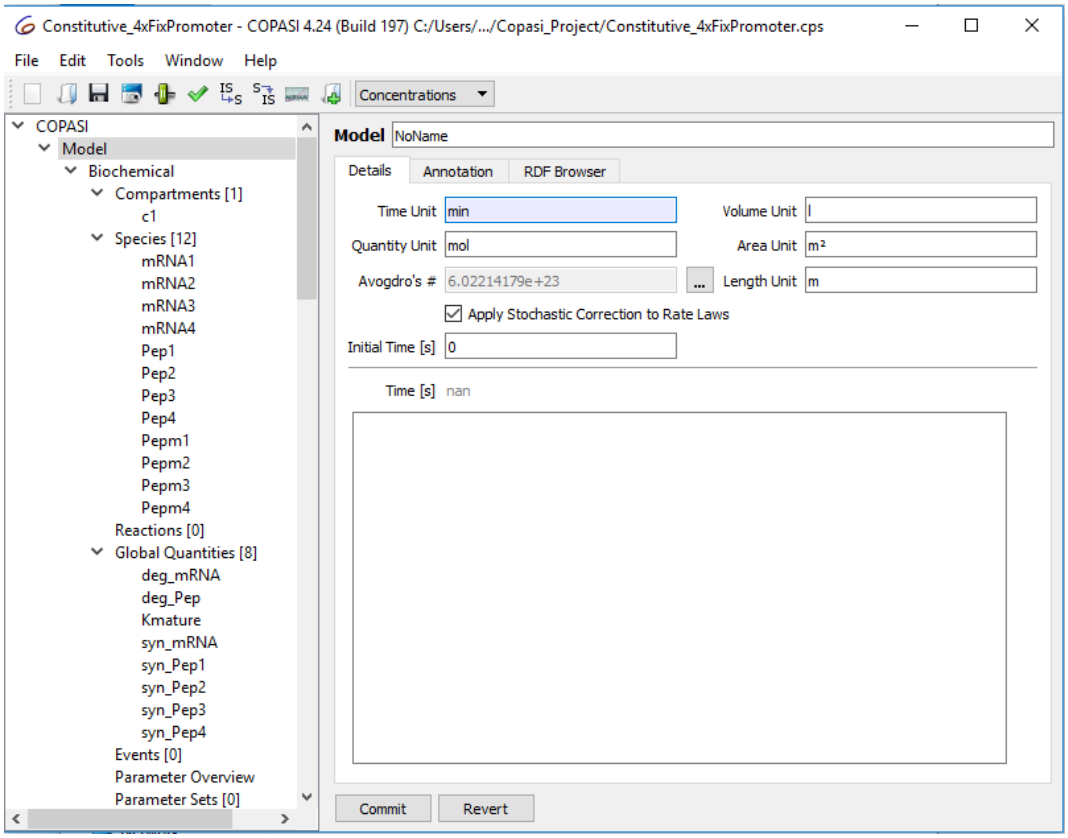
- Open the COPASIUI from the search menu, which is the graphical user interface for COPASI program.
- There are five branches below the root element at the Object Tree.
  - Model: contains all objects/details belong to the existing model
  - Task: all the features that COPASI can execute
  - Output Specifications: to handle the outputs such as result plotting and reports generated.

- Functions: includes all the kinetic functions that are defined as well as functions defined by users.
- Units: contains built-in units for parameters or variables as well as units defined by users.



- Create a folder called Copasi_Project in the working directory
- Copy and Paste the SBML (.xml) file generated from BMSS into the Project folder.
- In order to view the model network layout in COPASI, copy and paste the SBML (.xml) file with layout generated by iBioSim.
- From the **File** menu, select **Import SBML**, browse and open the SBML file.
- Click **OK** for any pop-out warning message

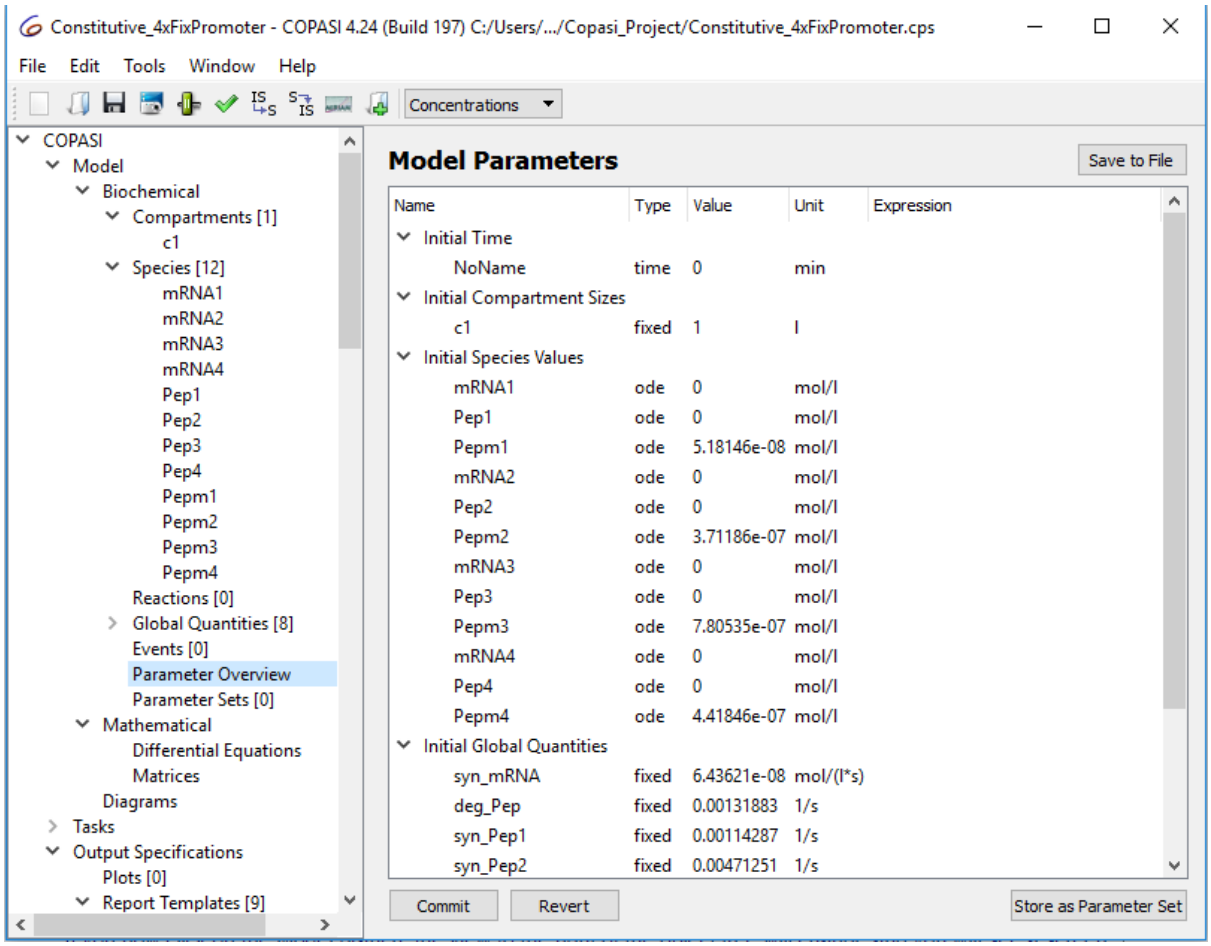


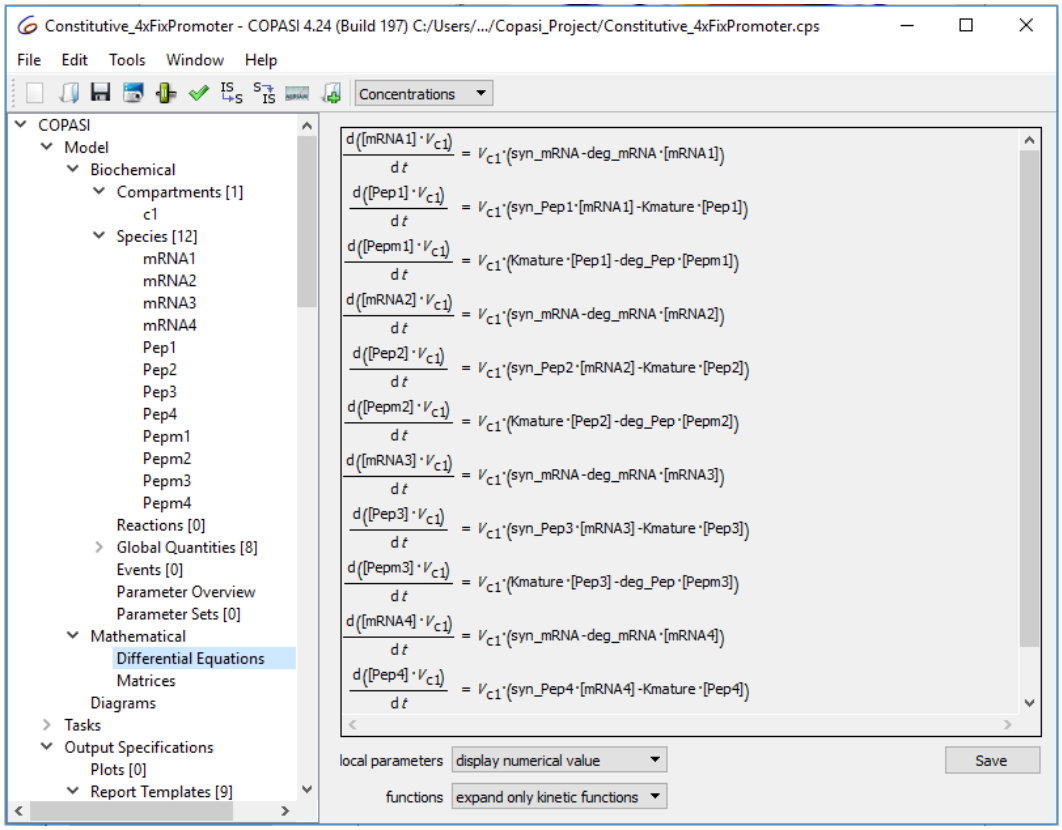


- All the model details will be loaded into the Model object tree displayed at the left panel.
- Change the **Time Unit** from s to min

- Click on the **Biochemical** to view the model **Compartments**, **Species** (Variables), **Global Quantities** (parameters), which are also available for editing or modification.
- An overview of all the variables, parameters, initial conditions, and parameters values is accessible under **Parameter Overview**.
- List of ODEs can be seen under **Mathematical → Differential Equations**.

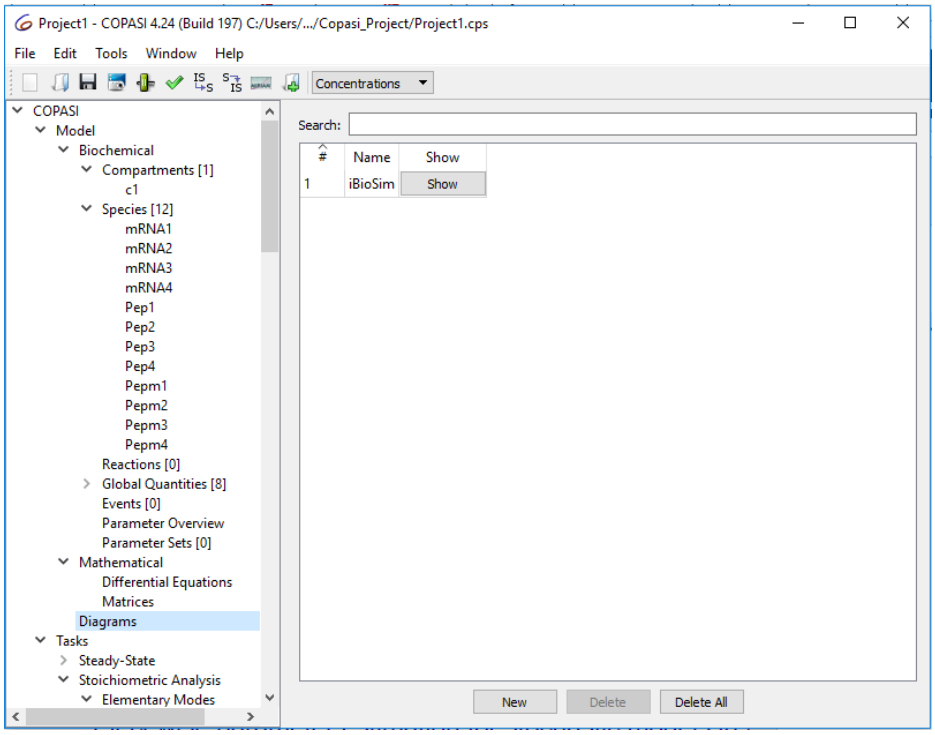
Note: the Unit shown for many of the parameters denoting kinetic rates should be in 1/min instead of the indicated 1/s. This is a limitation of the libSBML library used to generate the SBML files. However, this would not affect the final simulations.



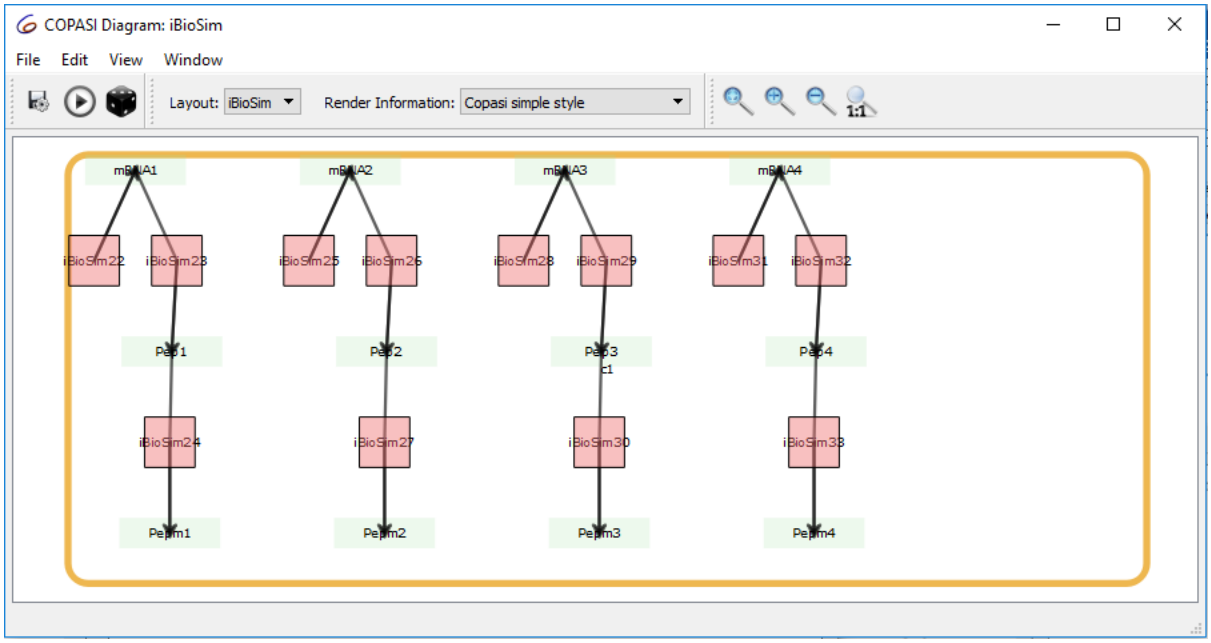


To view the model schematic diagram,

- Click on Diagrams under Model object tree
- There will be one layout from iBioSim
- Click on Show to display the model diagram

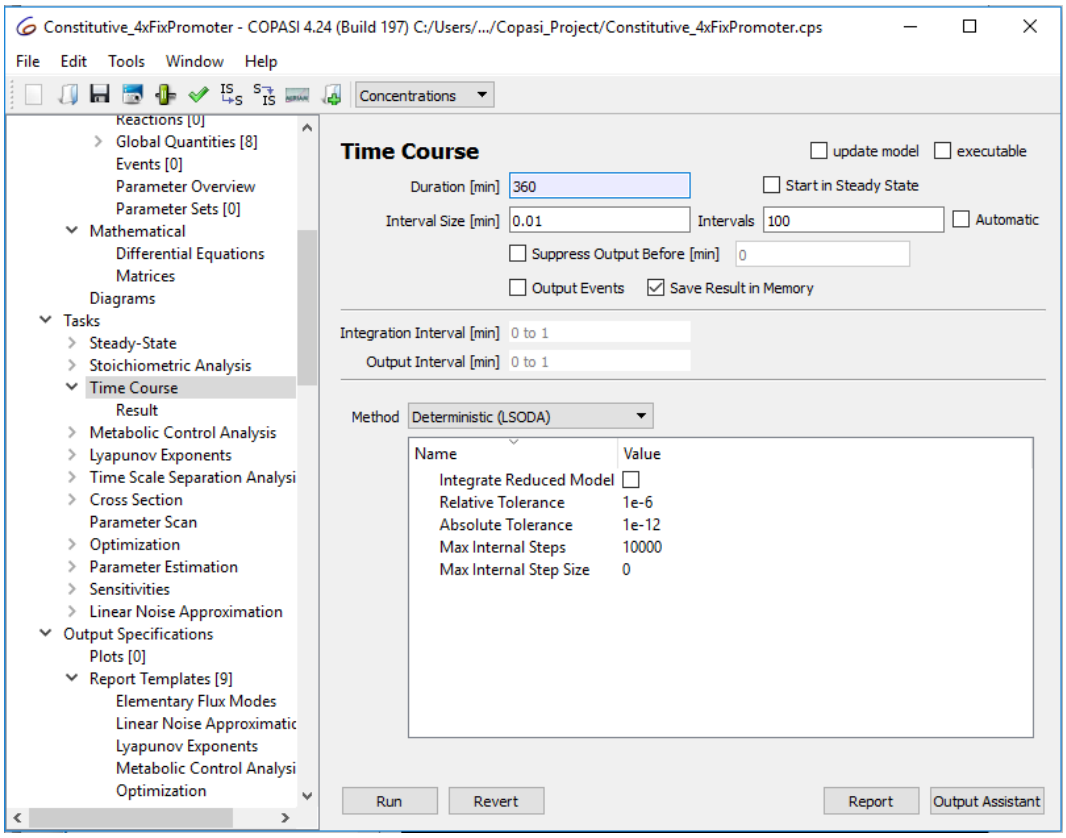


- The iBioSim layout will be displayed as follows:

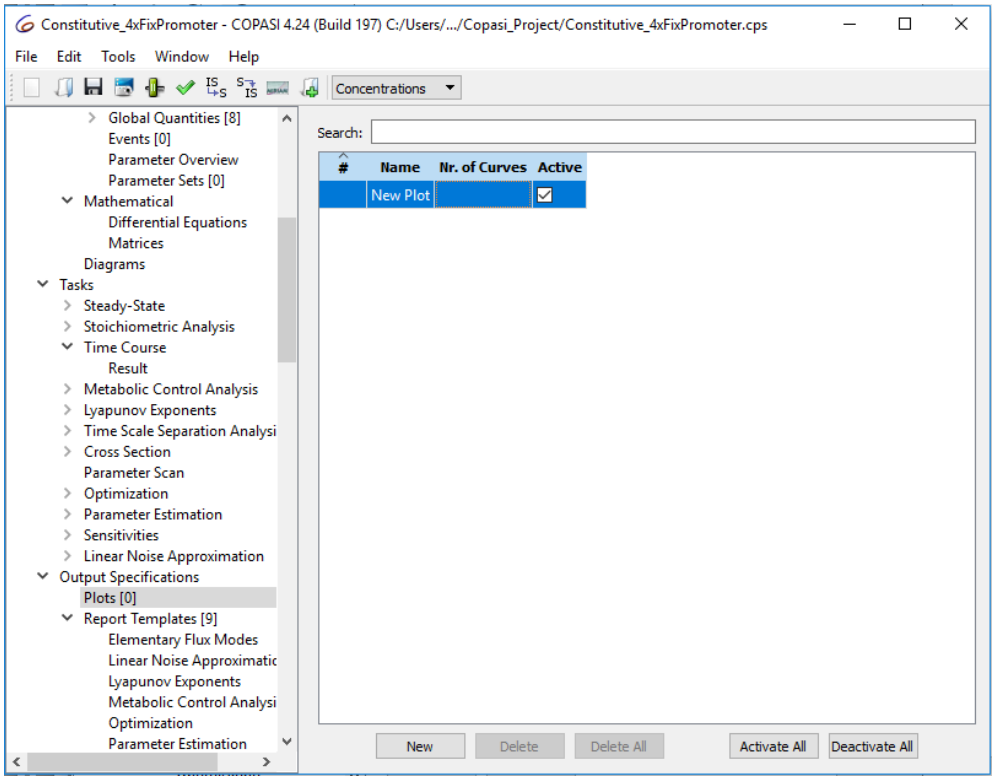


2.2.3. Running Model Simulations

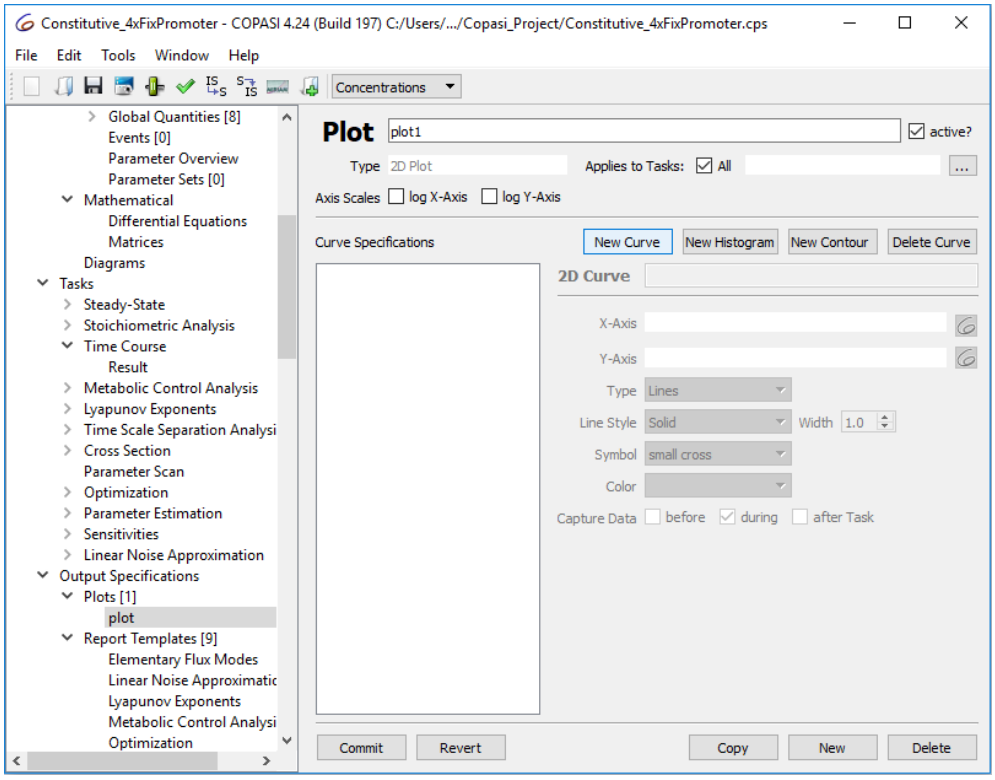
- Under **Tasks** object tree, click on **Time Course**
- Set the Time Course **Duration (min)** to 360 (simulation duration)



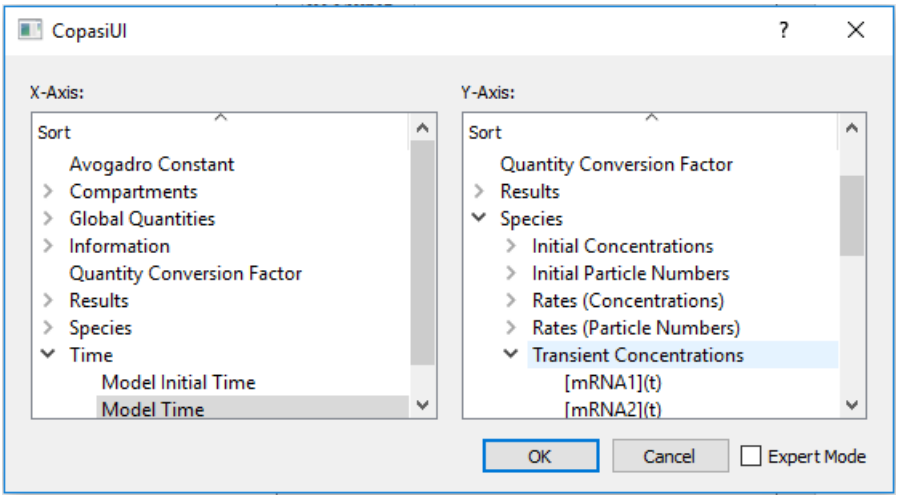
- Before running the simulation, click on Output Specifications
- Double click on New Plot



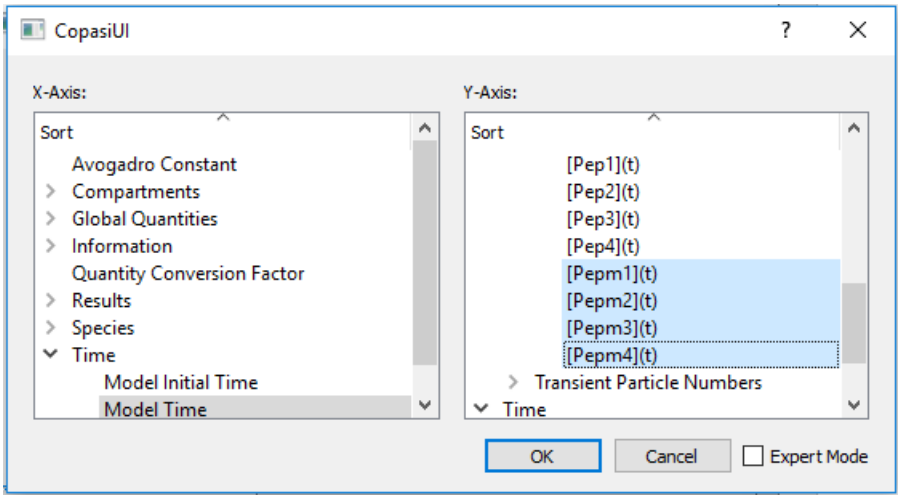
- Change the name of the **Plot** to plot1
- Click on **New Curve** to open a sub **CopasiUI** window to select the components for X-Axis and Y-Axis



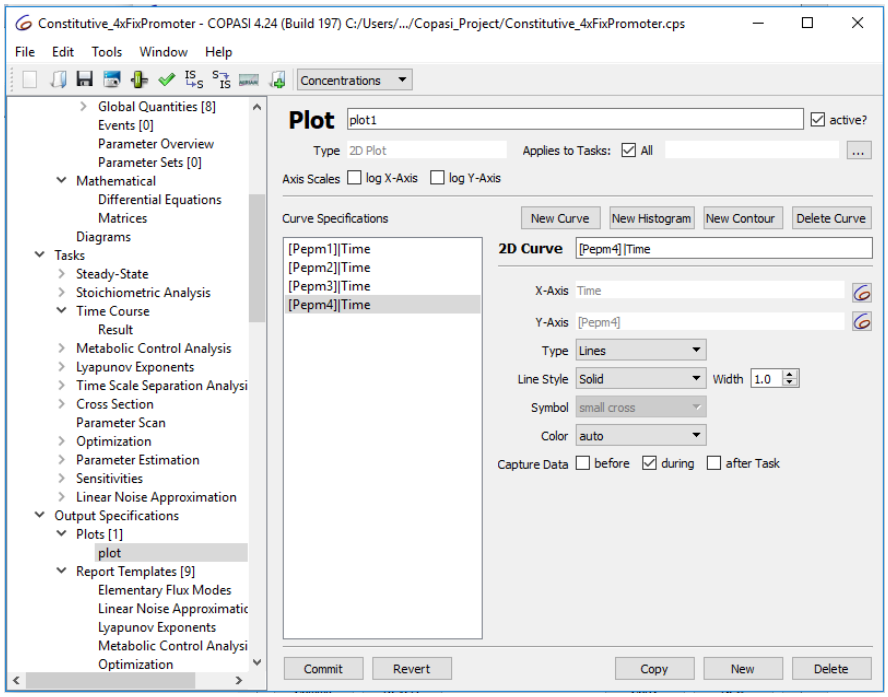
- Under **X-Axis**: Select **Time → Model Time**
- Under Y-Axis: Species → Transient Concentrations



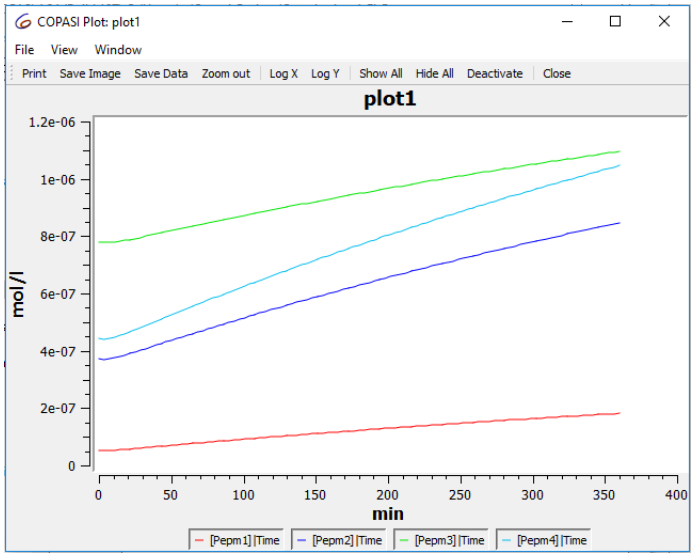
- Press **Ctrl** button on keyboard and select the four [PepmX] variables for plotting
- Click **OK**



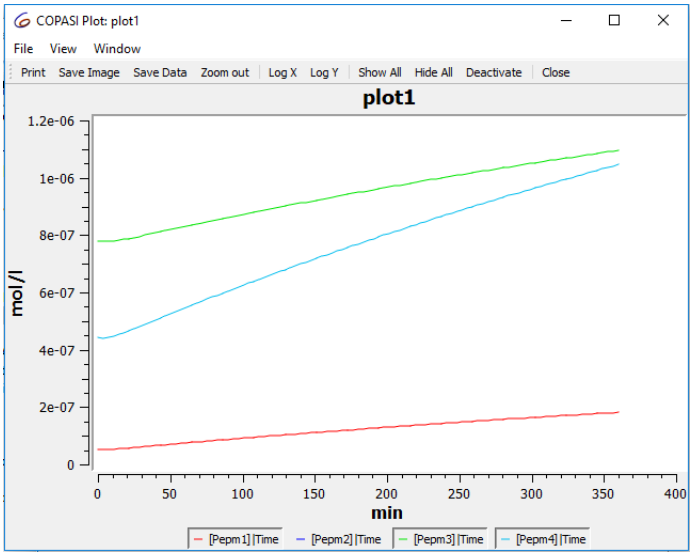
- The Four **Curve Specifications** will be displayed as below:
- Users can modify the **Type** (Lines, Points, Symbols, Lines & Symbols), **Line Style** (Solid, Dotted, Dashed, etc.), and the **Color** for the particular curve.
- Click on **Commit** button at the bottom to commit the change



- Go back to the **Tasks** (left panel) → **Time Course**
- Click on the **Run** Button at the bottom to execute the model simulations
- **COPASI Plot: plot1** will be generated as follows:



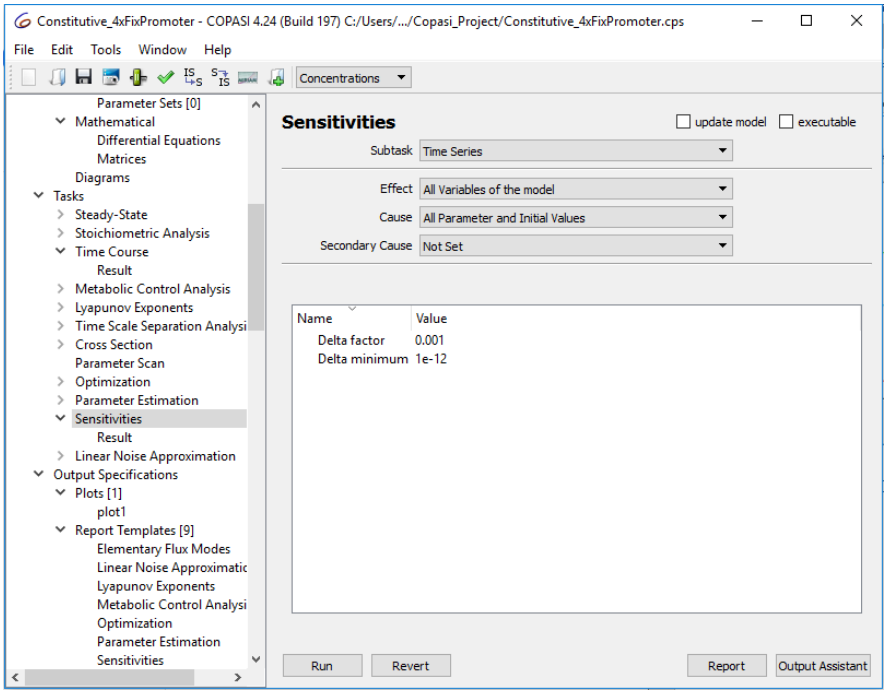
- Click on the Legend shown at the bottom of the plot to display/hide the particular curve.
- In the bottom figure, [Pepm2] Time curve is hidden after clicking on the legend.



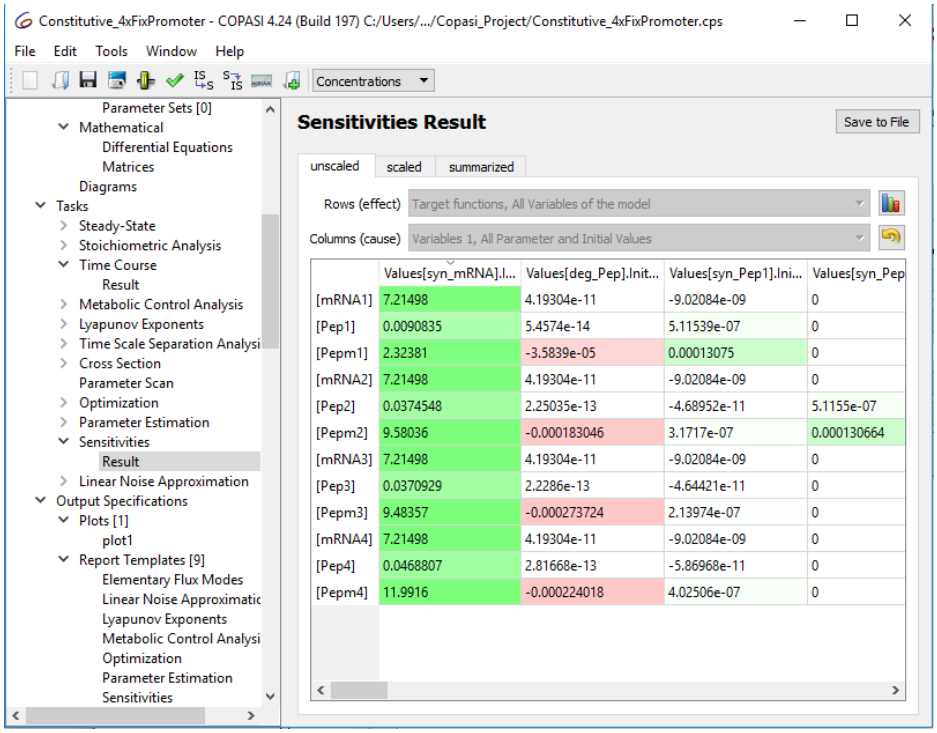
2.2.4. Model Analysis: Sensitivities

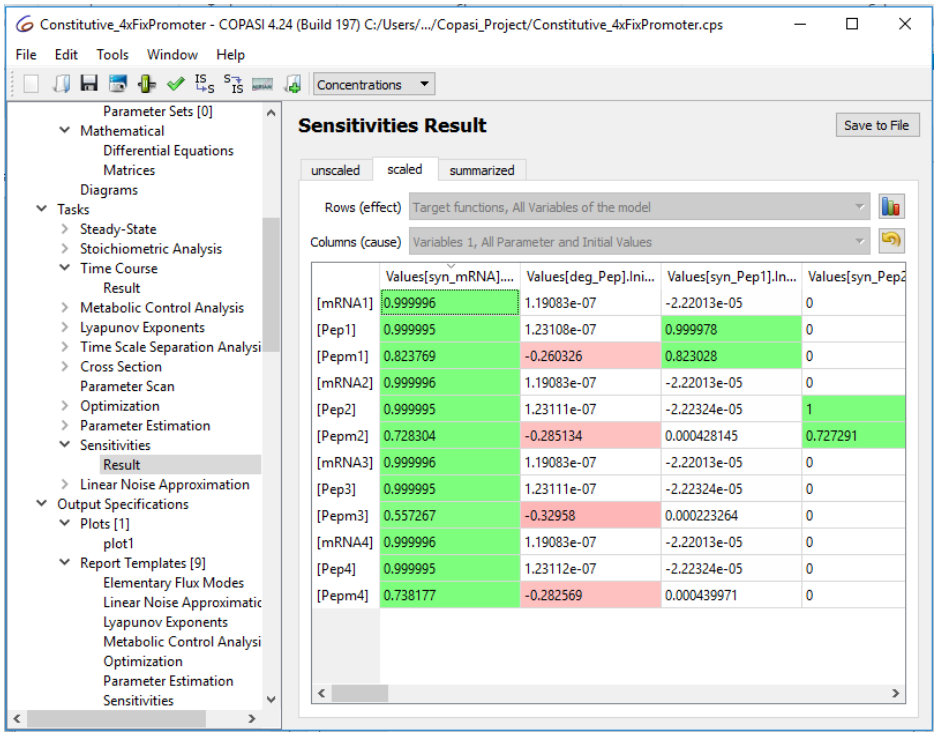
COPASI allows the calculation of sensitivities of the model (all variables or a specific output variable) in response to various parameters or even initial values set for the different variables. Generally, sensitivity analysis determines how changing an independent variable affects the change in a particular dependent variable (kinetic parameter) under a given set of assumptions.

- Click on **Tasks → Sensitivities**
- Select **Time Series** for **Subtask**
- For **Effect**, choose the option of **All Variables of the model** Or a **Single Object** and choose the particular Species by clicking the button  on the right (**Sort → Species → Transient Concentrations →**)
- For **Cause**, Choose the option of **All Parameter and Initial Values** Or **All Parameter Values**
- Click **Run** at the bottom



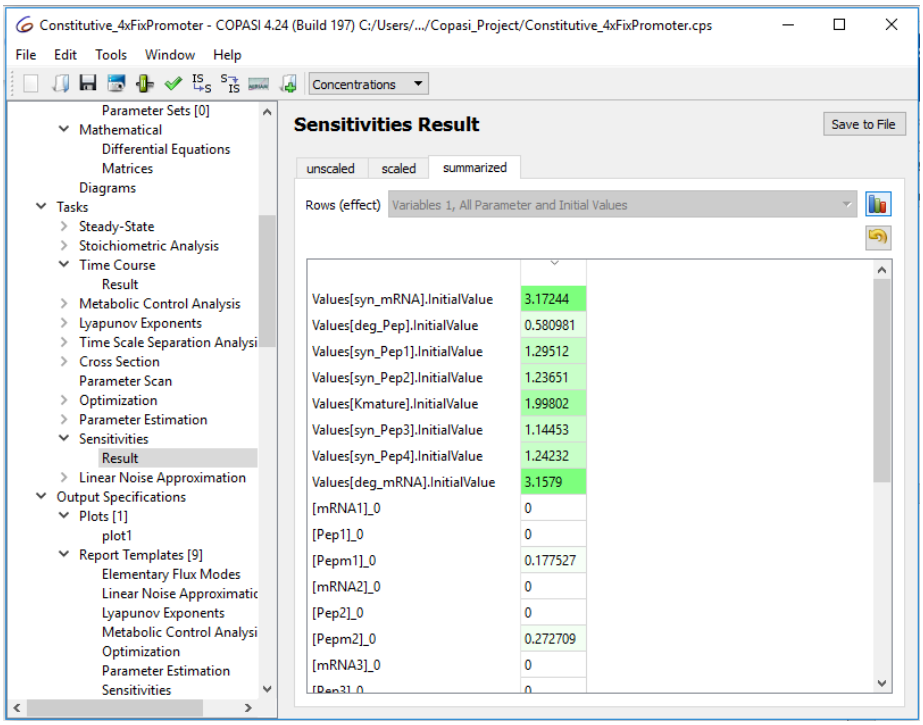
In the Result, there will be three tabs of different results. The results presented in the unscaled and scaled tabs are in the form of matrices, where the top row indicates the different causes, and the left column denotes the different effects/dependent variables. The highlighted green cells are associated with the positive sensitivity coefficient and the colour density relates to the relative magnitude of the coefficient, whereas the highlighted red cells denote the negative sensitivity coefficient. The base cells are associated with coefficients close to zeros.



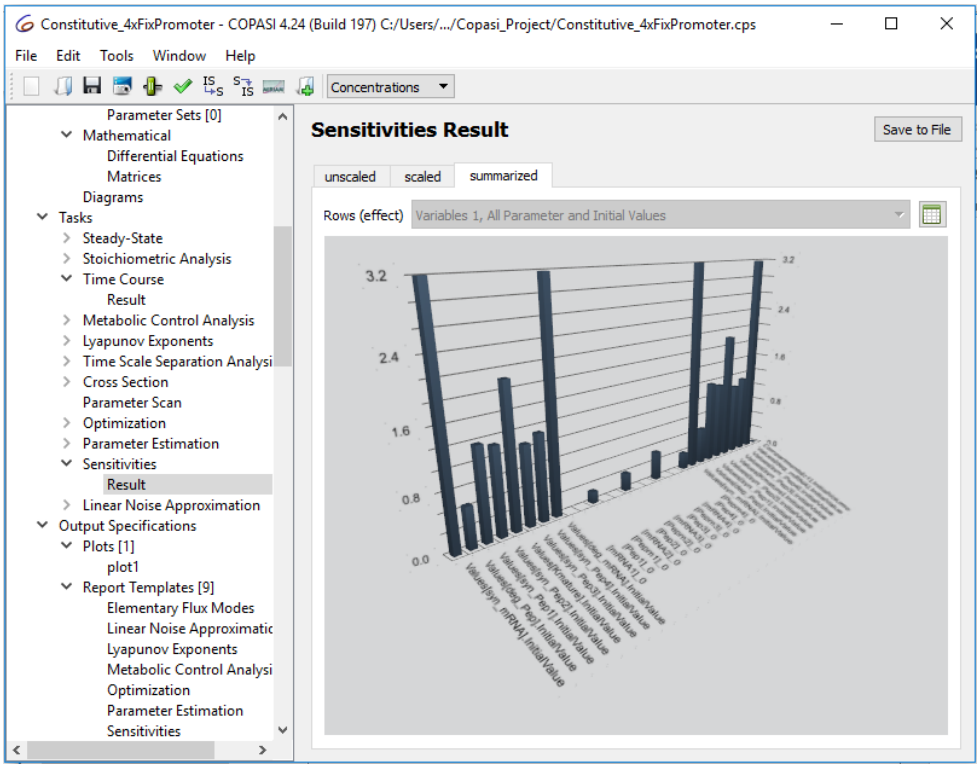


- Scaled: defined as the ratio between the relative change in the variable and the relative change in the parameter.

- Unscaled: ratio between the absolute change in the variable and the absolute change in the parameter.
- Users can navigate to the summarized results at the third tab.

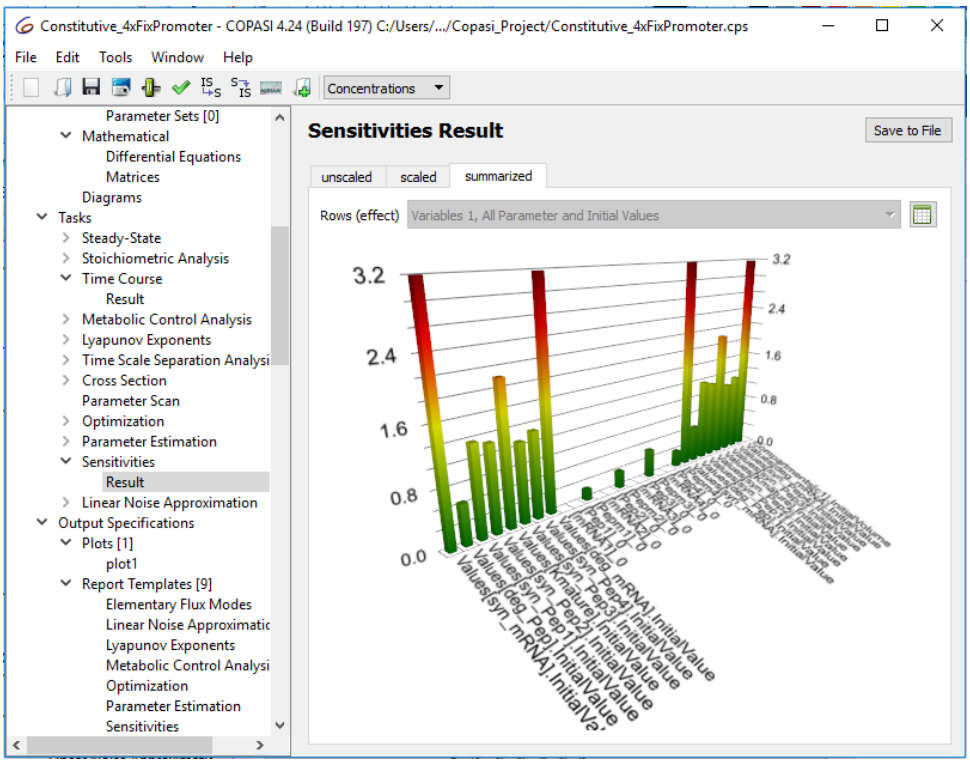


- Click on the top right  bar chart symbol to display the summarized results in the form of bar chart.



To change the chart settings:

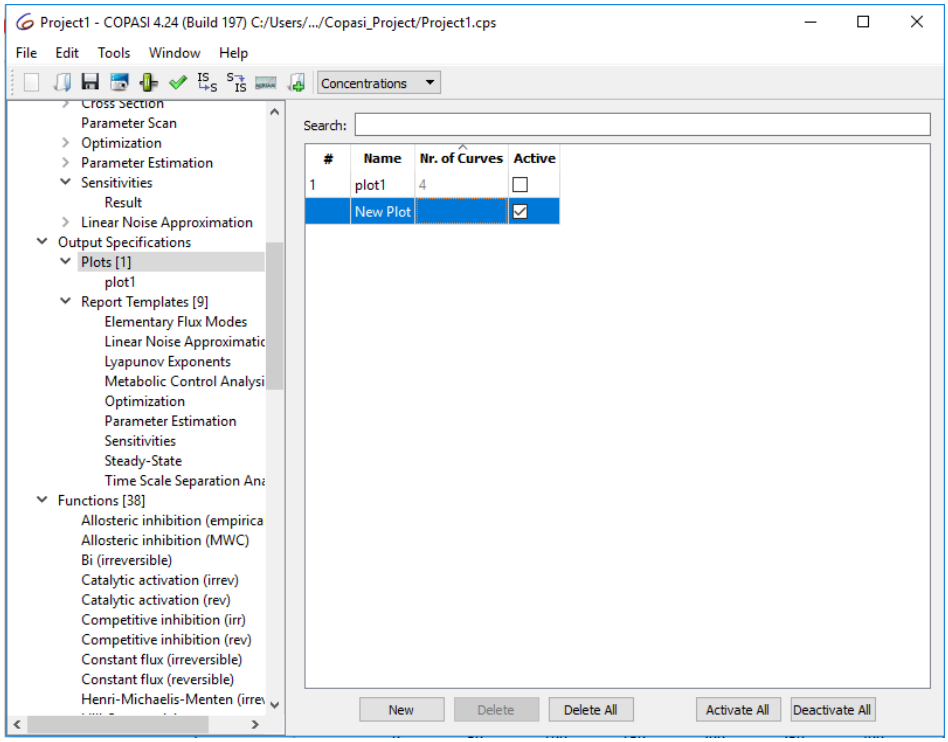
- Right-click on the chart to show the different chart properties
- Choose **Theme → Primary Colors**
- Click on **Change label style** to enlarge the font label
- Click on **Show Gradients** to use color gradient to represent the different magnitudes



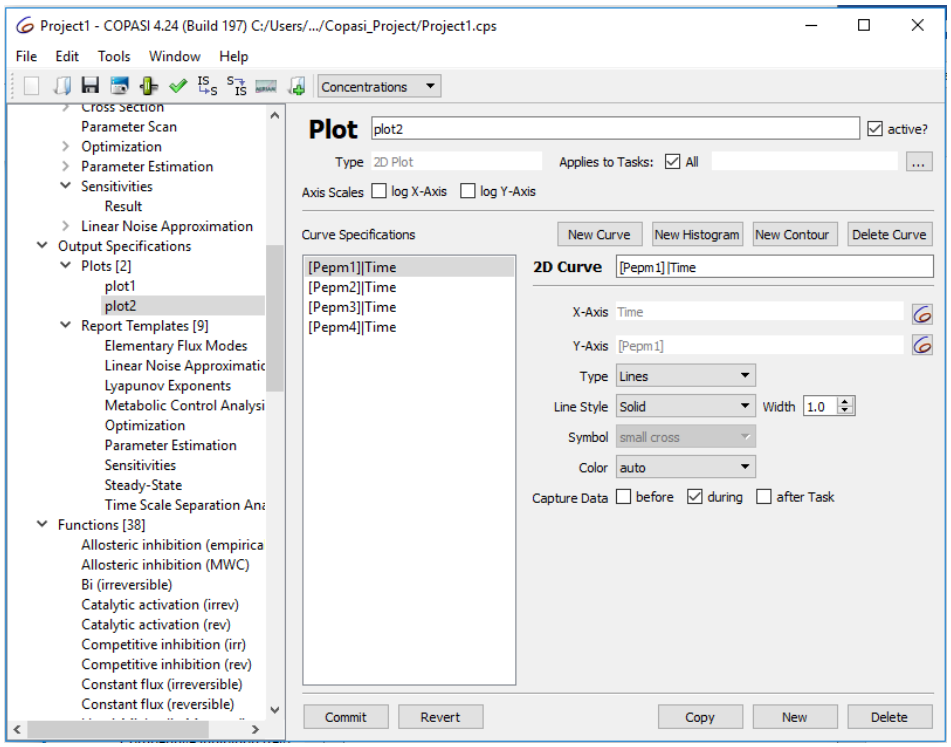
2.2.5. Modifying Model Parameters

To run simulation with modified parameter:

- Output Specifications → Plots[1]
- Untick the Tick Box under Active for plot1 that was generated in the previous run.
- Double click the New Plot

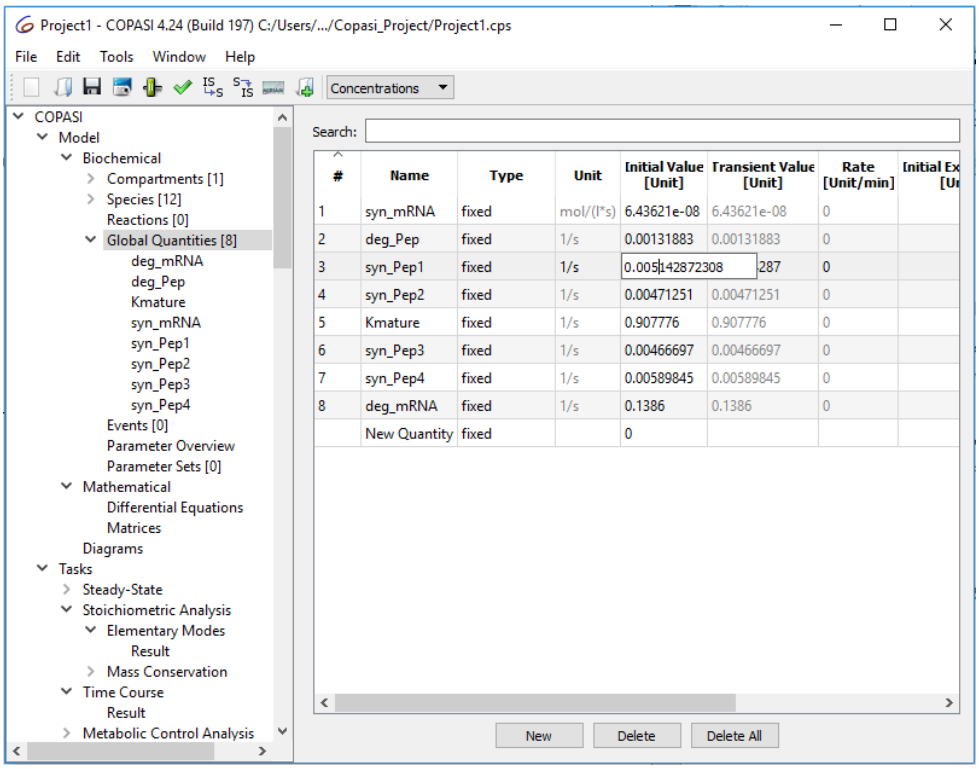


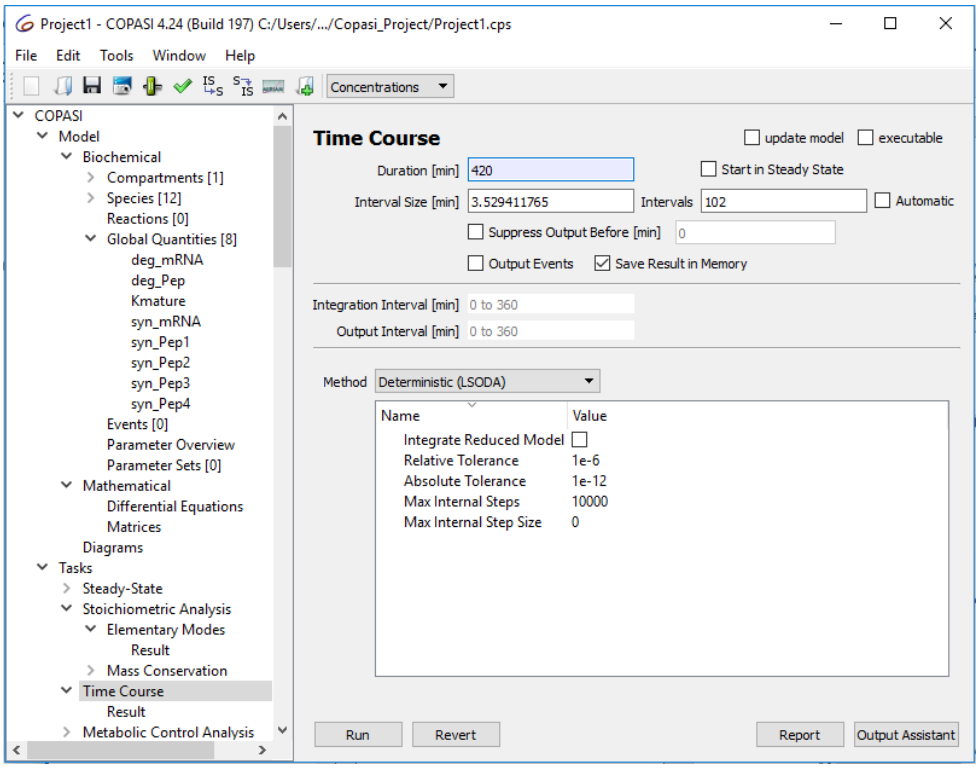
- Create a new plot by changing the **Plot** to plot2
- Click on **New Curve**
- Repeat the steps to select the **Time → Model Time** for X-Axis; and **Species → Transient Concentrations** and the four [PepmX](t) variables for Y-Axis for plotting.
- Click **Commit**



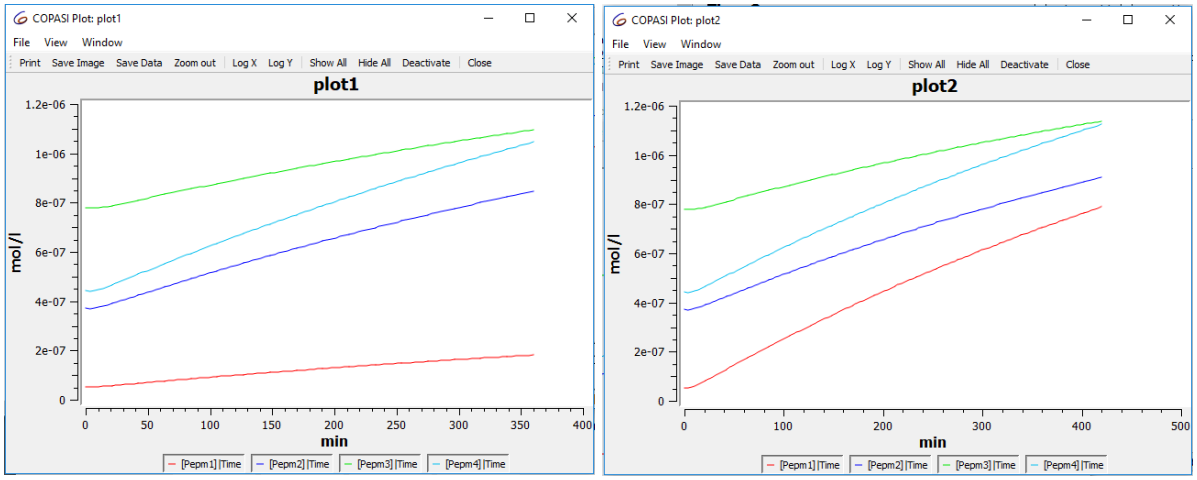
- Go to **Model → Biochemical → Global Quantities**
- Modify the **Initial Value** of parameter **syn_Pep1** (e.g) from 0.00114 to 0.00514

- Go to **Tasks → Time Course**, change the **Duration [min]** from 360 (previous run) to 420 to test the simulation for longer time
- Click **Run** at the bottom to execute the simulation





- The results after changing the parameter value of **syn_Pep1** and running for 420 min are displayed in **plot2** at the right, in comparison with the original **syn_Pep1** value run for 360 min shown in **plot1** at the left. There is an increase in the expression profile for [Pepm1] Time curve after increasing the Protein synthesis rate (from 0.00114 to 0.00514).



Conclusion

The developed automated system enables users to automatically fit many different models with distinctive underlying assumptions to their experimentally measured dynamic characterization data with minimal inputs and recommends the best model candidate from a list of model rankings. This system also forms the basis to capture knowledge using models. The model selection system could eventually serve as a pre-screening hypothesis testing tool to be coupled with human interpretations and other existing computer-aided analysis platforms (e.g. iBioSim, COPASI, etc.) to expedite the model development and experimental design optimization.