Tutorial

Justin Womack, Graduate Student
Medical College of Wisconsin
Biomedical Engineering
Jwomack@mcw.edu

Note to Readers

This is a tutorial based on paper of "Mathematical Modeling as a Tool for Investigating Cell Cycle Control Networks" by Jill C. Sible and John J. Tyson (2007).

Best of luck,

Justin Womack

Contents

1 Tutorial

This is a tutorial covering how to use the modeling program to develop a published model "Mathematical modeling as a tool for investigating cell cycle control networks" by Jill C. Sible and John J. Tyson. This paper constructs a model used to demonstrate how the cell cycle protein network functions during the mitosis phase. The typical flow of the model construction is as follows:

1. Enter Variables to find differential equations for
2. Build equations of model
3. Enter the input and outputs of the system
4. Enter numerical values for parameters and initial conditions
5. Set differential equation time and settings
6. Solve, plot, and export model

1.1 Background

Below is an image of the model pathway:

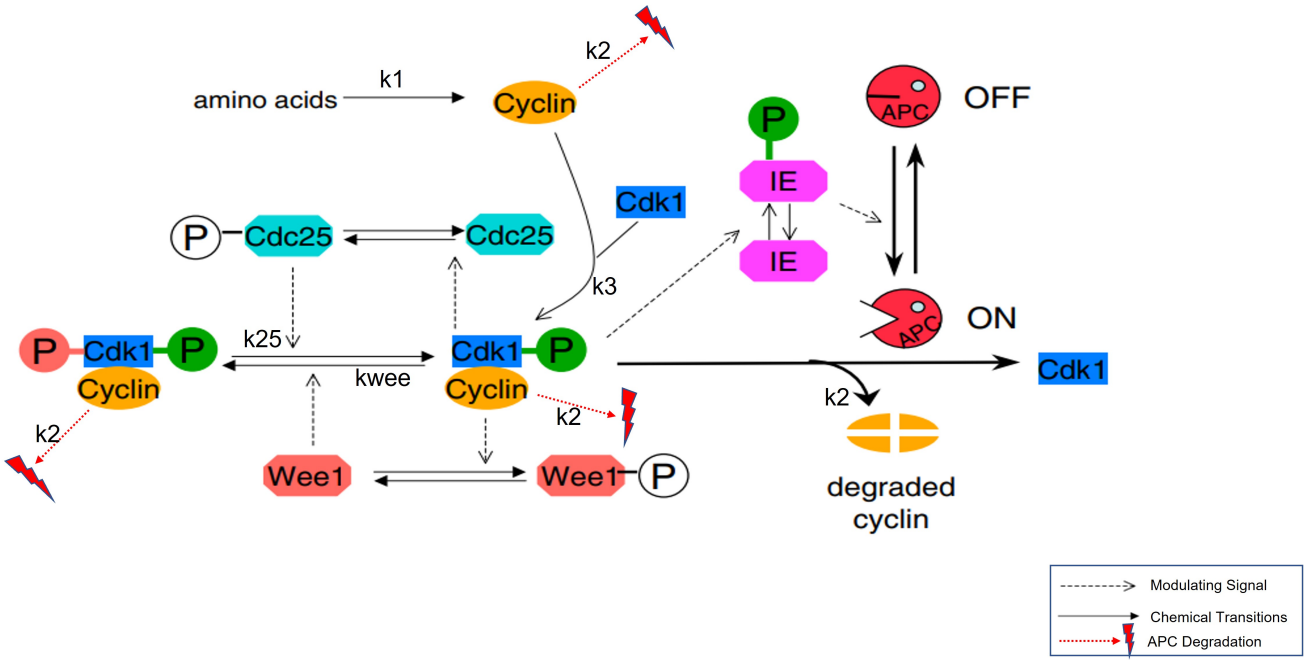

Figure 1: Model Schematic

1.2 Inputting Variables

To use the online version of this application, visit `https://jwomack7512.shinyapps.io/Model_Builder/`. Note this app is stored on an Rshiny server that close unused applications after five minutes of inactivity.

We start by adding all the variables that are used in the model that we want differential equations for (i.e parameters are not entered here). Variables simply need to be typed in the box and submitted by pressing enter or using the submit button. All entered variables show up in the text box below. This model has the following variables:

1. Cyclin
2. Cdk
3. preMPF
4. MPF
5. Cdc25
6. Cdc25P
7. Wee1
8. Wee1P
9. PPase
10. IE
11. IEP
12. APC
13. APCP

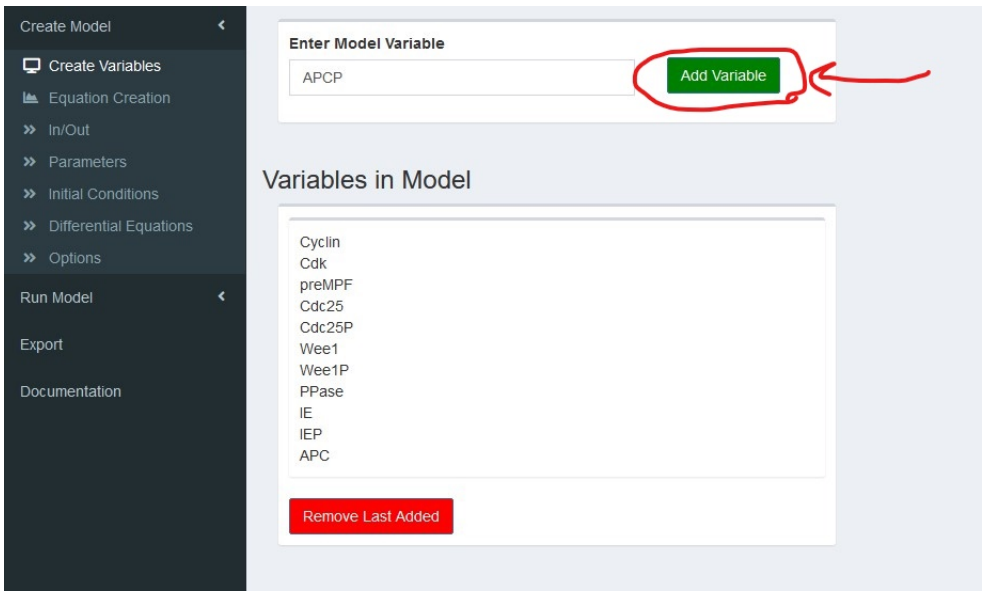

Figure 2: Depiction of create variables section showing all how to enter variables.

1.3 Create Equations

In this section, we will build the model by detailing the sets of reactions that make up the network. The first equation in this network is:

$$Cyclin + Cdk \xrightarrow{k_3} MPF \tag{1}$$

Here we have a chemical reactions with two inputs, one output, moving in the forward direction. The reaction uses the rate constant k_3. The input for this can be seen in the below image:

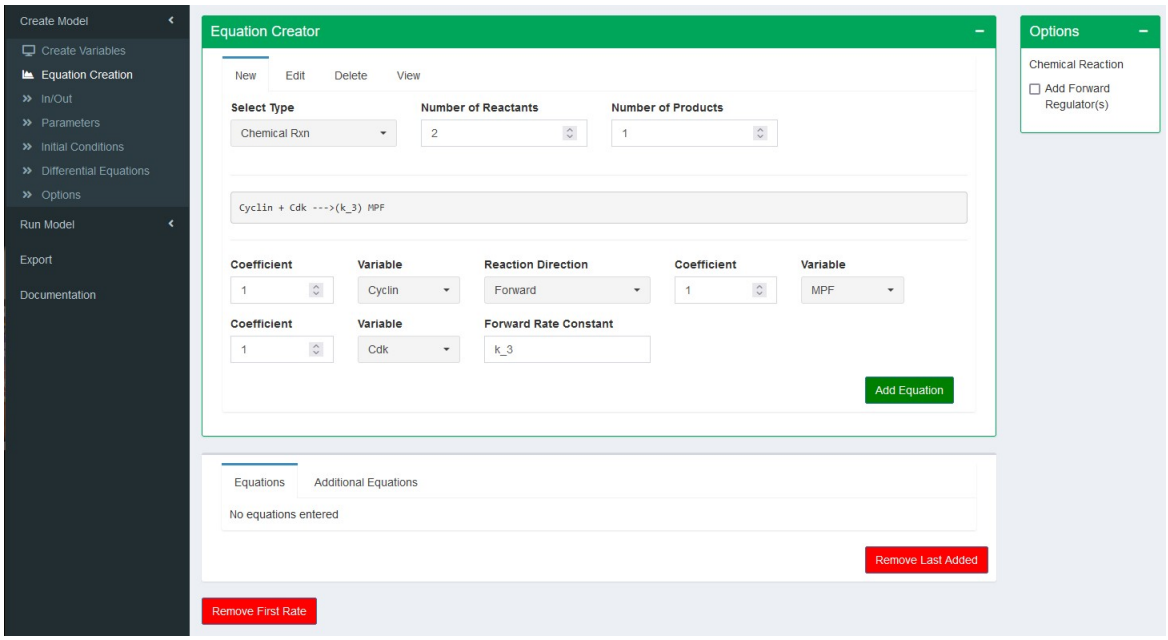

Figure 3: Data to be entered for first reaction: Create of MPF from Cyclin and Cdk.

Next, we enter the second equation, the conversion of preMPF to MPF which is a reversible reaction. This is much like the previous step except we change the value of "Reaction Direction" to "Both" to account for the reversible reaction. Then, both the forward and reverse rate constants are typed in.

$$preMPF \underset{k_{wee}}{\overset{k_{25}}{\rightleftharpoons}} MPF \tag{2}$$

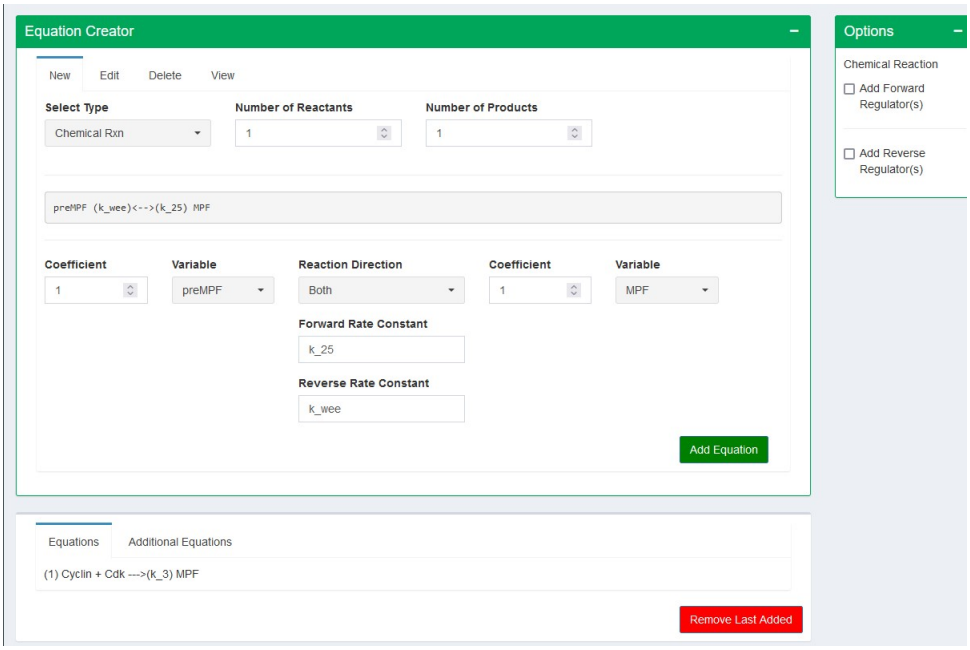

Figure 4: Conversion of preMPF to MPF in a reversible reaction.

The next reaction to be added is the conversion of Cdc25 to Cdc25P. This is an enzyme catalyzed reaction that uses the Michaelis-Menten equation in solving for the differential equation. Note that all enzyme reactions in this model use k_{cat} and the enzyme concentration over V_{max}. After choosing "Enzyme-Catalyzed Rxn" for the type of reaction, select the checkbox in the options sidebar to "Split Vmax". The rest of the equation information can be entered from the below equation:

$$Cdc25 + MPF \underset{k_a(EnzRxn)}{\overset{K_a}{\rightleftharpoons}} Cdc25P \tag{3}$$

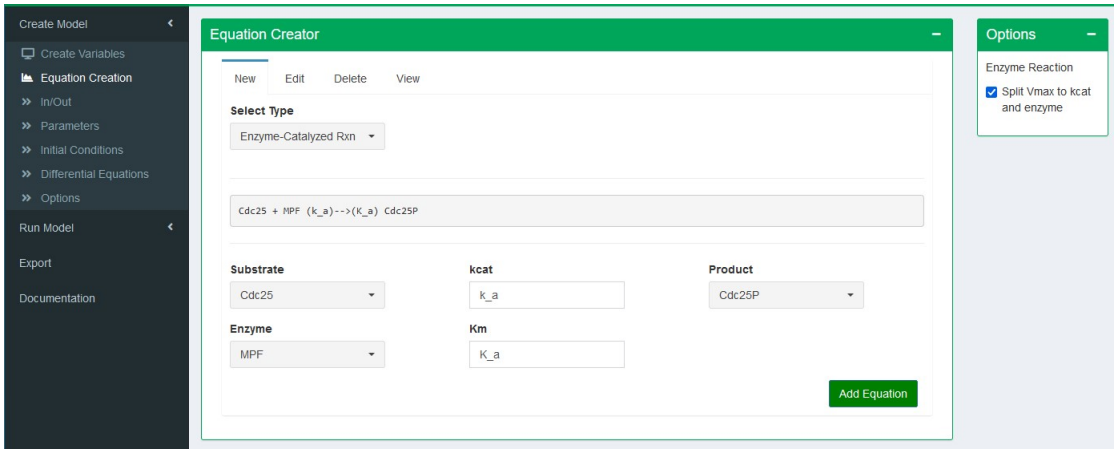

Figure 5: Enzyme conversion of Cdc25 to Cdc25P by enzyme MPF.

Those are all the types of equations that need to be added for this model. The following

are a list of all equations in the model:

$$Cdc25P + PPase \underset{k_b}{\overset{K_b}{\rightleftharpoons}} Cdc25 \tag{4}$$

$$Wee1P + PPase \underset{k_f}{\overset{K_f}{\rightleftharpoons}} Wee1 \tag{5}$$

$$Wee1 + MPF \underset{k_e}{\overset{K_e}{\rightleftharpoons}} Wee1P \tag{6}$$

$$IEP + PPase \underset{k_h}{\overset{K_h}{\rightleftharpoons}} IE \tag{7}$$

$$IE + MPF \underset{k_g}{\overset{K_g}{\rightleftharpoons}} IEP \tag{8}$$

$$APC + PPase \underset{k_d}{\overset{K_d}{\rightleftharpoons}} APCP \tag{9}$$

$$APCP + IEP \underset{k_c}{\overset{K_c}{\rightleftharpoons}} APC \tag{10}$$

$$MPF \xrightarrow{k_2} Cdk \tag{11}$$

$$preMPF \xrightarrow{k_2} Cdk \tag{12}$$

This is what the end result should look like:

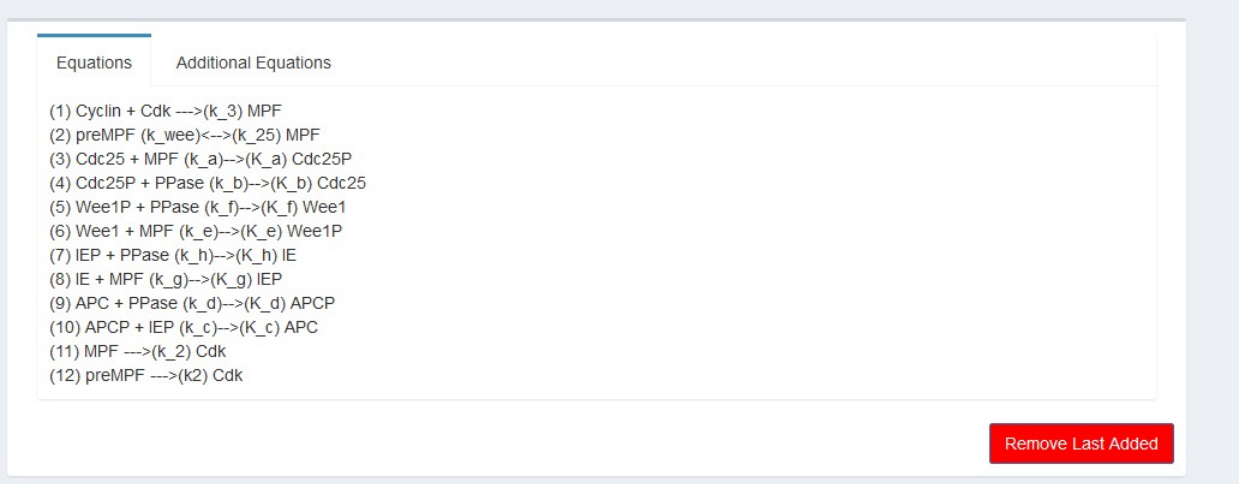

Figure 6: All equations in model shown in application

1.4 Adding Custom Equations for Rates

This model has parameters that are not single values but are instead dependent on concentrations of species in the model. In this case, an equation will need to be built for them. The three rates this model uses equations for are k_{25}, k_{wee}, and k_2 with the corresponding equations:

$$k_{25} = V_{25p} * Cdc25 + V_{25pp} * Cdc25P \tag{13}$$
$$k_{wee} = V_{weep} * Wee1P + V_{weepp} * Wee1 \tag{14}$$
$$k_2 = V_{2p} * APCP + V_{2pp} * APC \tag{15}$$

Begin by selecting "Rate Equation" in the equation creation box like before. This generates three columns of data. The first is a text printout of the equation, the second is a textInput box where additional parameters for the equation should be added separated by commas, and the last column is where you will select the rate constant and type in its corresponding equation (make sure everything is typed correctly). Below is an example of how to enter the first rate equation, k_{25}:

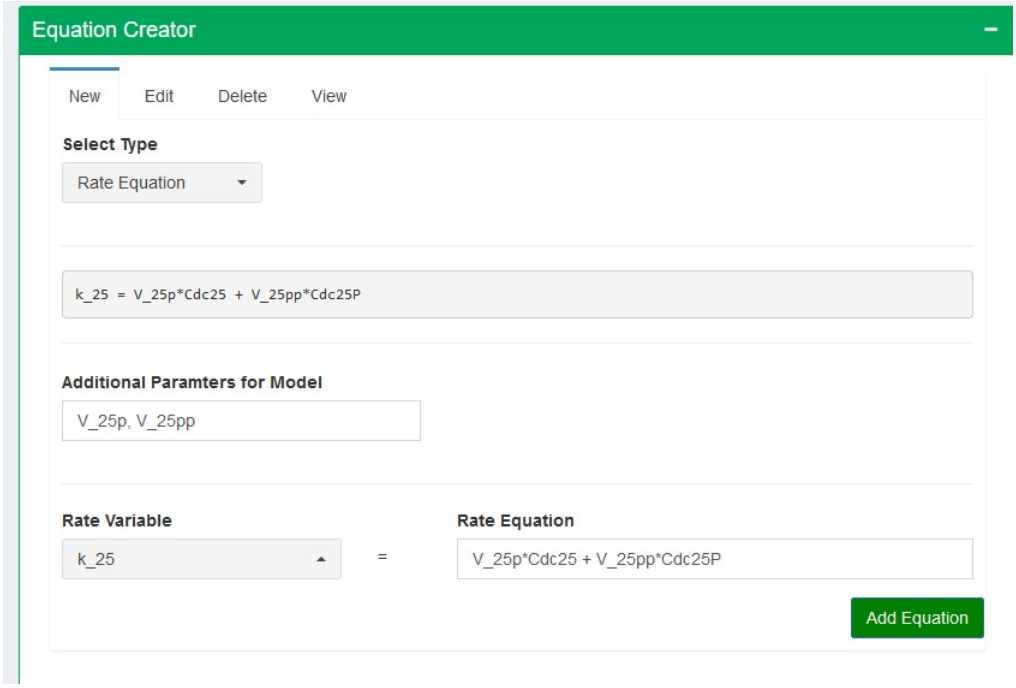

Figure 7: UI input for rate parameter k_{25}

After the "Add Equation" button is added, the textInputs will be cleared and you can find your equation in the box below the "Equation Creator" box under the "Additional Equations" tab. Enter the remaining two equations and your additional equations output should look like this:

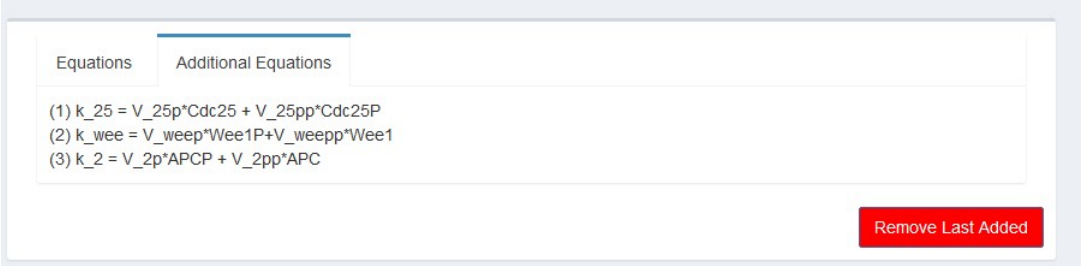

Figure 8: Caption

This concludes the equations section of this tutorial.

1.5 Input & Outputs

In this section we cover adding input and outputs (IO) of species into the system. In this model, we cover the synthesis and degradation of species through rate laws. We will cover the:

1. Synthesis of Cyclin from amino acids
2. Self Degradation of Cyclin

The synthesis of Cyclin is modeled with a simple rate parameter, k_1. For this step, be on the IO tab of the program. On the "Select Variable" box, make sure "input" is selected and make sure to select "Cyclin". In the box over, select "Rate" and type in the name of the rate constant as "k_1". Press the Add input button and proceed to the next IO. This setup is seen below:

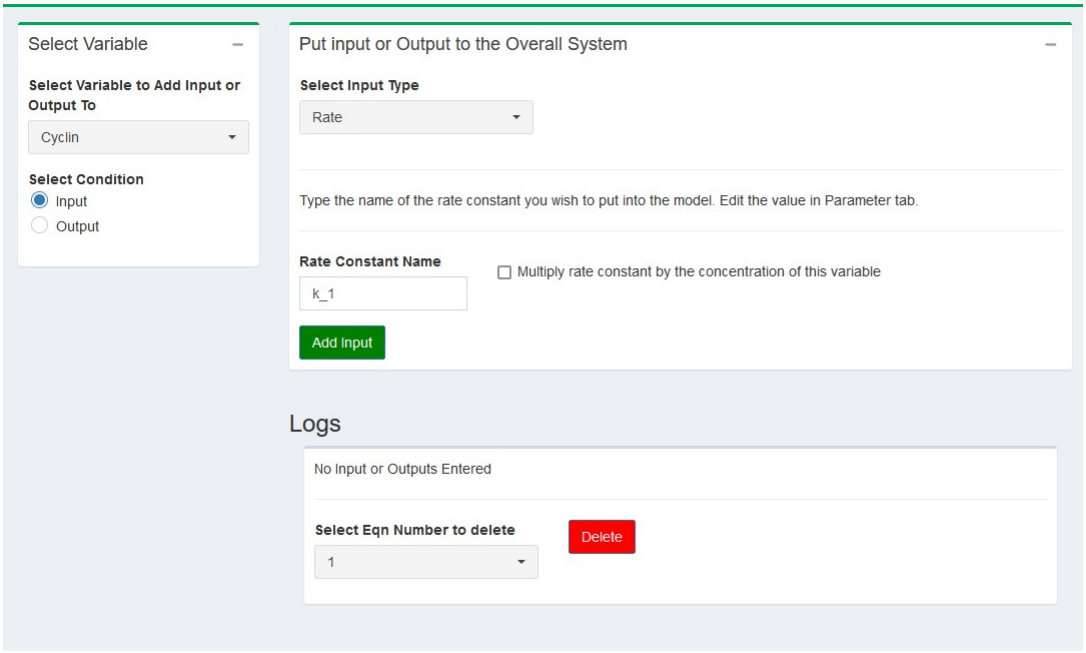

Figure 9: UI input for the synthesis of Cyclin from amino acids.

The output of Cyclin by self degradation follows much the same pattern, expect we change the radiobutton "input" to "output" and note that this self-degradation is concentration dependent. Because of this be need to check the checkbox next to where we enter the parameter name to signify this. This setup can be seen in the below figure.

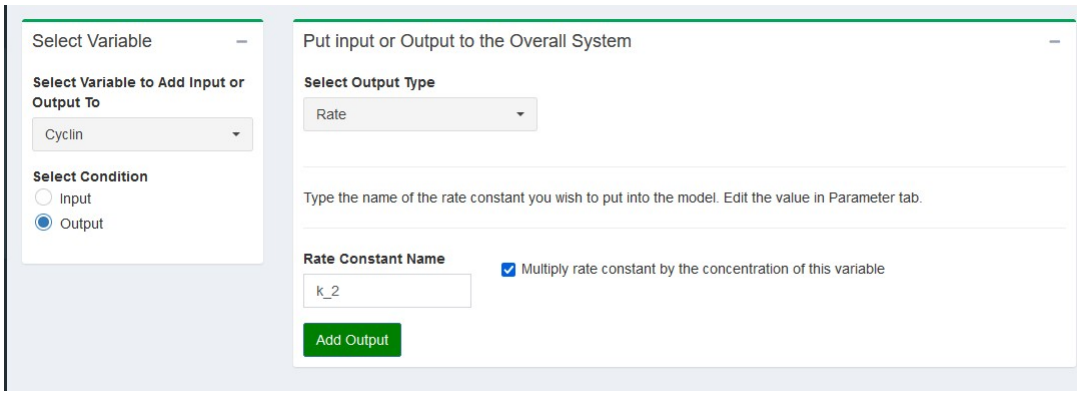

Figure 10: UI input for the degradation of Cyclin.

Lastly, below is an image of the logs which let you know what inputs and outputs have been added to the model:

Figure 11: UI input for the degradation of Cyclin.

1.6 Parameters

Click on the "Parameters Tab". In this tab, there should be a list of the parameters entered when the equations and IO were entered. There is an options to give them a value and a comment. The comment helps to keep track of what the parameter does and will be printed in tables and code output in subsequent section (but you can leave them blank). Go through the list and give the parameters their correct corresponding values. Press the "Store Parameters" button when done and a popup should appear showing that the variables are saved. **Note: you should do this before leaving the tab to enter different equations as this will reset the page to whatever the last stored values were.**

The parameter values are as follows:

$$
\begin{aligned}
k_1 &= 1 \\
V_{2p} &= 0.005 \\
V_{2pp} &= 0.25 \\
k_3 &= 0.005 \\
V_{25p} &= 0.017 \\
V_{25PP} &= 0.17 \\
V_{weep} &= 0.01 \\
V_{weepp} &= 1 \\
k_a &= 0.2 \\
K_a &= 0.1 \\
k_b &= 0.1 \\
K_b &= 1 \\
k_c &= 0.13 \\
K_c &= 0.01 \\
k_d &= 0.13 \\
K_d &= 1 \\
k_e &= 0.02 \\
K_e &= 0.1 \\
k_f &= 0.1 \\
K_f &= 1 \\
k_g &= 0.02 \\
K_g &= 0.01 \\
k_h &= 0.15 \\
K_h &= 0.01
\end{aligned}
$$

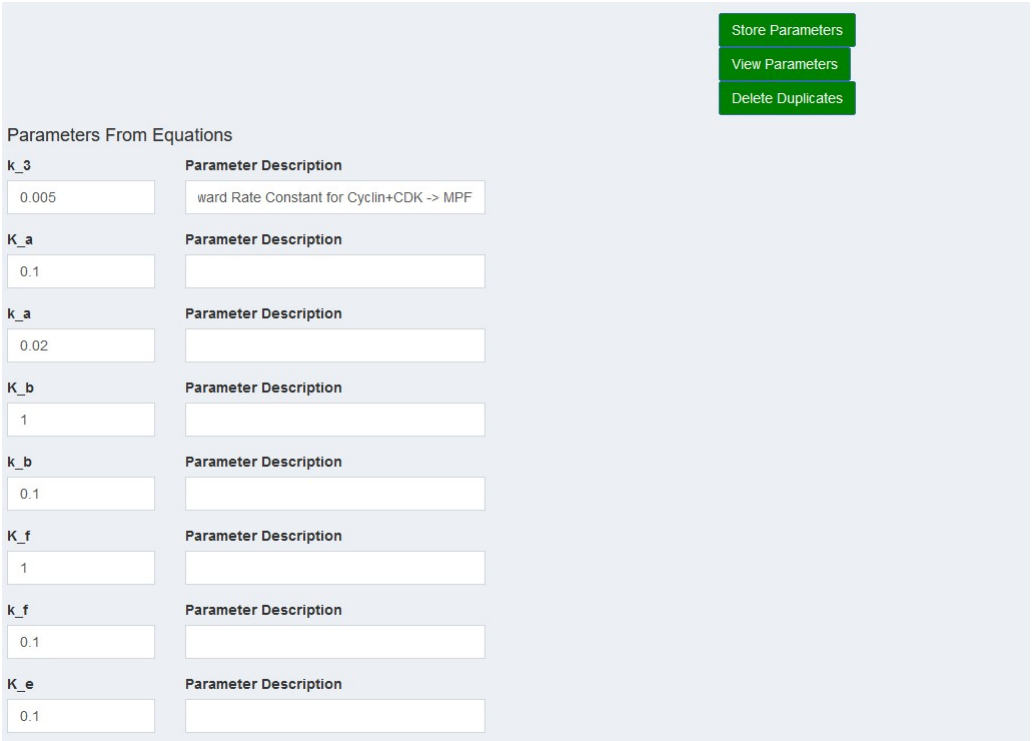

Figure 12: Example of what the enter parameter screen should look like.

1.7 Initial Conditions

Swith to the "Initial Conditions" tab. Notice that this tab is much like the parameter tabs and functions in the same way. Enter the values below and press the store button.

$$Cyclin = 0$$
$$Cdk = 100$$
$$MPF = 0$$
$$preMPF = 0$$
$$Cdc25P = 0$$
$$Cdc25 = 1$$
$$Wee1P = 0$$
$$Wee1 = 1$$
$$IE = 0$$
$$IEP = 1$$
$$PPase = 1$$
$$APC = 0$$
$$APCP = 1$$

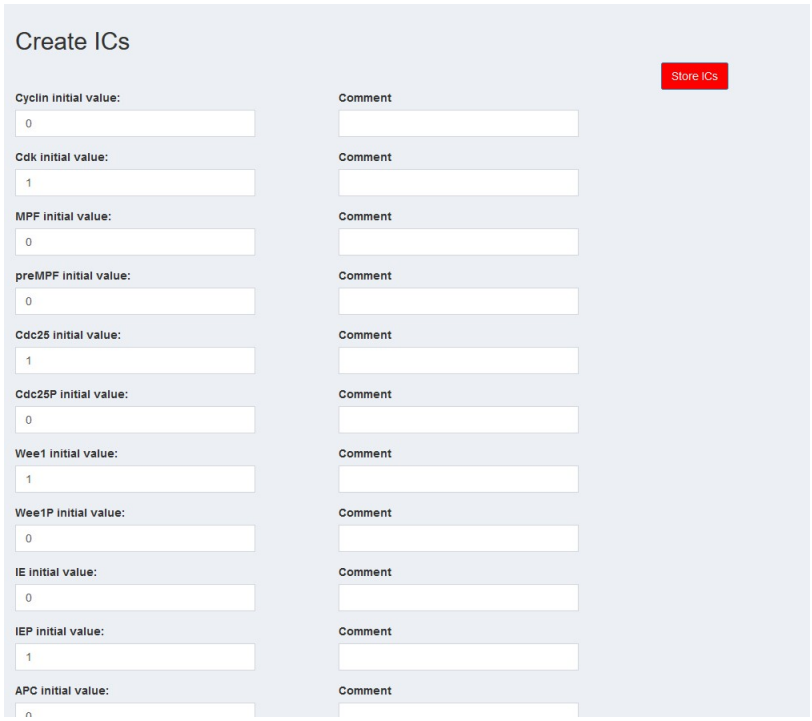

Figure 13: Example of entering the initial conditions into the model building program.

1.8 Differential Equations

In the "DiffEQs" tab, is a simple textoutput and a button. When the button is pressed, all the models differential equation will be generated from your equations and IOs. Here is what your current output should look like:

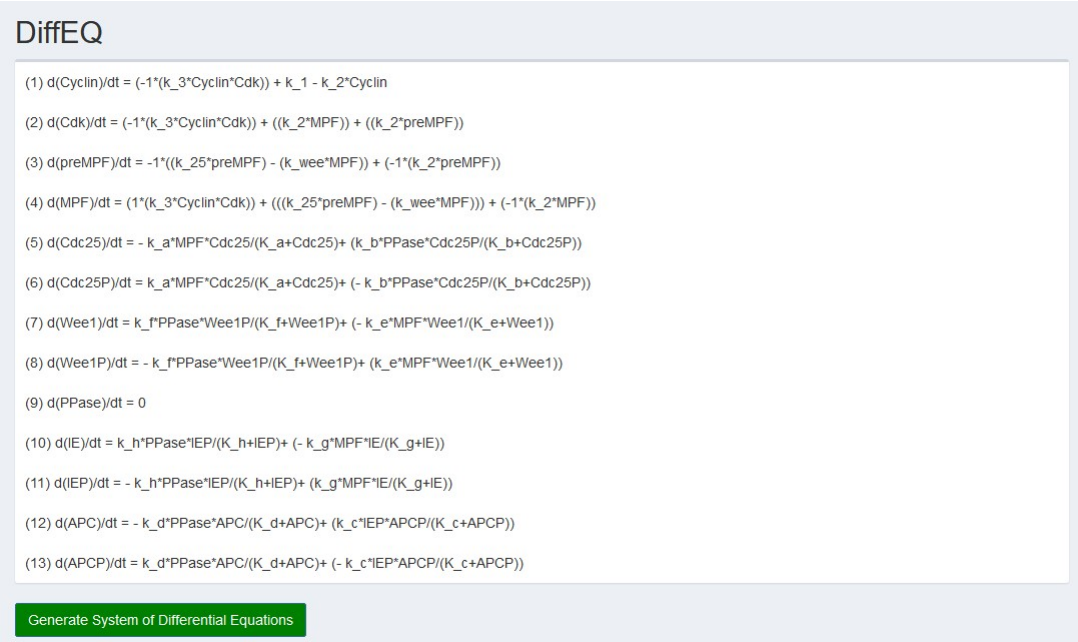

Figure 14: Output of the 13 differential equations in this model.

1.9 Time Selection & Options

The options tab allows to user to set the time the model will run at, scale the output, and select the type of ODE solver. For this model, we do not need to scale and we will keep the ODE solver on the default "lsoda". Set the time to be between 0 and 200 with a time step of 1. Finish by pressing the "Store Options" button to store these options to be used in the model.

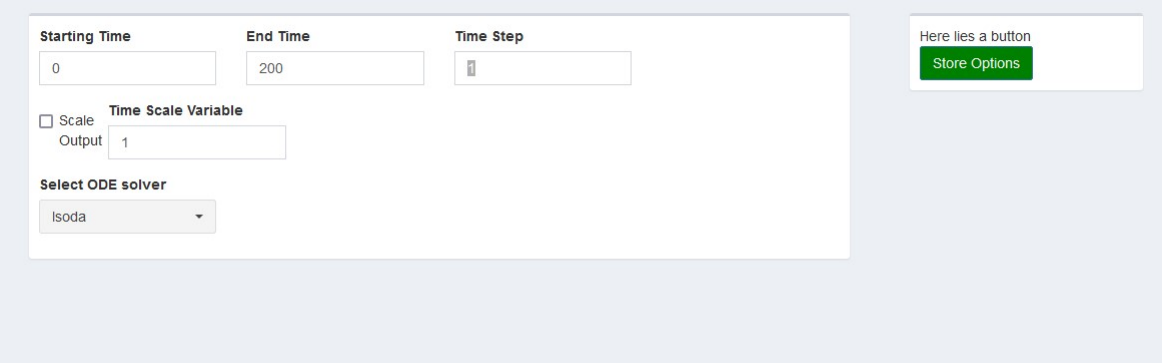

Figure 15: UI of the options for model.

1.10 Export Data

The last main tab of the application is "Export". In this tab, we have the option to three boxes that correspond to three separate functions:

1. Exporting the model to functioning MatLab or R code
2. Save the model data in a file that can be loaded later
3. Export latex code for documentation of equation information

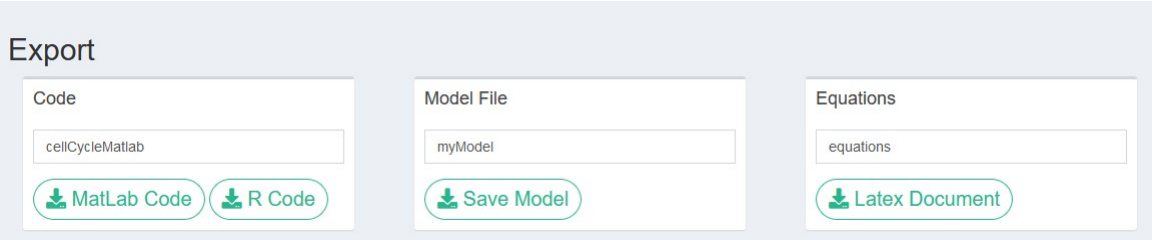

Figure 16: Export UI showcasing the three different types of export with titles placed in each box.

When exporting any of these types of data make sure to give the file a name with no extension (i.e no ".m" or ".R"). Then press the corresponding "Code" button and it will open a matlab or R file that is ready to run with all your model information.

The second box will download an ".rds" file which is a storage file for R data. This is a fancy container that stores all the model information. In the top right corner of the model there is a gear icon that will open the open to load a data file in. This takes in the ".rds" file that will be loaded. Note: the .rds file can be manually edited in R if the user is familiar.

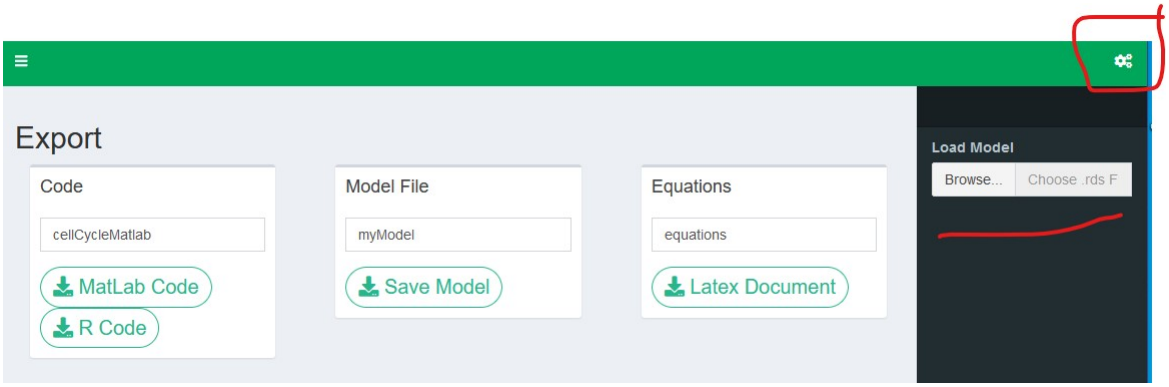

Figure 17: After saving a model, the user can open the side task bar and load the model in for future sessions.

The last box outputs relevant model information in latex formatted file. This file currently outputs a text file that can be copied over to your preferred latex editor (here we will show

overleaf.com). The relevant information that is outputted are the equations that make up the model and input/output information. Future iterations will include parameter and initial condition tables, as well as a list of the differential equations that make up the model.

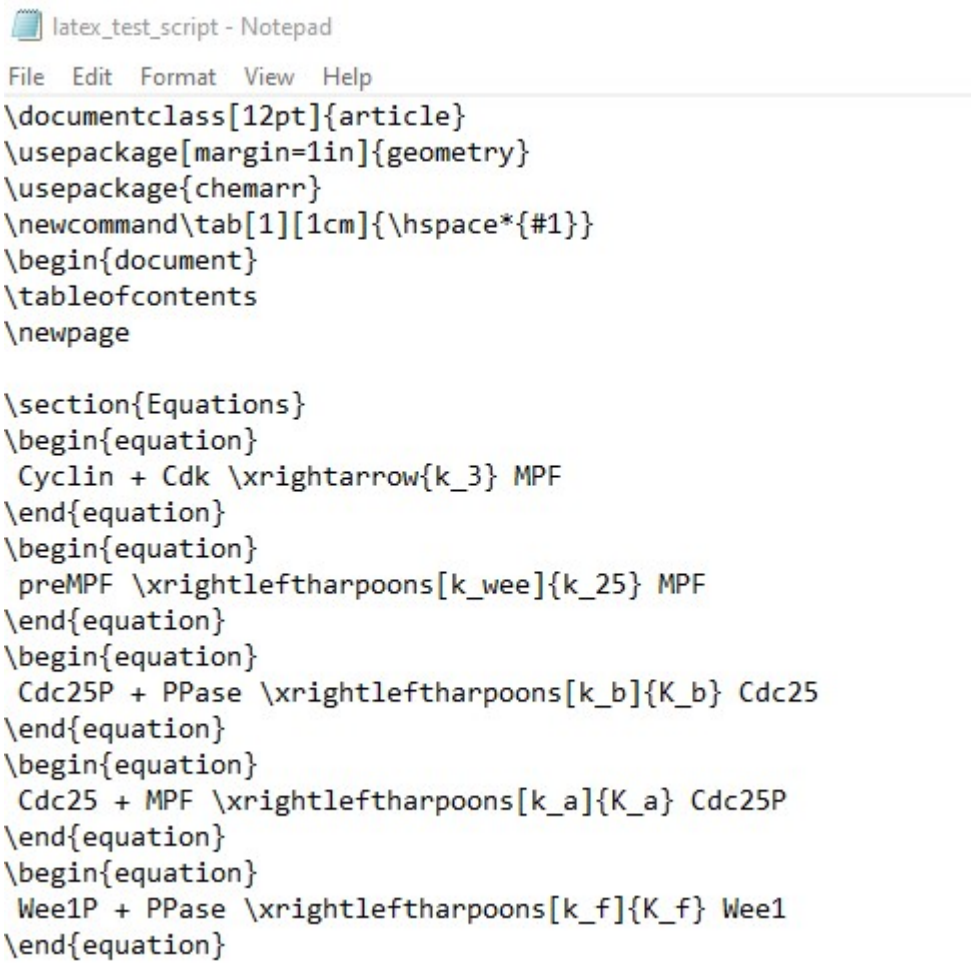

Figure 18: Text file output from pressing the Latex Document button of the application.

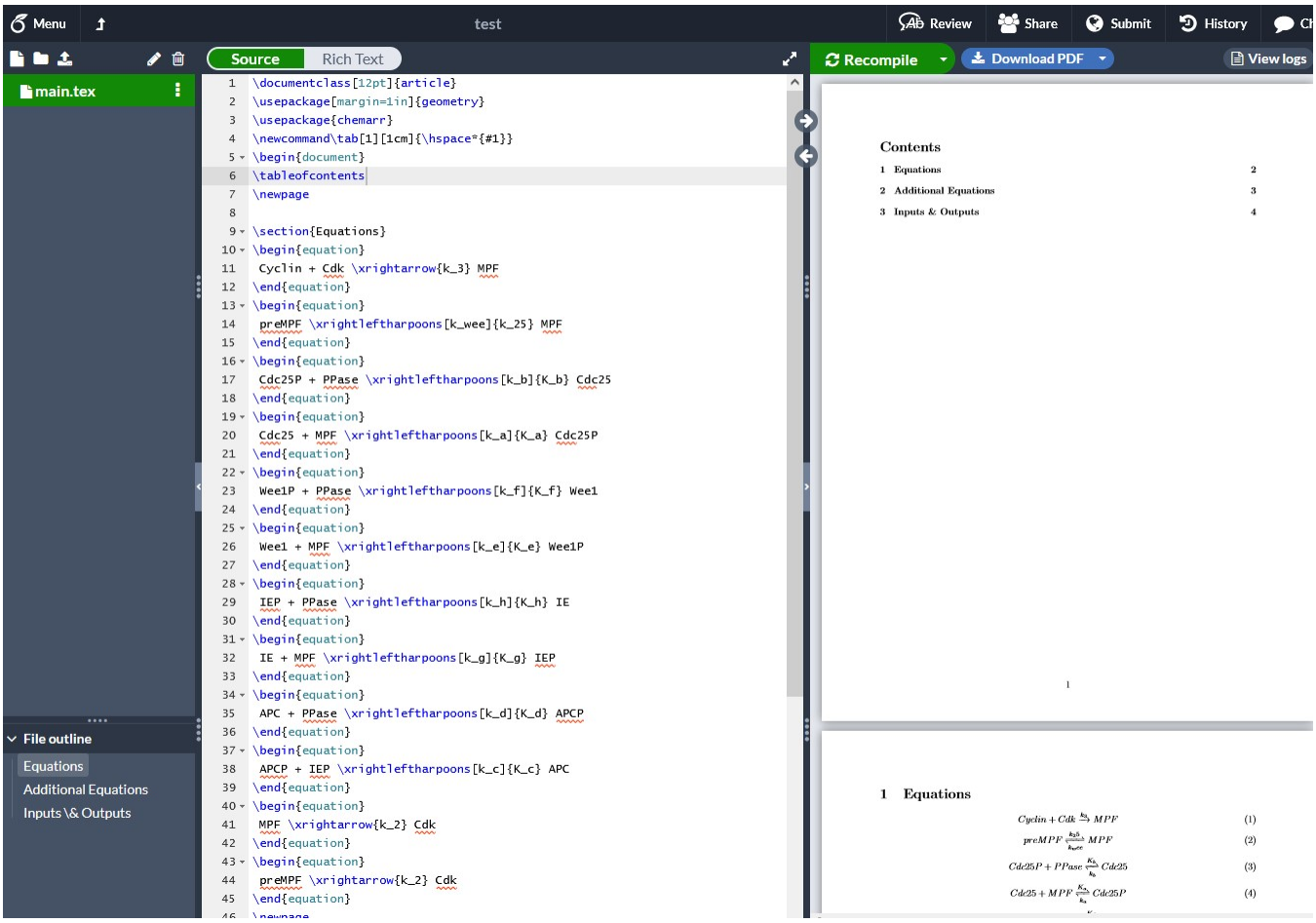

Figure 19: Example of pasting data into a blank overleaf file and compiling to make a pdf of model information.

2 Running Model

Click on the second main tab of the program "Run Model" and then the subtab "Execute Model". Here the model information will be displayed for your convenience and there is a button to run the solver. Once this button is pressed a table will appear with the results of the model.

2.1 Post Processing Model

This model has one post processing step. The model wants to know the total amount of cyclin being used which follows the following equation:

$$totCyclin = Cyclin + MPF + preMPF$$

This is done by typing in the new variable name and then selecting the variables to add together. After pressing submit, a data table should appear at the bottom of the page, showing the data with its new column.

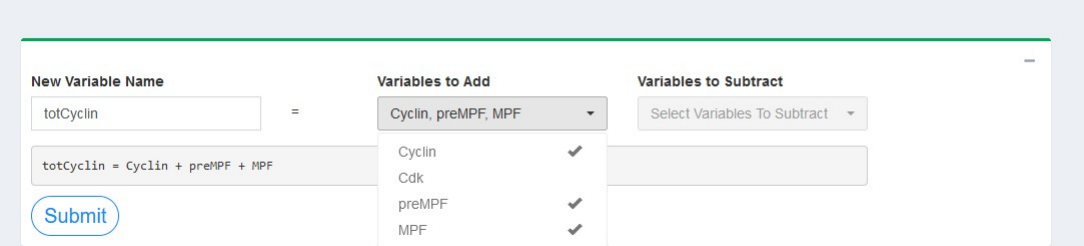

Figure 20: UI image of post process tab.

2.2 Plotting Model

The plotting tab has many modes. In this tutorial, we will keep the standard, default option which is just simple plotting. Click the "Input" drop down button. The "x variable" is automatically set to time. Select the y variables to plot:

1. preMPF
2. MPF
3. totCyclin

This should generate a plot seen below (compared with literature plot):

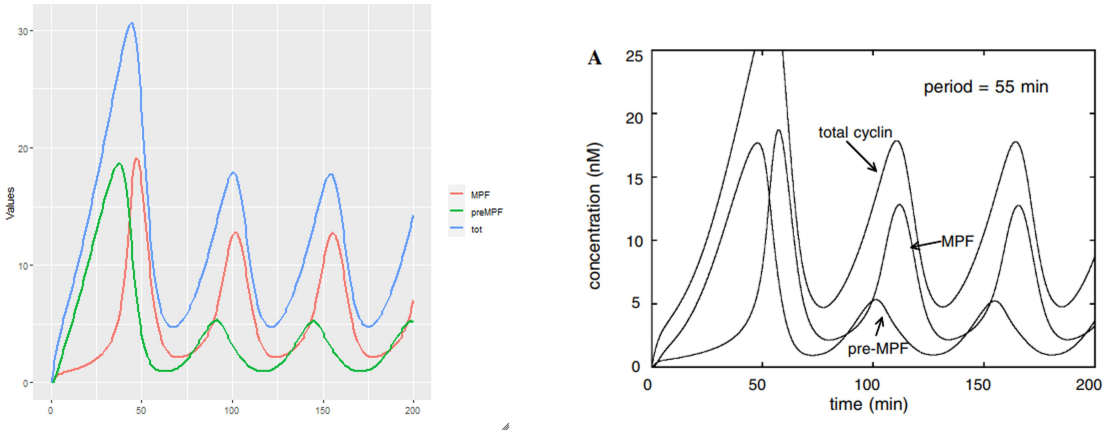

Figure 21: Left: Figure of plot generated from application. Right: Figure taken from publication.

Thats it! We have made it to the end result. Now there are options to play with to get your graph the way you like it. Check documentation for all options but we can add x,y labels, plot title, change the size and colors of the plot lines, the background of the plot, and much more.

3 Version Control

This section will contain all the version info of relevant R packages used to create this program. This is useful to have for our records as well as for users to have if they are directly downloading the code from github as packages under different versions can cause clashes. For example "RshinydashboardPlus" just did a whole revamp of their package. If the newest version of their package is installed instead of version 0.7, then the app will fail to load.