Dry Lab

HALF-TIME REPORT

Leandra Anali De Luca
IGEM STOCKHOLM 2019 | 12-07-19

Abstract

Computer modeling is finding increasing traction in the scientific community. This is especially prevalent in iGEM, as modeling has become a one of the gold medal criteria. The aim with this report is to reflect upon the work from this years Dry Lab team so far and determine the next steps in our work. The programming has been done on both MatLab and Python, the codes of which are available in GitHub. Almost everything has been coded by hand. In this report, we assessed that the system, in its current form isn't able to reach the demands we desire in our switch and needs some revision.

Table of contents

Introduction

To model a genetic circuit one needs to understand the concept of modelling. In this report, modeling isn't a person posing for a camera, musing whatever concept is presented to them, but rather a code posing as whatever system it's told to present to the programmer. Modeling in this sense is a mock-up of what is happening in our physical world and, if it is done well, it can reflect reality perfectly.

The following subchapters present the background the Dry Lab team have used to further understand computational modeling.

Our general project

The greater issue this year's iGEM Stockholm team is handling this year is multidrug resistant infections. Among many other consequences that will result from this problem, it's predicted that by 2050 multidrug resistant infections will kill 10 million people per year[1]. Among the most commonly reported resistant bacteria we have E.coli[2], this is the bacteria we are going to target directly.

An alternative to antibiotics is phage therapy, which is promising as phages naturally evolve with their bacterial prey, so resistance won't be as great of a problem as it is with antibiotics. Bacteriophages are specific and only attack bacterial cells, so in vivo treatments in eukaryotes is a viable. Another benefit to bacteriophages is that they replicate in vivo, so once they achieve a productive infection within its target, they will increase in numbers, continuing to replicate.[3]

While phage therapy seems all well and good, it is not viable in practice. Several studies prove that the effect of phage therapy is seen in 40% of cases, and within these cases, the effect of the therapy varies greatly.[4] The reason for this seems to be an issue of spatial distribution. The number of phages that reach their target are much smaller than those originally administered, this may be due to the instability of phages and diffusion. Bacteriophages need to be in a high concentration for a productive infection to begin[5].

Our plan is to overcome this spatial distribution problem by using the temperate phage P2. Our idea is to encapsulate this E.Coli bacteriophage inside of a bacterial carrier in lysogenic cycle (passive, non-killing phase). Once the carrier reaches the target, it triggers the lytic cycle (active, killing phase) so that the carrier explodes and releases the phages in a high concentration nearby the target. This should thereby solve the problem of spatial distribution and increase the efficacy in phage therapy.

We are going to model the transition from lysogenic cycle to the lytic cycle by modeling the mechanics of the P2 switch.

P2 Switch System

The switch between the lytic and lysogenic phase is controlled by a simple transcriptional switch. It uses two repressor proteins to hold itself in a certain phase and then the concentration of these repressor proteins change in such a way that the phage switches cycles. What the natural trigger to this is, is unknown. The transcriptional switch is illustrated in figure 1. The C protein is controlled by the Pc promoter while the Cox protein is controlled by the Pe promoter, the former triggers the lysogenic cycle while the latter triggers the lytic[5]. The C protein represses both promoters as a dimer while the Cox protein represses both promoters as a tetramer[6], this is vital information when we attempt to model the system later.

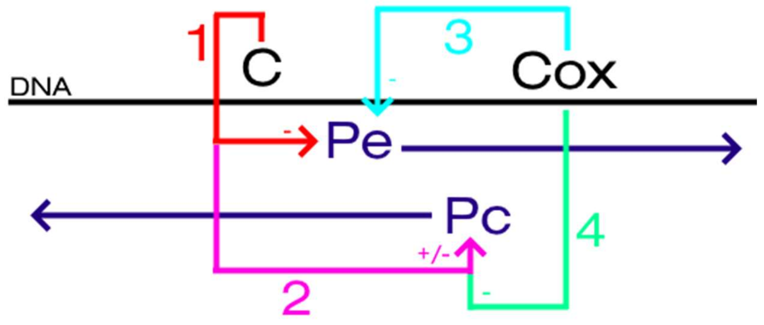


Figure 1: An illustration of the transcriptional switch. Arrow 1 signifies the C protein repressing the Pe promoter, thus repressing the expression of Cox. Arrow 2 signifies the C proteins repressive and promoting effect on the Pc promoter, the C protein autoregulates. Arrow 3 and 4 shows that the Cox protein represses both the Pe and the Pc promoter, thus suppressing itself and the expression of C protein.

Due to troubling circumstances, we cannot study the actual P2 phage, instead we will simulate the switch using plasmids. The plan is to use three different plasmids to simulate different parts of the system, the transcriptional system is illustrated in figure 2. The first part of the idea is to use the Pe promoter which will promote the expression of Cox, which is accurate to the real P2 system. The second part of the idea is to replace the Pc promoter with the constitutive promoter pTet promoter. Under a different promoter, we will express the pTet's repressor, TetR, along with Cox and thus simulate the inhibition Cox has on the expression of C.

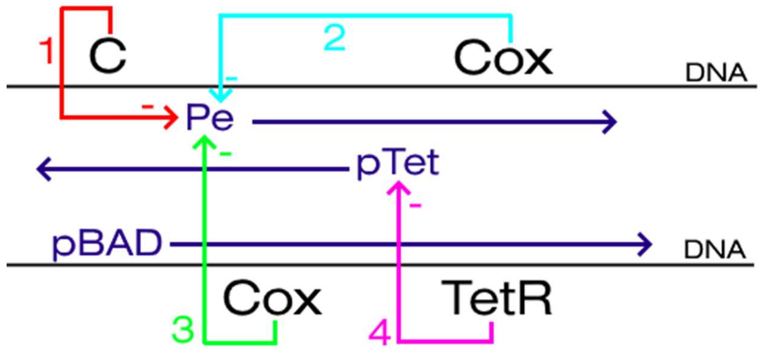


Figure 2: An illustration of the transcriptional system of the plasmids. Arrow 1 shows that the C protein suppresses the Pe promoter, similar to the real P2 switch. Arrow 2 and 3 shows how the expression of Cox inhibits the Pe promoter, similar to the real P2 switch. Arrow 4 shows how the TetR protein suppresses the pTet promoter from expressing the C protein, this simulates the effect the Cox protein would otherwise have had on the Pc promoter in the P2 switch.

The three different plasmids (see figure 3) will yield information on each part of the system. Plasmid A contains the Pe and the pTet promoters and therefore express only the C and

Cox proteins. This plasmid will yield insight in how the Pe promoter is repressed by the two proteins. Plasmid B contains the pTet and the pBAD promoters. This will simulate how the C protein is repressed and characterize the strength of the pTet promoter. Plasmid AB is a combination of all the previous, this, in theory, should simulate all of the P2 system.

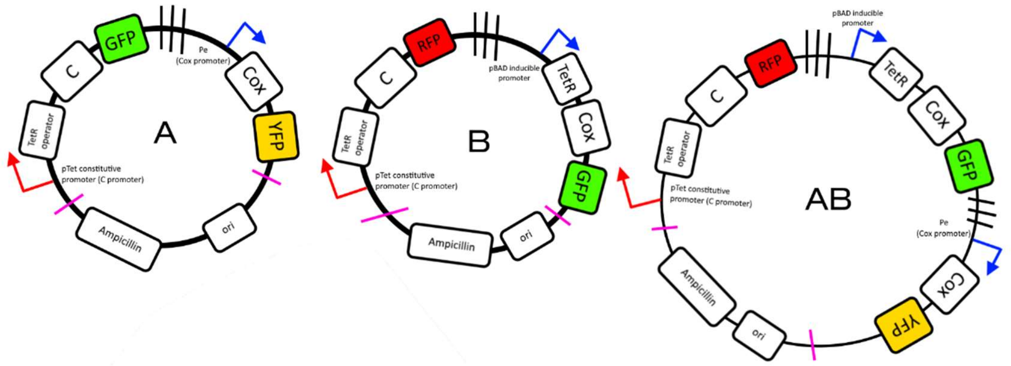


Figure 3: A simplistic representation of the plasmids we will use in our experiments.

Genetic circuit modeling – an overview

Euler forward

Modelling is based on writing functions for the rate of change in a concentration or population of some kind. These changes in concentrations/populations are then manipulated in a way to calculate the concentration/population at a certain time point. There are many ways of manipulating different mathematic axioms to derive different methods for doing this, such as Runge-kutta and Eulers methods, each with their benefits and issues. Our chosen method is Euler Forward, as it is simple and effective.

Eulers method is derived from the equation for the slope of a graph (Eq. 1), as presented below.

$$\frac{dy}{dx} = \frac{y_2 - y_1}{x_2 - x_1} \qquad \text{(Eq. 1)}$$

The slope of a point in a graph is calculated using this formula and is done so accurately if the distance between the points is very small.

Let's say the concentration of protein x changes over time (t), the slope (the change) in a given instance is denoted as $f(x_i, t_i)$ and the instance following that of (x_i, t_i) is (x_{i+1}, t_{i+1}), using these denotations the previous equation can be rewritten as the following (see Eq. 2).

$$f(x_i, t_i) = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} \qquad \text{(Eq. 2)}$$

This equation can be rewritten by multiplying both sides with $(t_{i+1} - t_i)$ and then simplified by rewriting $(t_{i+1} - t_i)$ as presented in the following equation (see Eq. 3).

$$t_{i+1} - t_i = dt \ \rightarrow dt + t_i = t_{i+1} \qquad \text{(Eq. 3)}$$

This yields the following equation (see Eq. 4).

$$f(x_i, t_i) * dt = x_{i+1} - x_i \qquad \text{(Eq. 4)}$$

Finally, by adding x_i to both sides, we have the iterative method, Euler Forward (Eq. 5).

$$f(x_i, t_i) * dt + x_i = x_{i+1} \qquad \text{(Eq. 5)}$$

This method assumes that the change in concentration does not change over time and that demands the values of the initial point (x_0, t_0) are known and that there is a specific value for the change in time (dt). If the change in concentration can determined accurately, this method can be used to calculated the concentration at a given time point. For this to be applied to a genetic circuit, one needs to know what qualities of this circuit will affect the concentration of a given molecule.

Analyzing a genetic circuit

In a genetic circuit there will be constants that positively affect the concentration of a molecule and things that do the opposite. One such simple factor is used in Malthus's law, the per capita growth rate, where the growth rate of a population is proportional to its number (see Eq. 6).

$$\frac{dy}{dt} = ry \qquad \text{(Eq. 6)}$$

Where y is the size of the population, $\frac{dy}{dt}$ is the change in population over time and r is the per capita growth rate of the population – a factor that positively affects the concentration of people in an area. Say we add a per capita death rate to Malthus's law, a factor that negatively affects the population. See Eq. 7.

$$\frac{dy}{dt} = ry - ky \qquad \text{(Eq. 7)}$$

Note that the per capita death rate, k, has a minus in front of it, which signifies that the per capita death rate indeed has a negative effect on the population.

With this basic understanding of how to translate events in reality, such as growth and death, into mathematical models, we can move on to more complex systems and the potential of the results models can yield.

Graphs and modeling concepts

Now that we have and understanding of the background of modelling, we can evaluate what a model can yield in terms of results.

Let's review the predator pit model; a species disappears when the population becomes too small and has a maximum capacity. In the model below (see Eq. 8) there is the parameter r, which is the per capita growth rate as well as K, which is the carrying capacity of the population and θ which is the survival threshold.

$$f(y) = \frac{dy}{dt} = ry(1 - \frac{y}{K})(\frac{y}{\theta} - 1) \qquad \text{(Eq. 8)}$$

Using the Euler method let's study the population change over time using different initial values; let's create a time series. In this time series we set K to 100, θ to 30, r to 0.5 and the initial values for the population 5, 25, 50 and 100.

By reviewing the figure below (figure 4) we notice that the population can reach two different steady states, 0 and 100. This means that the initial value of the population determines whether the population thrives or goes extinct. We need to do model analysis to determine the conditions and reasons for the model's behavior.

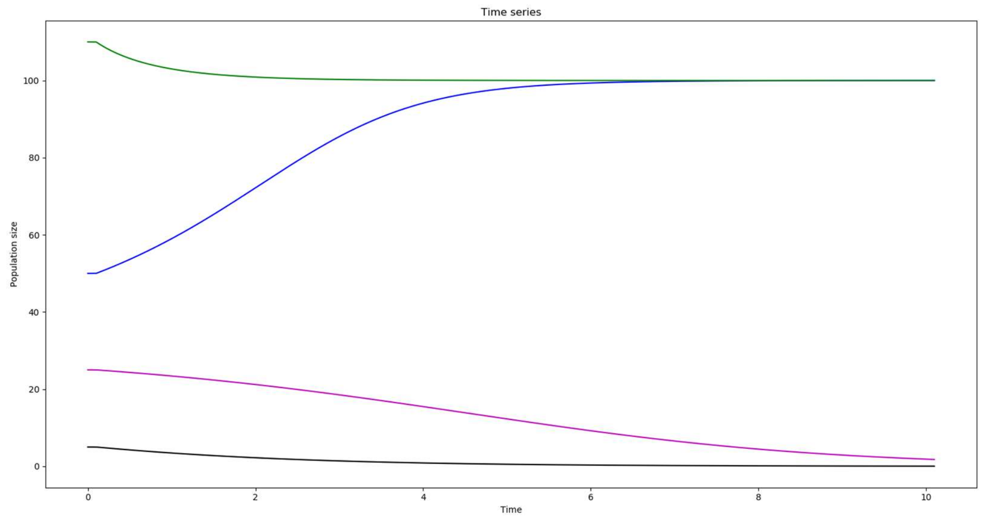


Figure 4: Time series of the predator pit model. K = 100, θ = 30, r = 0.5 and the initial values for the population was 5, 25, 50 and 100.

A graph for determining the steady states of a model is a steady state analysis, in which we plot the change of y ($f(y)$) on the y-axis and the concentration on the x-axis. This way we find the concentrations when $f(y)$ equals zero, which means the population has stopped changing and therefore have reached a steady state. The concentrations that cause $f(y)$ to equal zero are called fixed points, y*. Using the same values for the parameters as before, let's create a steady state analysis and analyze what we see.

By reviewing figure 5, we confirm y_1* = 0 and y_2* = 100 are fixed points, but we also see that there is a third one, y_3* = 30. This shows us a concept called stability. The stability of a fixed point determines whether the population is attracted to or repelled from the value of the fixed point. As we can see in the previous graph, figure 4, the concentrations move away from y_3* and towards y_1* or y_2* even if they are closer to y_3* – the fixed points y_1* and y_2* are stable while y_3* is unstable. This is seen in the steady state analysis, the slope at y_1* and y_2* is negative while the slope at y_3* is positive.

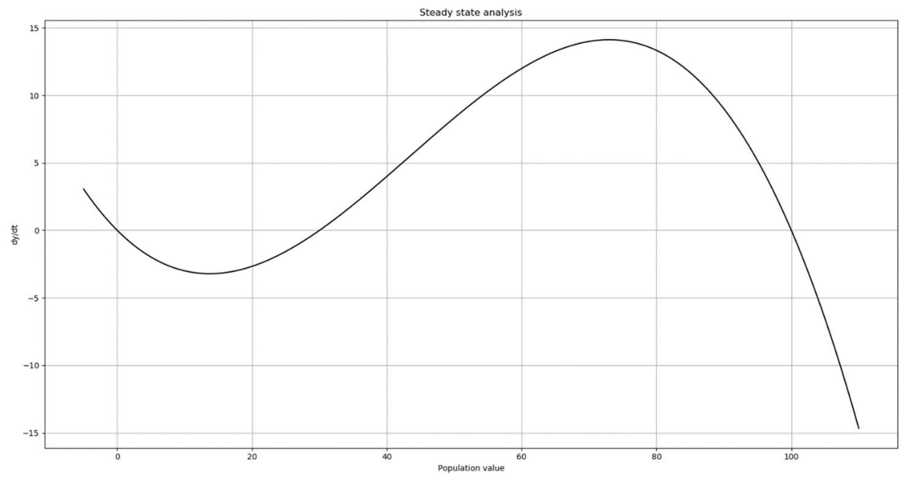


Figure 5: Steady state analysis of the predator pit model. K = 100, θ = 30, r = 0.5 and the population value ranged from -5 to 110.

To summarize, the steady state analysis shows us at which values the model will reach a steady state and if those values are stable or not.

In regard to stability, a concept worth noting is bistablility. A bistable model is a model which has two steady states, just like the predator pit function. This is especially relevant for our project. We want to manipulate a switch; inactive to active, lysogenic to lytic. By changing the conditions of the system we want to move from one steady state to the other – we want a system that is bistable.

Another analysis worth doing is the bifurcation analysis. This analysis tells us how the parameters affect the value of the fixed points, which is to say, how each parameter affects the long term of the model. A bifurcation analysis is done by plotting a range of values for a parameter on the y-axis and the fixed points when using that parameter value on the x-axis – so we are investigating the fixed points of the model given a certain parameter value. Let's do a bifurcation analysis of the parameter θ and see what we learn.

By reviewing figure 6 we can note that for all values of θ there are three fixed points, two of which are always 0 and 100 while the third is always equal to θ. This tells us that the unstable steady state is always equal to the value of the θ parameter, which in turn exposes the interval at which the population will either survive or die. This makes sense as the θ parameter represents the survival threshold, the minimum value for the species survival.

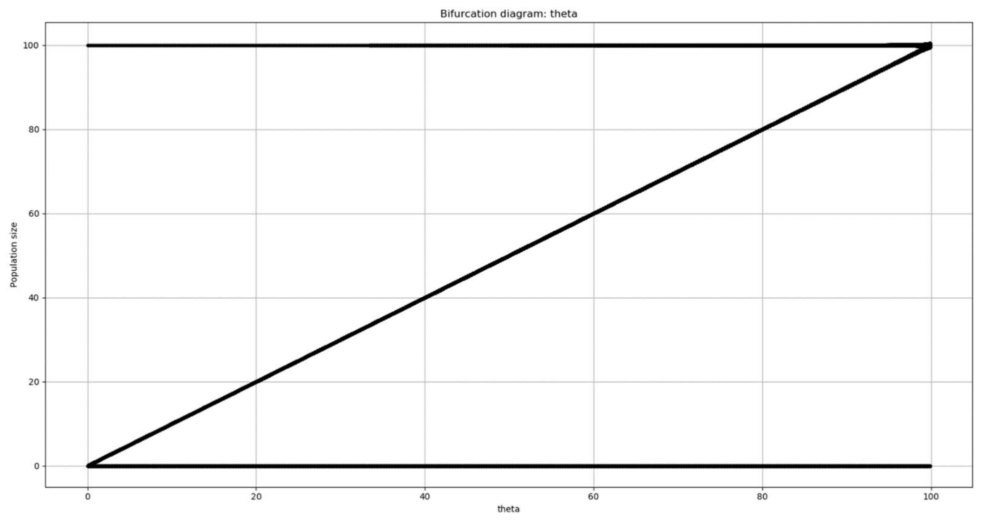


Figure 6: Bifurcation analysis of the predator pit model, evaluating θ. K = 100, r = 0.5 and θ ranged from 0.001 to 100.

When we do this for the parameter K, in figure 7, we see a similar behavior. In the range of K=[0,30)U(30,100] we can see that there are three steady states, one being 0, the second being the value of θ (30) and the third being the value of K. When K is equal to 30 we see only two steady states, which is due to K and θ being equal. This makes sense as K is the carrying capacity, which means that the value of K is the maximum value that the population can take.

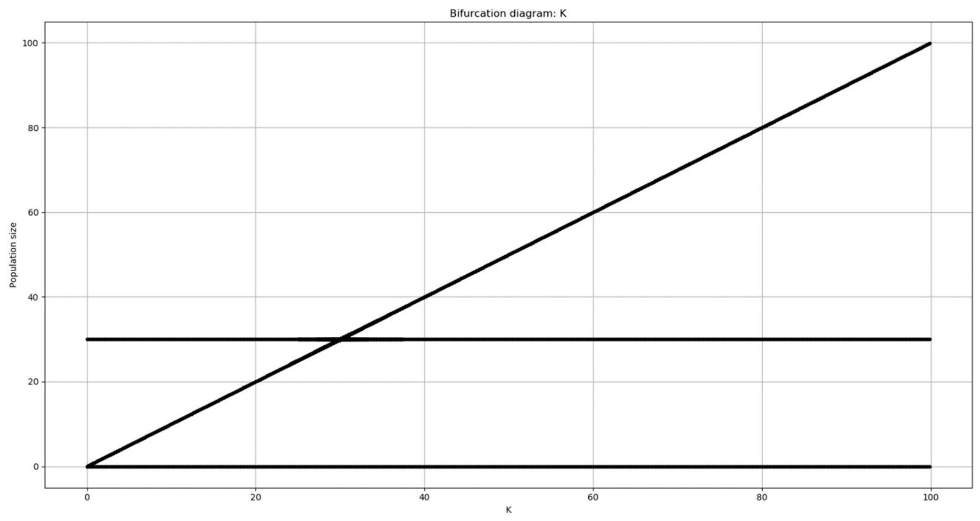


Figure 7: Bifurcation analysis of the predator pit model, evaluating θ = 30, r = 0.5 and K ranged from 0.001 to 100.

The bifurcation analysis yields not only knowledge of the limits of the system, but also confirms that the chosen parameters affect the system as expected. When adapting the model to real data, the bifurcation analysis can tell you what you need to change to fit the model.

Goals and objective

Modelling is becoming increasingly popular and is a one of the requirements for a gold medal in the iGEM competition this year. For this reason, this year's iGEM Stockholm team is making a big push in developing a strong modeling subproject to complement the whole teams greater goal.

Our objective is to formulate a model for the P2 switch as described in the introduction and assist the wet lab team in their study of the system.

Methods

There are several ways to approach modeling, especially if the system hasn't been examined before. As a group, Dry Lab members have chosen to do certain things differently, but we agreed to work on the same kinds of modelling (deterministic, stochastic etc.) and to use Python and MatLab. The following is the way I have worked.

Deterministic modeling

Detministic modeling can be used as a base for stochastic modeling and only demands the Euler method.

As explained in the introduction under Analysing a genetic circuit, I approached making a model of the system by assessing the factors that positively and negatively affect the concentrations of the C, Cox and TetR protein. This is how I summarized the system:

The factors that affect [C]:

+ Effectiveness of the pTet promoter in binding to the RNA polymerase
    - Should just be a constant ➔ P_{pTet}
        - It depends on the amount of plasmids and the number of RNA polymerase, but we assume both are constant and it can therefore be summarized into one constant.
- Effectiveness in TetR binding to the TetR operator which is strongly connected to [TetR].
    - Should be a constant and proportional to concentration of TetR ➔ $I_{TetR}[TetR]$
- ± Copy number
    - Unknown, save for later
- Degradation of C
    - Should be a constant and proportional to the concentration of C ➔ $D_C[C]$
        - If there is a great abundance of C protein then there is more C protein being degraded.
- Dimerization of C
    - It should be the rate of dimerization and be proportional to the concentration of C ➔ $M_{C_2}[C]$

The factors that affect [Cox]:

+ Effectiveness of the Pe promoter binding to the RNA polymerase
    - Should just be a constant ➔ P_{Pe}
- Effectiveness in the C dimer binding to Pe promoter which is strongly connected to [C_2]
    - Should be a constant for the effectiveness in C_2 binding to the promoter and be proportional to the concentration of C_2 ➔ $I_{C_2}[C_2]$
- The tetramerization of Cox
    - Should be a constant for the rate of tetramerization and be proportional to the concentration of Cox ➔ $T_{Cox_4}[Cox_4]$
- The effectiveness in Cox_4 binding to the Pe promoter

    - o Should be a constant for the effectiveness in Cox_4 binding to the promoter and be proportional to the concentration of C_4 ➔ $I_{Cox_4}[Cox_4]$
- + Effectiveness of the pBAD promoter binding to the RNA polymerase
    - o Should just be a constant ➔ P_{pBAD}
- ± Copy number
    - o Unknown, save for later
- - Degradation
    - o Should be a constant and proportional to the concentration of C ➔ $D_{Cox}[Cox]$

The factors that affect [TetR]:

- + Effectiveness of the pBAD promoter binding to the RNA polymerase
    - o Should just be a constant ➔ P_{pBAD}
- + Effectiveness of Arabinose inducing the promoter, which is strongly linked to [Arabinose].
    - o Should be a constant and proportional to the concentration of Arabinose ➔ $I_{Ara}[Arabinose]$
- ± Copy number
    - o Unknown, save for later
- - Degradation
    - o Should be a constant on combination with the concentration of C ➔ $D_{TetR}[TetR]$

The finished model:

$$f(C) = \frac{d[C]}{dt} = P_{pTet} - I_{TetR}[TetR] - D_C[C] - M_{C_2}[\mathrm{C}]$$

$$f(Cox) = \frac{d[Cox]}{dt} = P_{Pe} + P_{pBAD} - D_{Cox}[Cox] - I_{C_2}[C_2] - T_{Cox_4}[\mathrm{Cox}] - I_{Cox}[Cox_4]$$

$$f(TetR) = \frac{d[TetR]}{dt} = P_{pBAD} + I_{Ara}[Arabinose] - D_{TetR}[TetR]$$

This model can be applied to all plasmids.

The code (for both Python and MatLab) showing how I deterministically modelled the time series, the steady state analysis and the bifurcation analysis can be found in the GitHub link under References.

Stochastic modelling

Stochastic simulation data can be used for determining event waiting times and autocorrelations. A deterministic model will always yield the exact same results every time you run it, while the stochastic approach takes into account the heterogeneous nature of cells and how the concentrations of proteins may differ from cell to cell. The change in species amount is discrete, not continuous. However, stochastic modelling is time consuming and demands a lot of processing power[7], to get past this we chose a method similar to Euler's.

As a group we chose to work with the Tau-leap method. The Tau-leap method speeds up the stochastic simulation by skipping some values without loosing accuracy[8]. The method, as I have understood it, uses the Poisson random variable function in Python and we input each factor multiplied by Tau (who's value we assumed to be 0.01 which was recommended by a previous iGEM team). The modified factor is then added to or subtracted from (depending on the effect the factor has in the model) from the calculated $f(x)$ to create a modified $f(x, \tau)$. This modified $f(x)$ is then used in the Euler method to calculate the concentration of protein x.

The code used to do this was only written in Python, and also can be found in the GitHub folder.

SimBiology

SimBiology is an app on MatLab focused on biological applications. It has programmatic tools to model, simulate and analyse dynamic systems[9]. Basically, I inputted our translational system into the app and allowed the program to create the equations, given my inputted conditions.

In the figure below (figure 8) presents the plasmid AB and what reactions occur in this specific system. It seems rather intricate but its more reliable than the generalized model as previously described as it encompasses all of the reactions that takes place in the system.

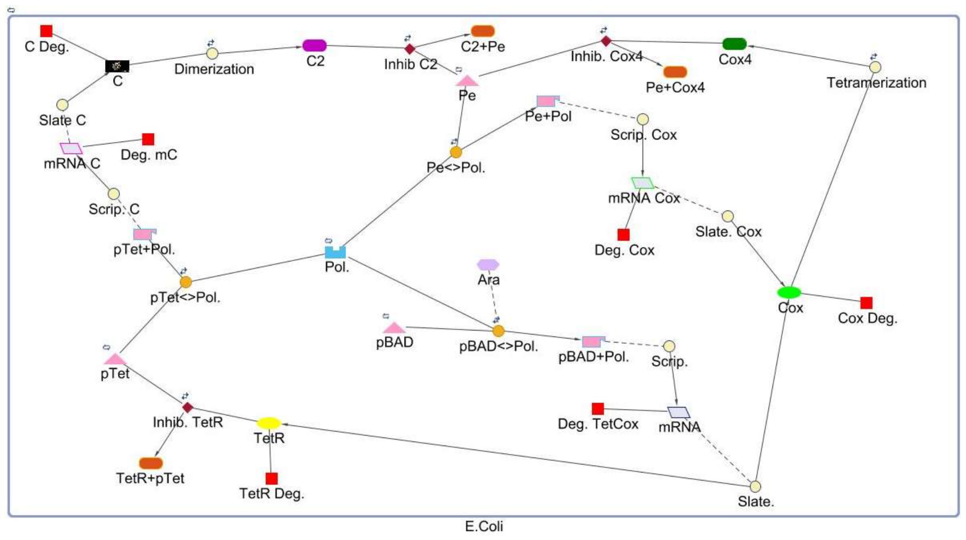


Figure 8: Plasmid AB as seen in SimBiology.

Once inputted in the app, it can simulate a time series and sensitivity analysis. A sensitivity analysis is an extension of the bifurcation analysis as it illustrates the effect a parameter has on a species.

The .sbproj and .xml files are available in the GitHub link,

3D Modeling

3D modeling is always an eye-catching and engaging type of modeling as we can visualize the molecules. Our goal with 3D modeling is to study the binding energy between the Pe promoter and the C_2 dimer. The programs I used were Chimera and AutoDock.

The .pd-file for the dimer was found on the RSCB Protein Data Bank and, using the the DNA sequence for the Pe promoter, the DNA structure was created using 3D-DART. The figures below (figure 9 and 10) are images of the molecules.

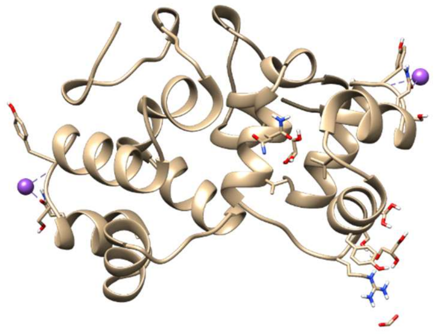


Figure 9: Image of the C_2 dimer taken on Chimeria[10].

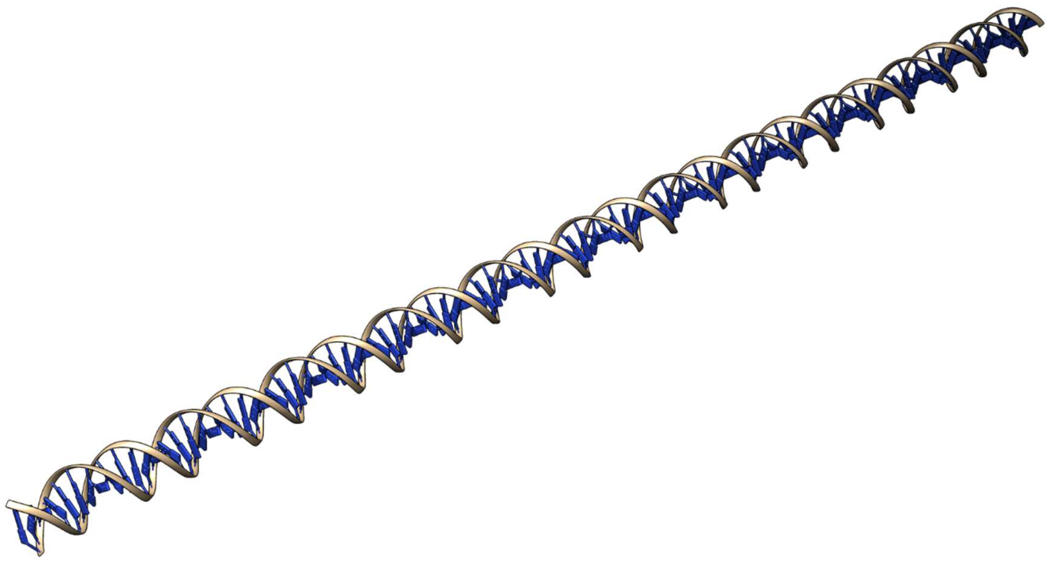


Figure 10: Image of the Pe segment in the plasmids A and AB[11]. The sequence used was:
ATCTGCAACTTCAAGACCACATACAGATCCAAGAAACCCGCTAAGAACCTCAAGATGCCCGGCGTCTACTATGTGGACC
ACAGACTGGAAAGAATCAAGGAGGCCGACAAAGAGACCTACGTCGAGCAGCACGAGGTGGCTGTGGCCAGATACTGC
GACCTCCCTAGCAAACTGGGGCACAAACTTAATTGATACTAGAttgacatggtgtttagatctcaatagtatttagtttagatgtagattgtttagtgct
tggatgtgggcactaaaagggctagaggacgacatgagcaagcaagtaacactcatgactgatgcgattccttatcaggagttcgcaaaactaataggaaaatcgac
aggagcggttcgtcggatgatcgataaaggaaagctgcctgtaattgatatgaccgatccacaatcagcttcaggtcgtgcaggtgaatattgggtataccttccggcat
ggaataacggactaaaactggcttatgaaagccgccctaaagagattcgtgacggctggttgatgtg.

Here are the exact steps I took:

- Open AutoDockTools (If it doesn’t open after the loading screen is at 100% click Register Later.

- Then click on File > Read molecule > choose your .pbd file for your C protein (make sure to put everything for AutoDock in 1 file)
- To create a .pbdqt file do this
    - Edit > Delete Water
    - Edit > Hydrogens > Add > Check Polar Only > OK
    - Edit > Hydrogens > Merge Non-Polar
    - Edit > Charges > Add Kollman Charges > OK
    - Edit > Charges > Compute Gasteiger > OK
    - Grid > Macromolecule > Choose > Select your C protein file (should be the only thing to choose) > Select Molecule > OK > save as [Whatever name you use for C protein].pdbqt > Save
    - Close Autodock and repeat for the Dna file of .pbd
- Open the DNA and C protein .pbdqt file and prep for docking
    - File > Read molecule > choose your .pbdqt
- Grid > Set Map Types > Choose Ligand > Select the file DNA (doesn't work with the C protein) > Select Ligand > OK
- Grid > Grid box > Cover the whole C protein > File > Close saving current
- Grid >Output > Save GPF > Name it as you want, ex. dna_ 2xcj.gpf. Make sure to write out ".gpf" when naming it, or it wont save properly > Save
- Run > Run AutoGrid > in the window pop-up:
    - Program Pathname: Click Browse and select autogrid4.exe in your working directory.
    - Parameter Filename , click Browse and click on your .gpf file (dna_ 2xcj.gpf)
    - The rest of the information is filled in automatically
- Launch
    - This step will generate several grid map files for use in the docking step and should take 10-30 seconds.

Here is where I met my first road block, Autodock doesn't understand how to handle Na atoms. I tried adding the parameter data to AD4_parameter.dat but it still doesn't work. I have yet to manage repairing this.

When I cheat my way past the first block I have a problem with docking; the C protein is too big for AutoDock to handle (2248 when it actually handles 2048). On the web it says to change the constants.h file and change the MAX, but I have yet to find that. Another idea is to remove the nonessential segments of the C protein, but that may change the structure. Other than yielding the images above, I haven't managed to yield any results.

Results

Graphs showing C and Cox protein dynamics in a deterministic setting, stochastic setting, results of analyses.

The model has not yet been fitted to any data, so all of the graphs here may be completely different from what they will look like when fitted. With the acceptation of the sensitivity analysis results from SimBiology, everything was plotted in both MatLab and in Python. All the plots are made by written code unless otherwise indicated.

Deterministic model results

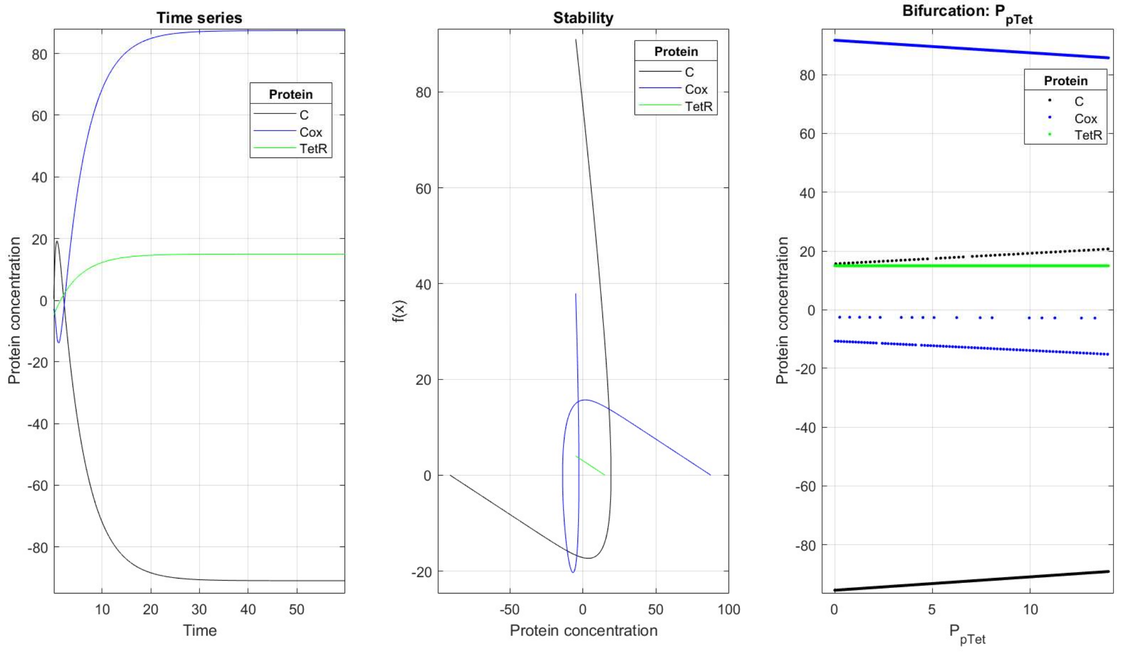
The following plots contain a time series, steady state analysis and a bifurcation analysis of the AB plasmid. The studied parameter is varied from 0 -14 with steps of 0.01 and in the time series and steady state analysis has the value of 10. The initital concentration for all proteins are all set to -5, after doing test runs of the graphs this was deemed to yield the most interesting results. These are the values of each parameter in every graph when it's not being studied:

$P_{pTet} = 10$ $\quad I_{TetR} = 1$ $\quad D_C = 0.2$ $\quad M_{C_2} = 2$

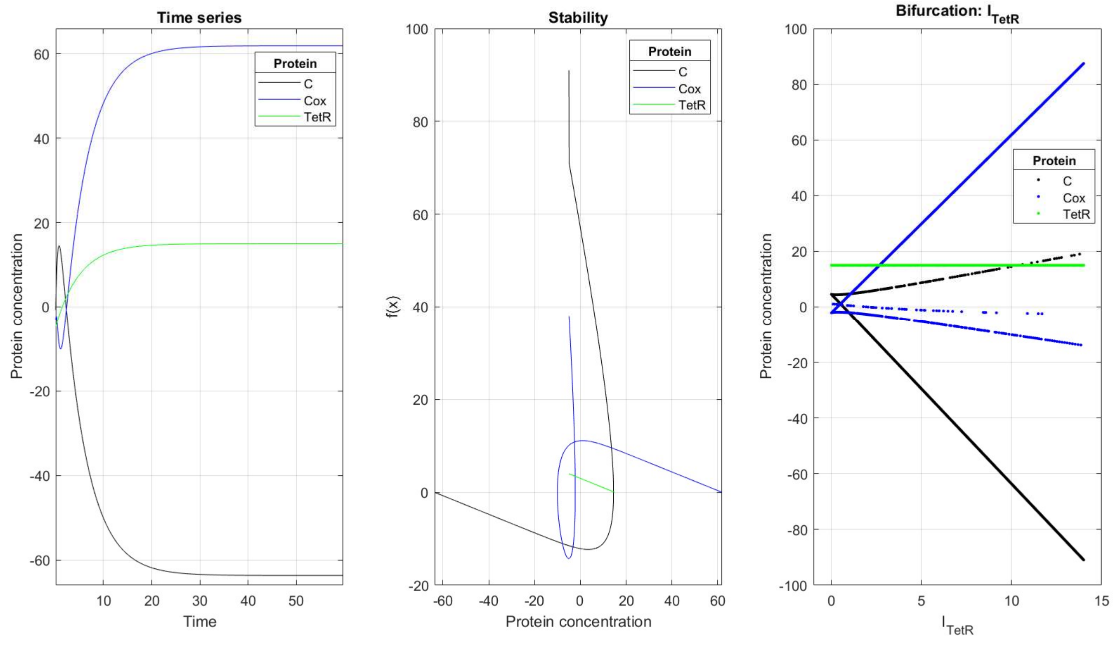
$P_{Pe} = 4$ $\quad P_{pBAD} = 3$ $\quad D_{Cox} = 0.2$ $\quad I_{C_2} = 1.5$

$T_{Cox_4} = 1$ $\quad I_{Cox} = 2$ $\quad I_{Ara} = 6$ $\quad D_{TetR} = 0.2$

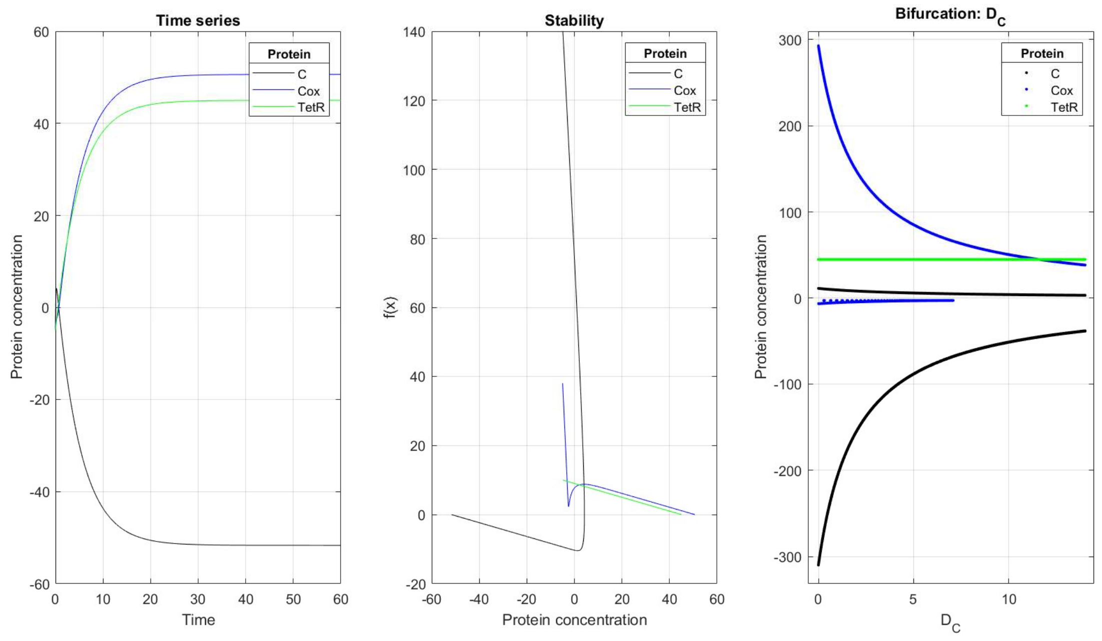
$[Ara] = 0$



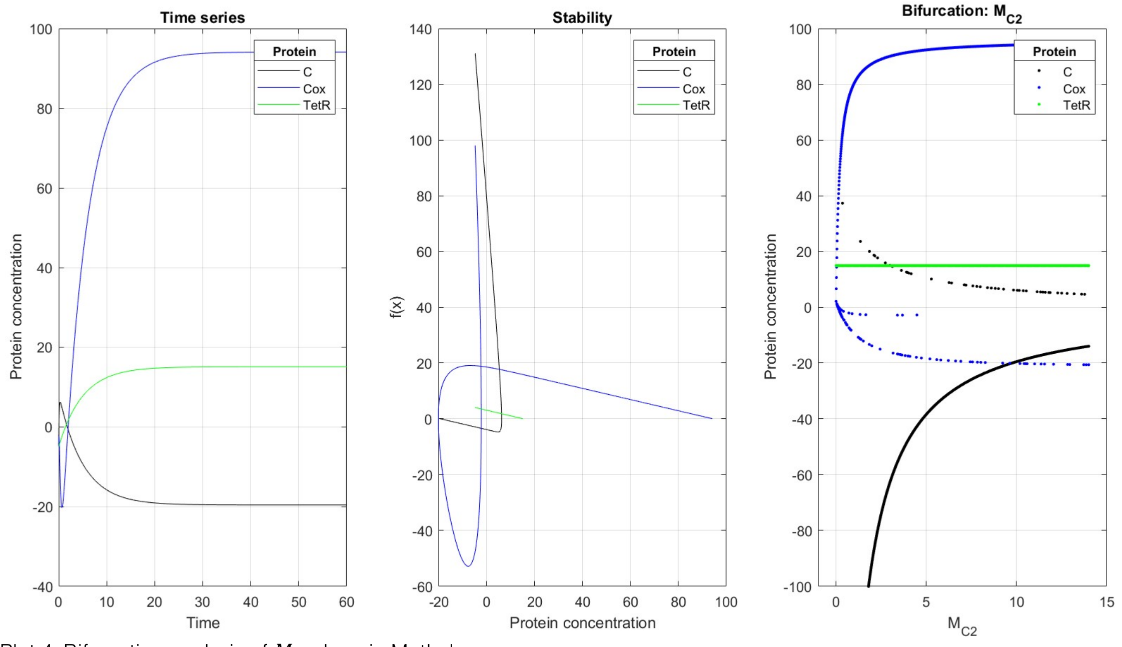
Plot 1: Study of P_{pTet} done in MatLab.



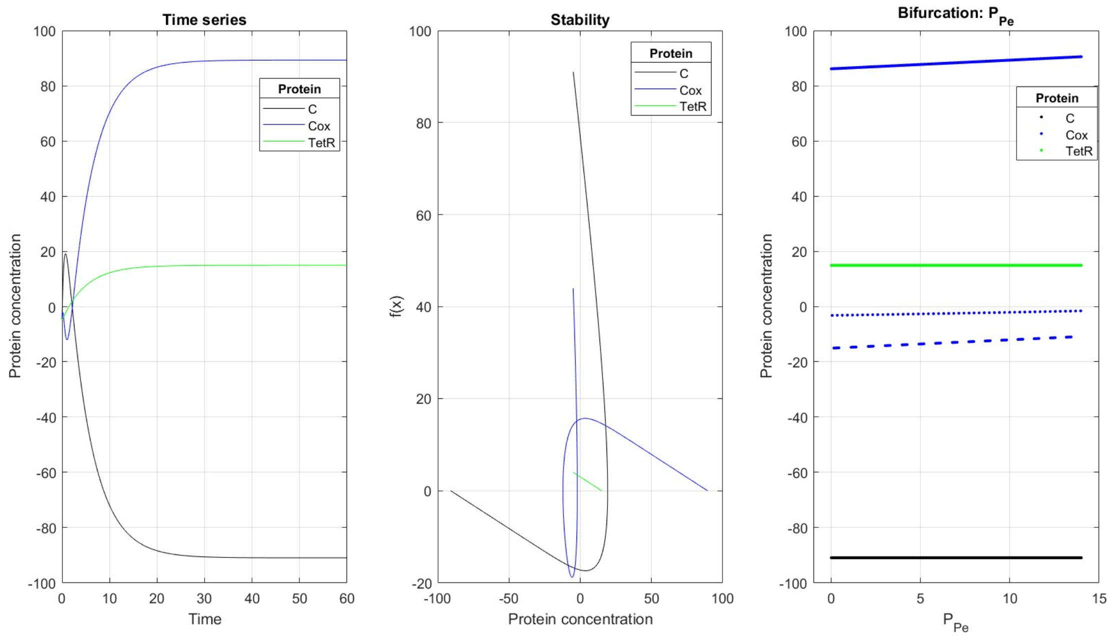
Plot 2: Bifurcation analysis of I_{TetR} done in MatLab.



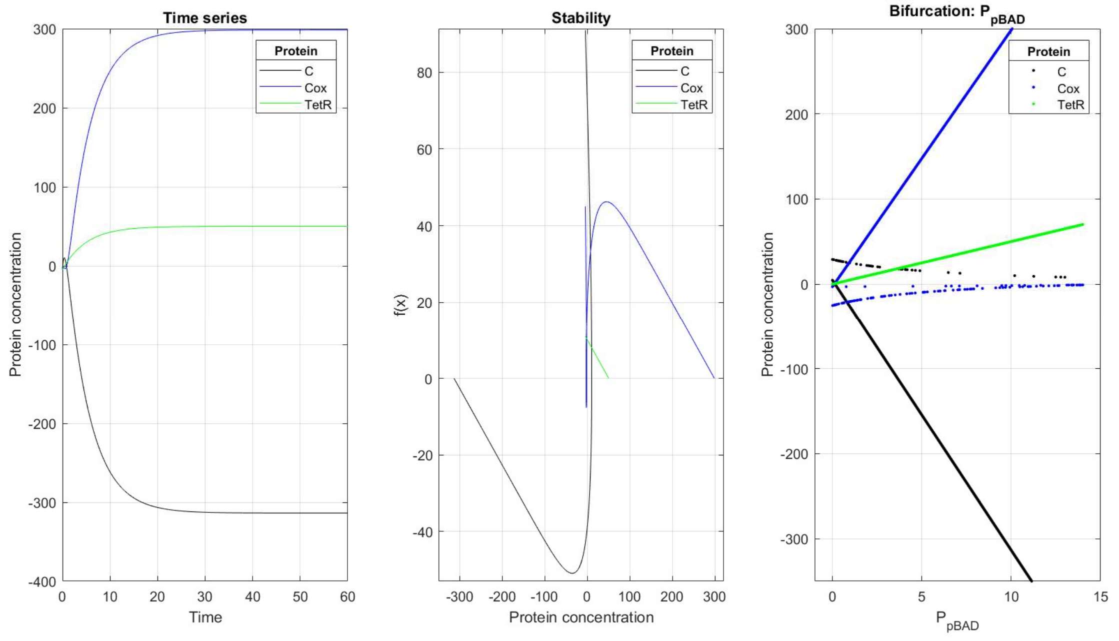
Plot 3: Bifurcation analysis of D_C done in MatLab. The concentration of arabinose was set to 1.



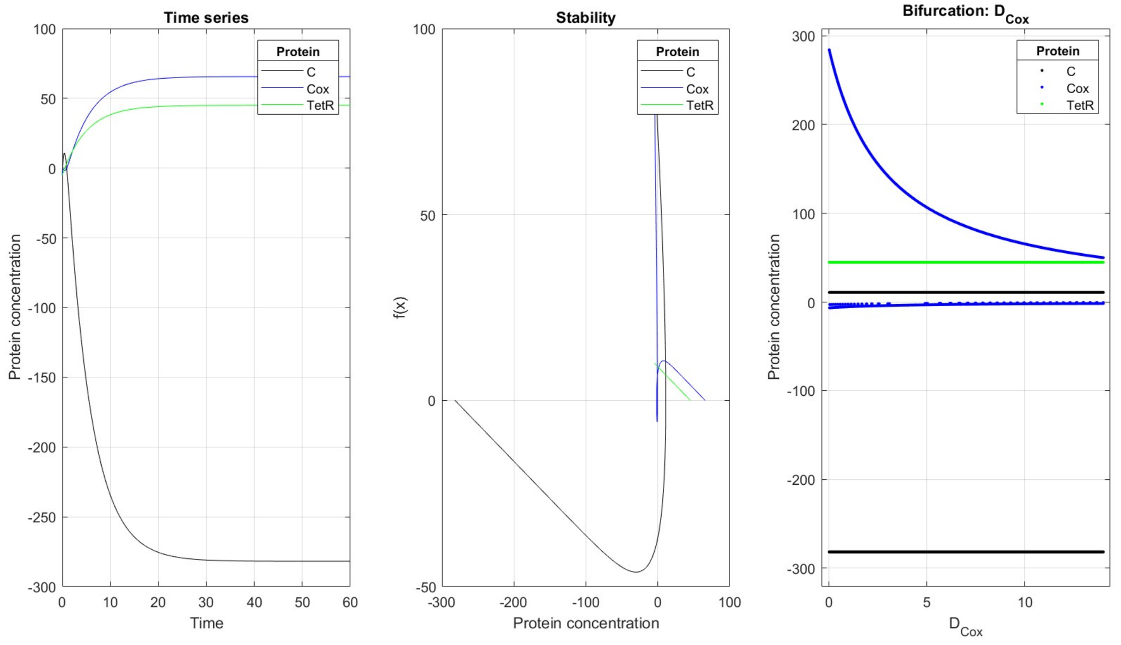
Plot 4: Bifurcation analysis of M_{C_2} done in MatLab.



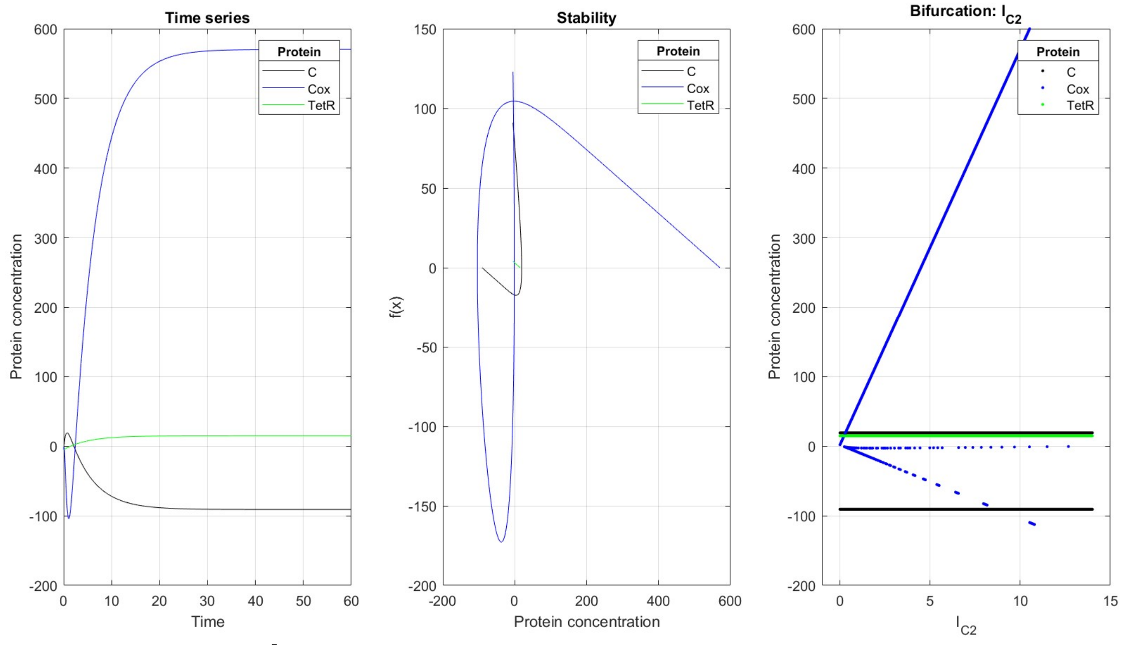
Plot 5: Bifurcation analysis of P_{Pe} done in MatLab.



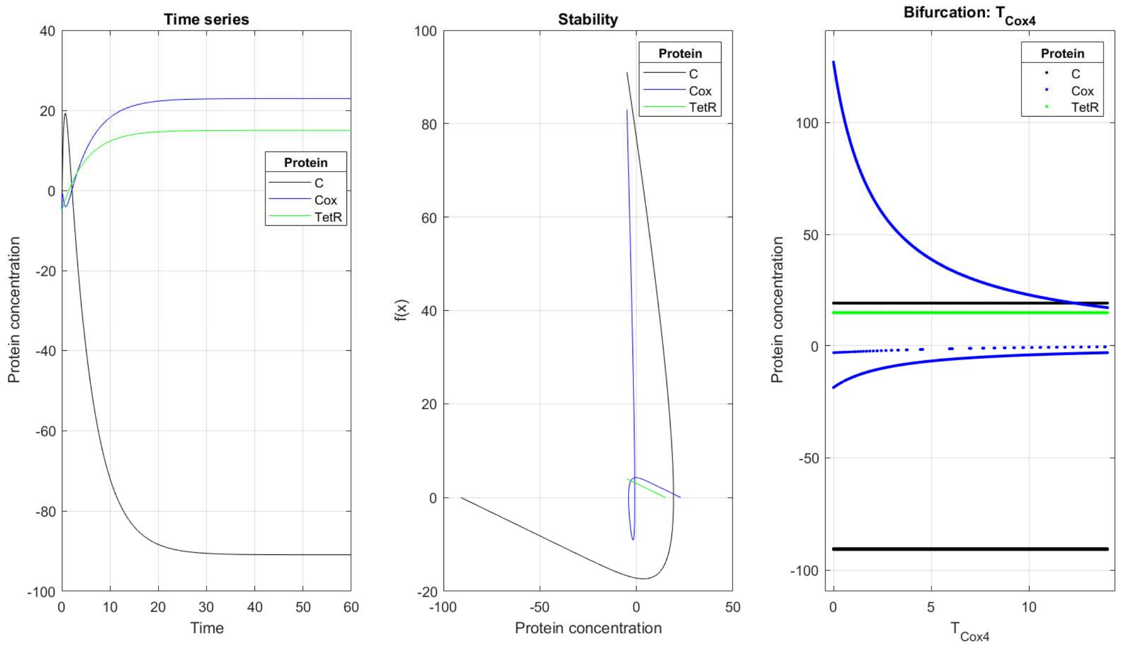
Plot 6: Bifurcation analysis of P_{pBAD} done in MatLab.



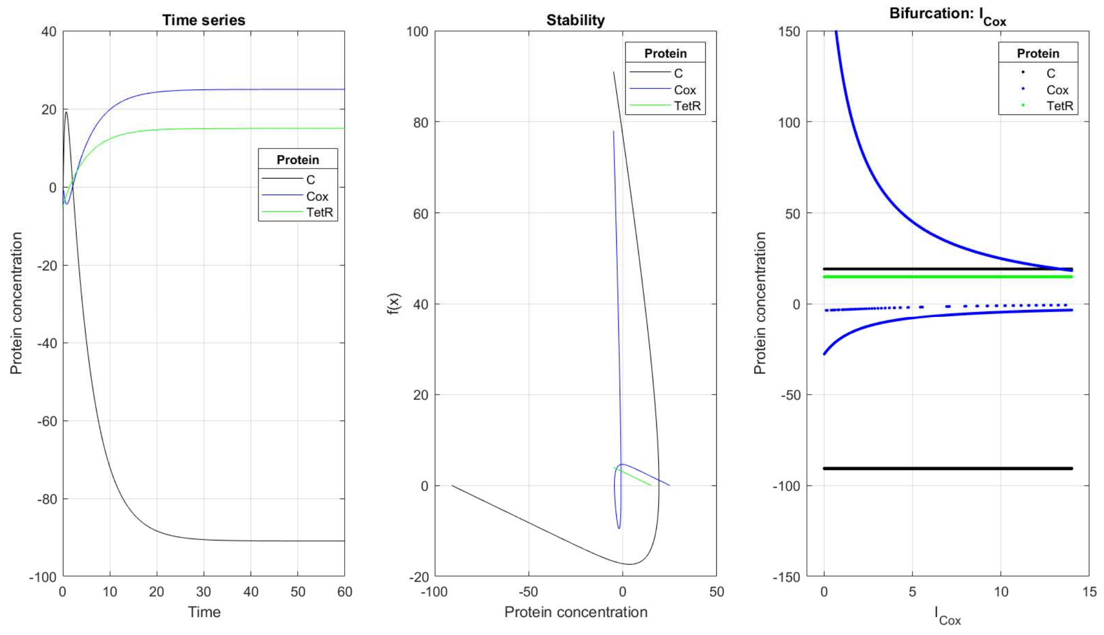
Plot 7: Bifurcation analysis of D_{Cox} done in MatLab. The concentration of arabinose was set to 1.



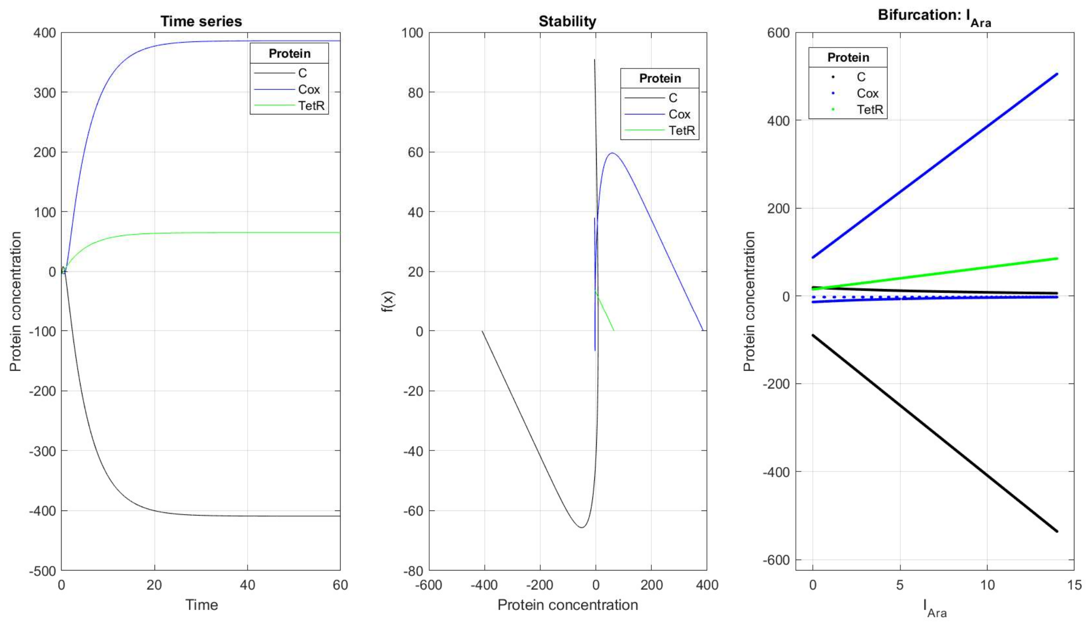
Plot 8: Bifurcation analysis of I_{C_2} done in MatLab.



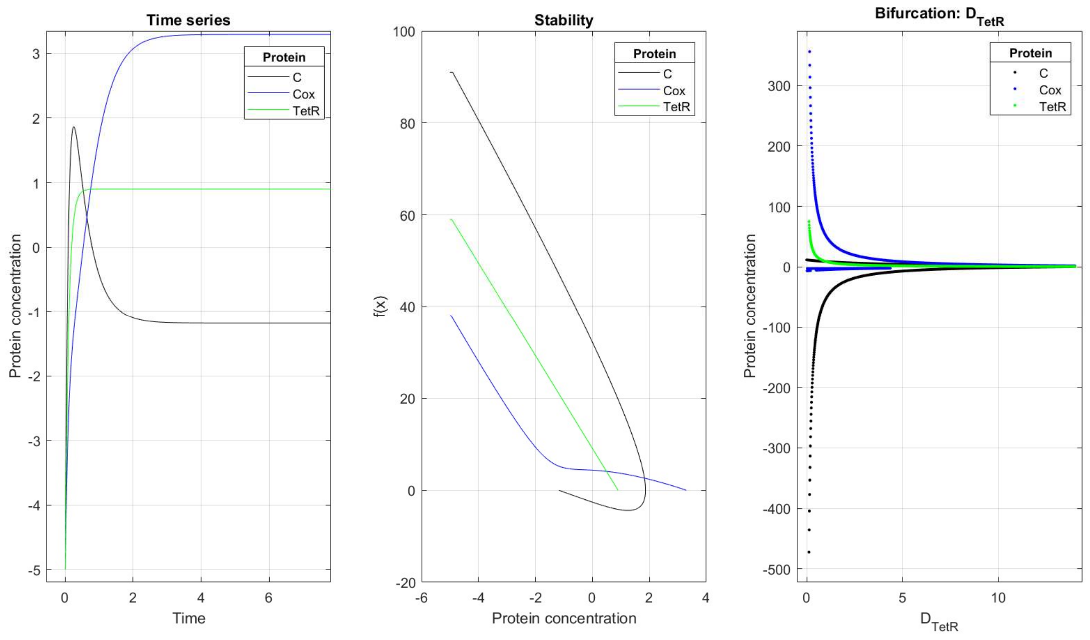
Plot 9: Bifurcation analysis of T_{Cox_4} done in MatLab.



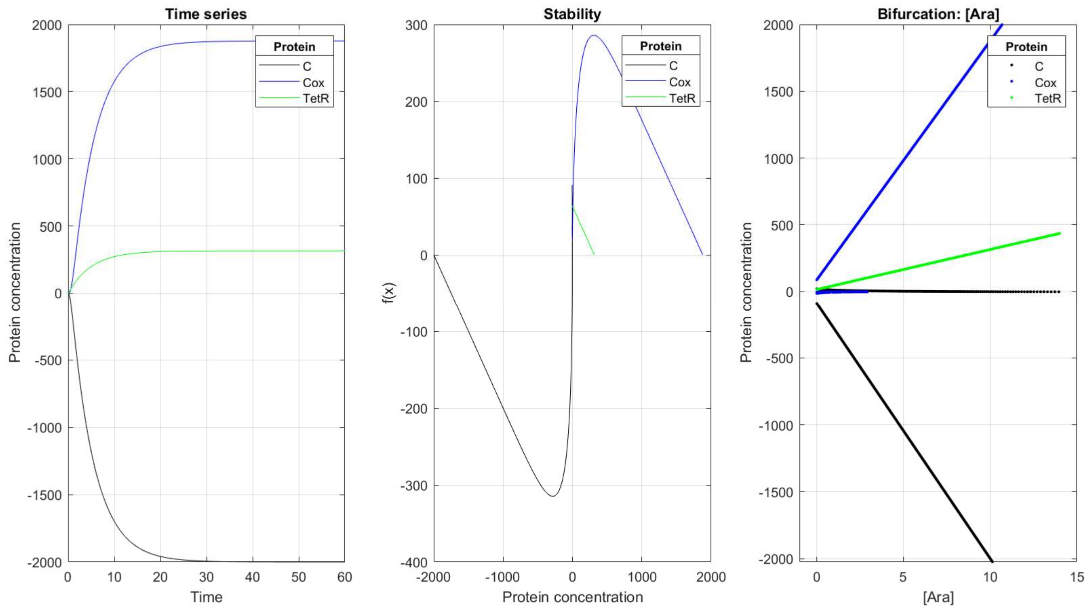
Plot 10: Bifurcation analysis of I_{Cox} done in MatLab.



Plot 11: Bifurcation analysis of I_{Ara} done in MatLab. The concentration of arabinose was set to 1.



Plot 12: Bifurcation analysis of D_{TetR} done in MatLab. The concentration of arabinose was set to 1.



Plot 13: Bifurcation analysis of $[Ara]$ done in MatLab. The concentration of arabinose was set to 1.

Stochastic model results

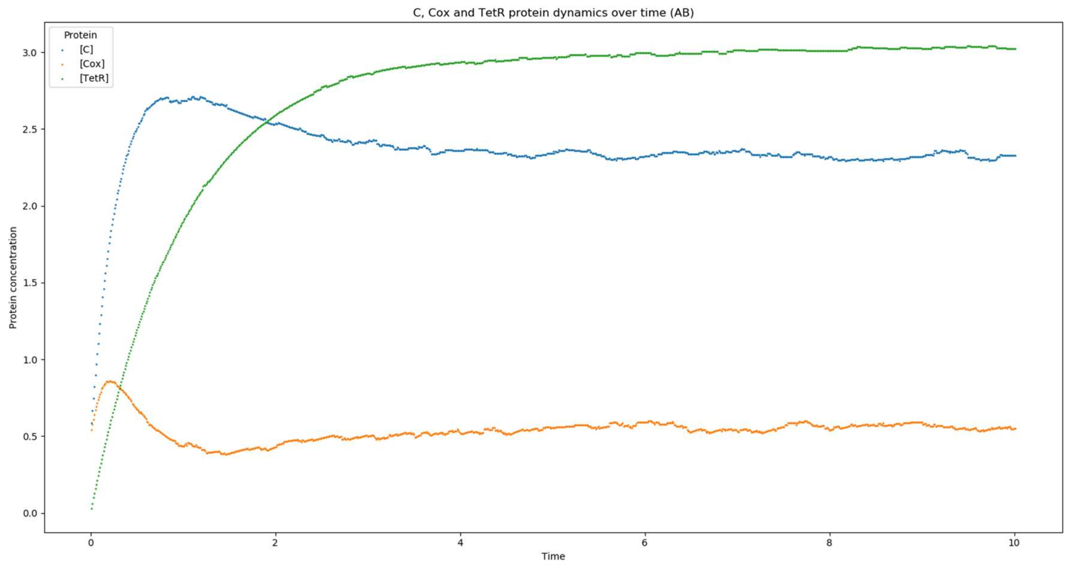
These are the values of each parameter in every graph unless otherwise indicated:

$P_{pTet} = 10$ $I_{TetR} = 1$ $D_C = 1$ $M_{C_2} = 2$

$P_{Pe} = 4$ $P_{pBAD} = 3$ $D_{Cox} = 1$ $I_{C_2} = 1$

$T_{Cox_4} = 1$ $I_{Cox} = 2$ $I_{Ara} = 6$ $D_{TetR} = 1$

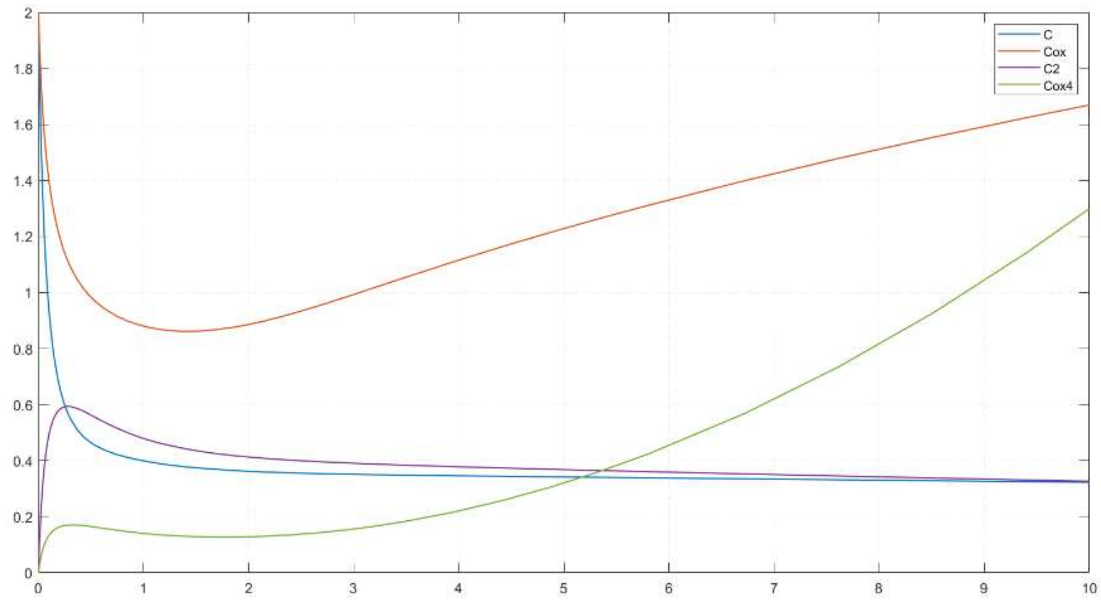
$[Ara] = 0$



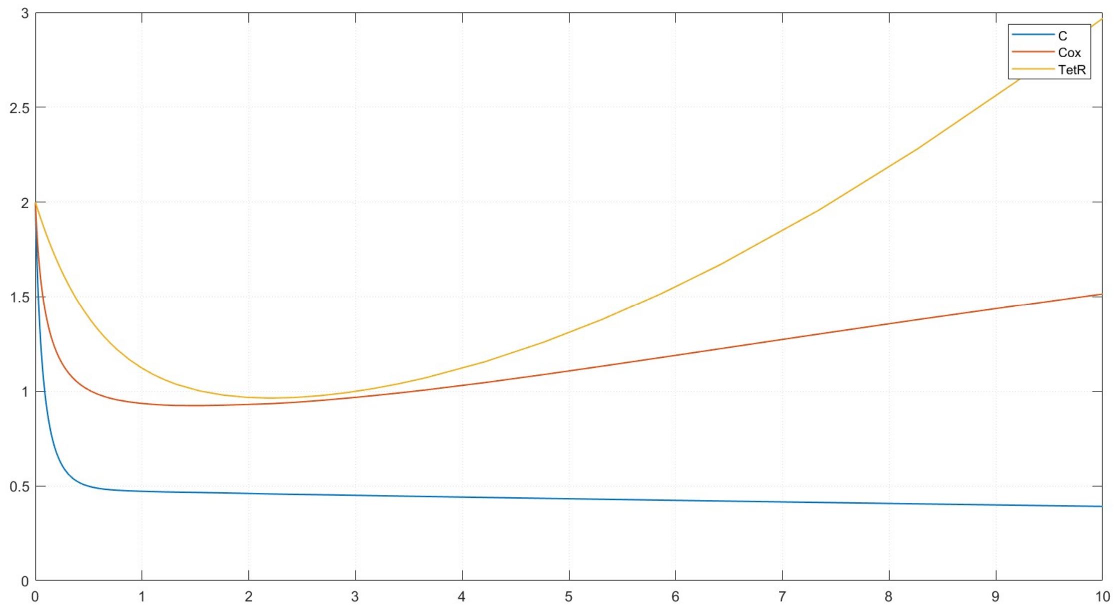
Plot 14: Time series done on Python.

SimBiology

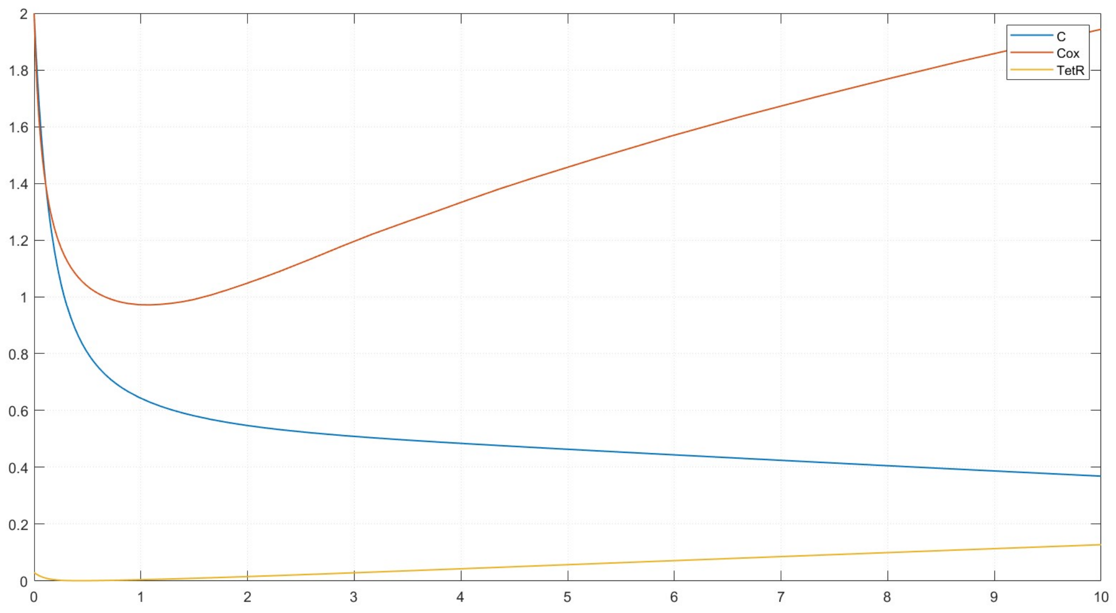
These are the results of the graphs run in Simbiology, the chosen parameter values can be seen in the project file shared under the GitHub link.



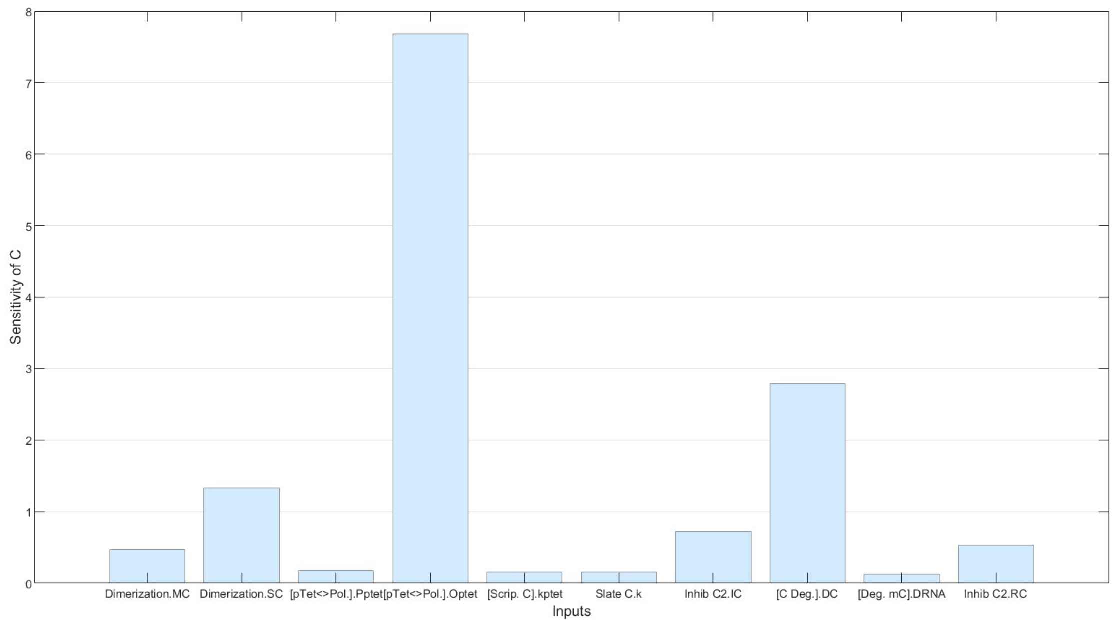
Plot 15a: Time series done in Simbiology. Plasmid A.



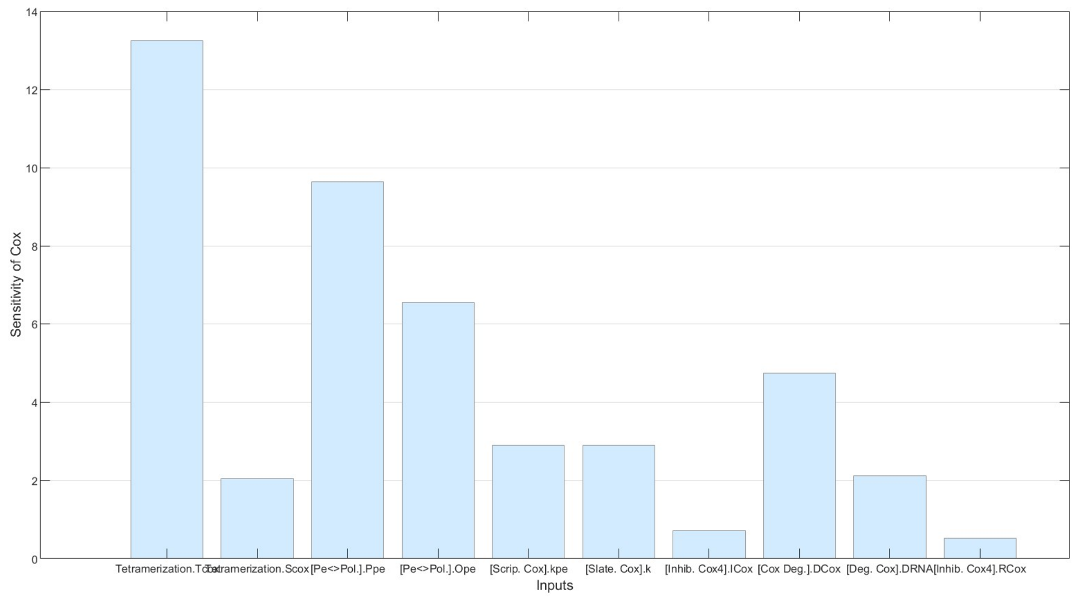
Plot 15b: Time series done in Simbiology. Plasmid B.



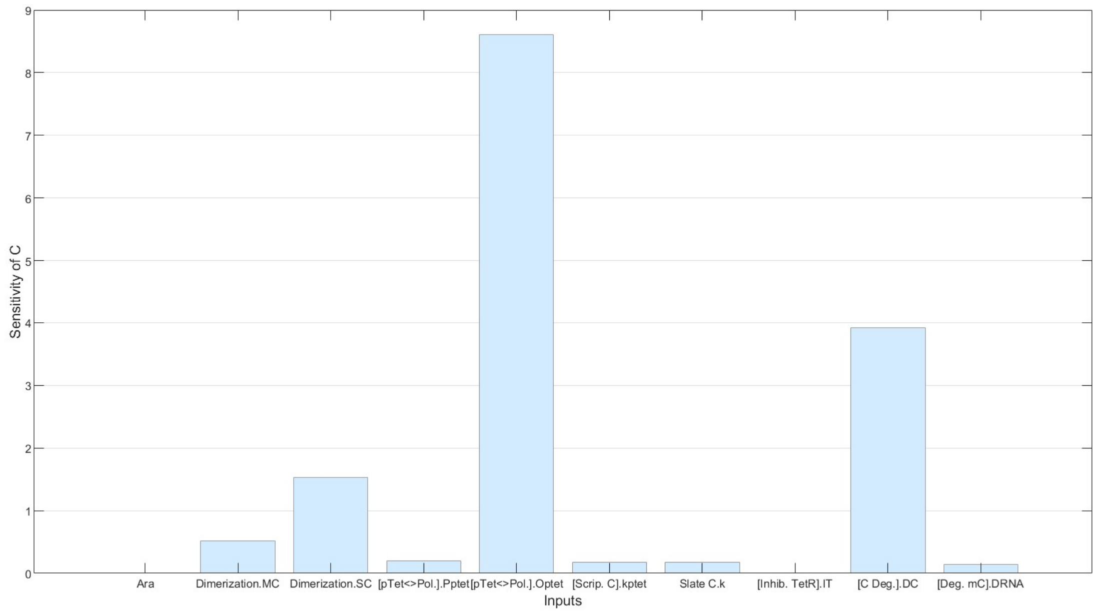
Plot 15c: Time series done in Simbiology. Plasmid AB.



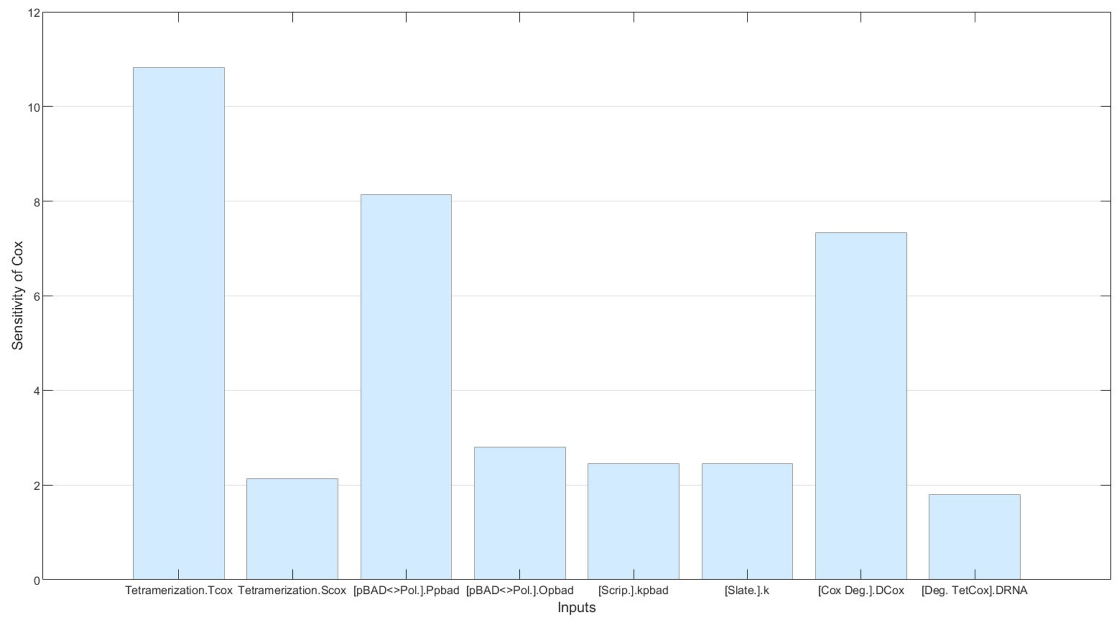
Plot 16a: Sensitivity analysis done in Simbiology. Plasmid A, protein C.



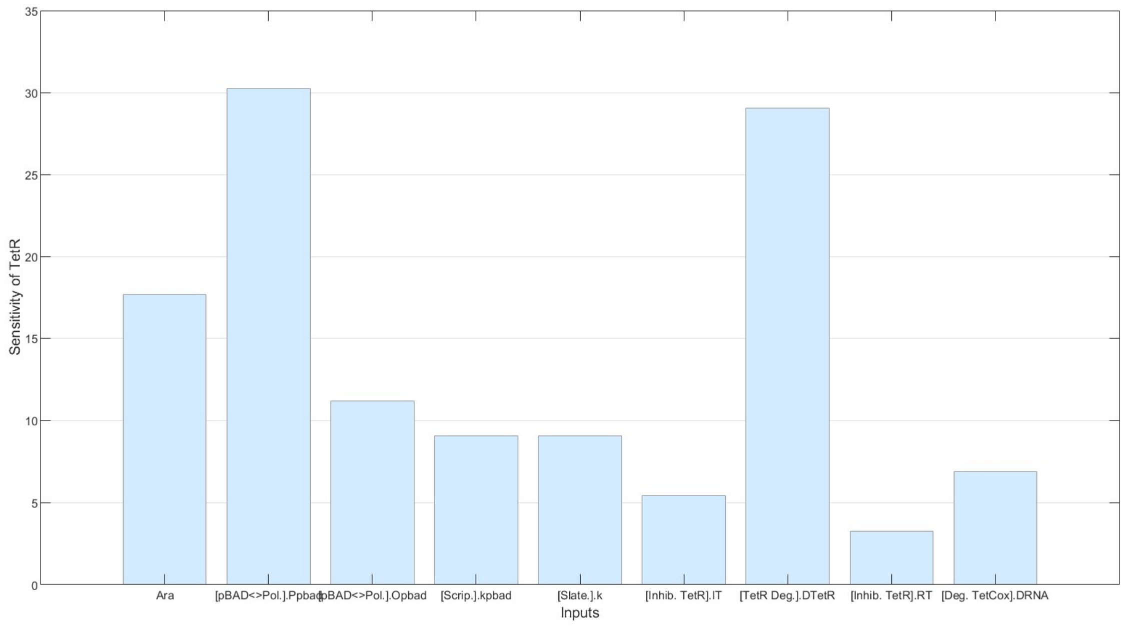
Plot 16b: Sensitivity analysis done in Simbiology. Plasmid A, protein Cox.



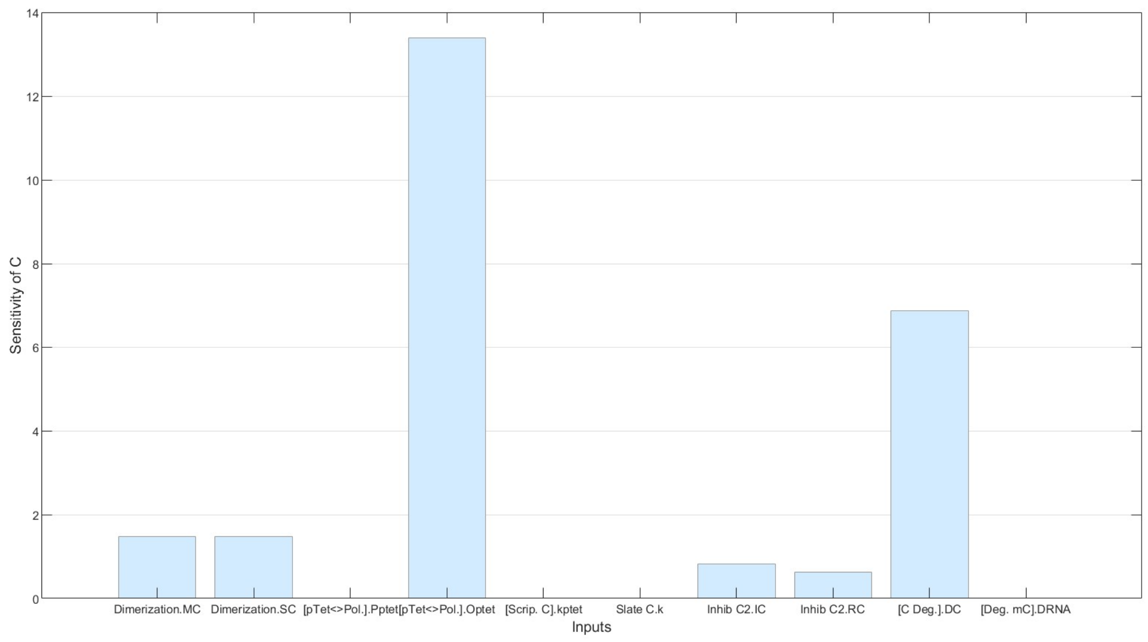
Plot 17a: Sensitivity analysis done in Simbiology. Plasmid B, protein C.



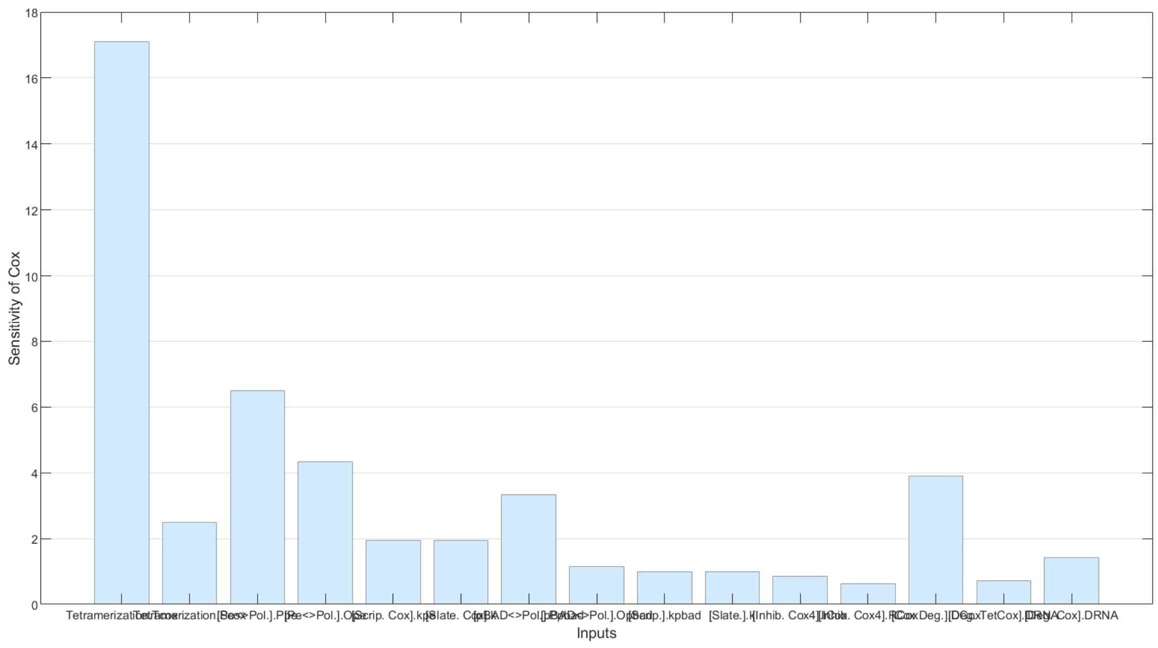
Plot 17b: Sensitivity analysis done in Simbiology. Plasmid B, protein Cox.



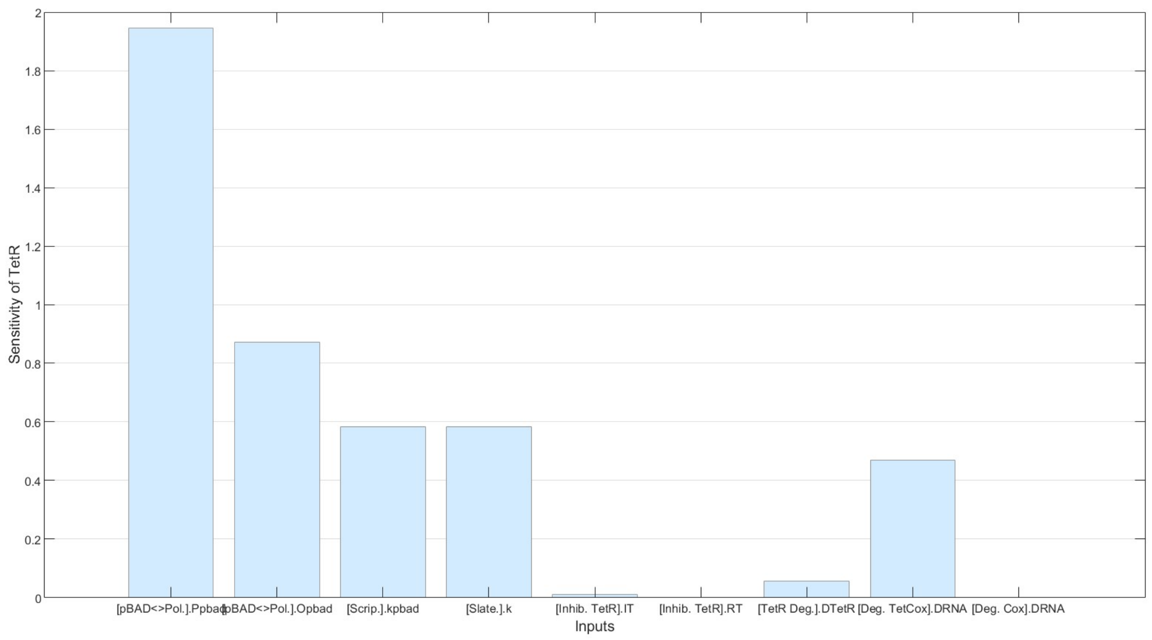
Plot 17c: Sensitivity analysis done in Simbiology. Plasmid B, protein TetR.



Plot 18a: Sensitivity analysis done in Simbiology. Plasmid AB, protein C.



Plot 18b: Sensitivity analysis done in Simbiology. Plasmid AB, protein Cox.



Plot 18c: Sensitivity analysis done in Simbiology. Plasmid AB, protein TetR.

Discussion

The results aren't exactly what we want, but all of the parameters are affecting the proteins in a reasonable way. We want bistability but at best we could observe impulses.

When strengthening the pTet promoter (see plot 1) the expression of Cox decreased, of C increased and the xpression of TetR remained unchanged. This is reasonable as the pTet promoter expresses the C protein which inhibits the Pe promoter which in turn inhibits the expression of Cox. TetR should be unaffected by this promoter. The Time series, stability and bifurcation plot correspond to each other (note that the growth of C grows, peaks and falls into a steady state, thus the stability graph crosses 0 twice and the bifurcation graph shows 2 lines, similar behavior can be seen with the other proteins) so this assessment is well grounded.

When strengthening the inhibition by TetR on the pTet promoter (see plot 2) we see that the expression of C is decreases quickly while the expression of Cox increases at the same rate. This is reasonable given the systems set up.

When increasing the degradation rate of C (see plot 3) the expression of C increases and of Cox decreases at the same rate. This could be due to the concentration of C being negative and therefore mathematically making the affect of D_C positive. Another thing worth noting if that the Cox protein has 2 steady states until $D_C \sim 7$. This can be seen in the stability analysis, where we can notice a spike toward zero but not reaching zero, and thus there is only one steady state.

When increasing the dimerization rate of the C protein (see plot 4), we notice a similar behavior to that of increasing the C proteins degradation. This must also be due to the C protein being negative.

When increasing the strength of the Pe promoter (see plot 5) we note an increase in the expression of Cox and no change in the TetR and C protein expression. The graphs differ slightly in plotting the C protein, the bifurcation diagram is missing a line for the initial bump present in the time series and stability analysis. However, the lines that are present corresponds well and the graph for the expression of C is the stable steady state. The results given in plot 5 are reasonable.

When increasing the strength of the pBAD promoter (see plot 6) we can note that the concentration of TetR and Cox increases while that of C decreases drastically. This is reasonable given the connections between the proteins. All of the graphs correspond well with each other.

When increasing the degradation of the Cox protein (see plot 7) we can note that the expression of the C and TerR protein remains unchanged while that of Cox decreases. This is reasonable given that none of the proteins are affected by the concentration of Cox, except Cox. The graphs correspond well with each other.

When increasing the inhibition rate of the C_2 dimer on the Pe promoter (see plot 8) we can note that only the expression of the Cox protein is affected and increases with the parameter. The reason for this could be that since the concentration of C is negative, that makes the concentration of C_2 negative, and therefore increasing I_{C2} only positively affects the expression of Cox.

When increasing the tetramerization rate of Cox (see plot 9) we notice, again, that only the expression of Cox is affected, it decreases. This is reasonable given that the tetramerization affect the concentration of Cox and the resulting tetramer inhibits the production of Cox. The tetramer should have no affect in the other proteins.

When increasing the strength of the inhibition done on the Pe promoter by the Cox tetramer (see plot 10) we can again note that only Cox is affected, and negatively so. This is reasonable given that the inhibition done on the expression of Cox is greater and since its done on the Pe promoter, none of the other proteins should be affected.

When increasing the strength of the inducement done by arabinose (see plot 11) we see an increase in the expression of Cox and TetR and a decrease of that in C. This is reasonable given the relationship between the proteins and arabinose. Arabinose induces the pBAD promoter which promotes the expression of TetR and Cox, more TetR leads to an increased suppression of the C protein, and less C protein leads to a weakened suppression of the Pe promoter, which expresses more Cox.

When increasing the degradation rate of TetR (see plot 12) we notice a decrease in the expression of TetR and Cox and an increase of that in C. This is reasonable given the relationship between the proteins.

When increasing the concentration of arabinose (see plot 13) we notice the same behavior to when the strength of induction was increased; the Cox and TetR expression is increased and the expression of C is decreased. This is due to the same reasons, the relationship between the proteins and arabinose.

When reviewing the result of our stochastic model (see plot 14), we can note that there is very little noise, which isn't realistic but may be true. When we fit the model to real data we may notice more noise.

When reviewing the Simbiology results we can also plot the concentrations of other molecules, this was done when modeling the plasmid A time series. The plasmid A time series (see plot 15a) shows us some reasonable behavior. When the C protein concentration decreases the C_2 dimer increases, which is appropriate. The total concentration of C decreases as well as the concentration of C_2 so the Cox protein grows and so does the Cox_4 tetramer. This is strongly due to the parameter values set in the program.

The plasmid B time series (see plot 15b) we notice that TetR and Cox continues to increase which the C protein decreases into a steady state of ~0.5. This makes sense as the TetR protein inhibits the C proteins production, thus allowing Cox to grow.

The plasmid AB time series (see plot 15c) displays the fact that the Cox protein is being expressed by two promoters and thus grows quickly, while the other two are promoted by a single promoter and grows slowly. We can note that the C protein expression is slowly decreasing and the TetR protein is slowly increasing.

In reviewing the sensitivity plots in 16a through 18c we can assess the most powerful parameters for each of the proteins. In the tables below the parameters are ranked depending on their affect in the given protein, 1 being the strongest. The strength of the parameter is dependent on its value when plotting the sensitivity plots, so this comparison of the parameters ma not hold when the model is fitted to true data.

Table 1 shows us the reverse rate of a reaction is often times more important than the forward moving rates. The parameters that come up are reasonable due to the set up of our transcriptional system. What isn't an obvious parameter is the I_C parameter which is the fourth most impactful parameter in the plasmids it has an effect in. The reason this parameter has such an impact is due to its effect on the dimerization reaction; when C_2 is bound to the Pe promoter, the concentration of the dimer decreases, this pushes the dimerization reaction in favor of producing more C_2 which decreases the total concentration of C. If we hadn't run this sensitivity analysis we wouldn't have noticed the strong effect indirect parameters may have.

Table 1: The results of the sensitivity analysis for the C protein, see figure 16a, 17a and 18a.

C protein			
Parameter/Plasmid	A	B	AB
O_{pTet} (reversal rate of pTet binding to RNA polymerase)	1	1	1
D_C (the degradation rate of C)	2	2	3
S_C (the rate that the C_2 dimer splits into to momomers)	3	3	2
I_C (the inhibition done by C_2 n the Pe promoter)	4		4
M_C (the dimerization rate of the C protein)	5	4	5
R_C (reversal rate of C_2 binding to the Pe promoter)	6		6
P_{pTet} (the rate at which pTet binds to RNA polymerase)	7	5	7
k_{pTet} (the rate at which pTet and RNA pol. produce mRNA)	8	6	8
k (the translation rate of mRNA)	9	7	9
D_{RNA} (the degradation rate of mRNA)	10	8	10

Table 2 shows us that the inhibition by C_2 has no effect on the expression of Cox, which is interesting as it should do so. This marks a potential flaw of the system and explains why the C protein is always in a much lower concentration then that of Cox – the C protein isn't suppressing the Cox protein. This may be how the actual P2 switch works but its doubtful. Otherwise all of the parameters have a reasonable effect on the expression of Cox.

Table 2: The results of the sensitivity analysis for the C protein, see figure 16b, 17b and 18b.

Cox protein			
Parameter/Plasmid	A	B	AB
T_{Cox} (tetramerization rate of Cox)	1	1	1
P_{Pe} (the rate at which Pe binds to RNA polymerase)	2		2
O_{Pe} (reversal rate of Pe binding to RNA polymerase)	3		3
D_{Cox} (the degradation rate of Cox)	4	3	4
P_{pBAD} (the rate at which pBAD binds to RNA polymerase)		2	5
S_{Cox} (the rate that the Cox_4 tetramer splits into to momomers)	8	8	6
Ara (the concentration of arabinose)		4	7
k_{Pe} (the rate at which Pe and RNA pol. produce mRNA)	5		8
k (Pe) (the translation rate of mRNA, done by Pe)	6		9
D_{RNA} (Pe) (the degradation rate of mRNA, done by Pe)	7		10
O_{pBAD} (reversal rate of pBAD binding to RNA polymerase)		5	11
k_{pBAD} (the rate at which pBAD and RNA pol. produce mRNA)		6	12
k (pBAD) (the translation rate of mRNA, done by pBAD)		7	13
D_{RNA} (pBAD) (the degradation rate of mRNA, done by pBAD)		9	14
I_{Cox} (the rate at which Cox_4 binds to Pe)	9		15
R_{Cox} (the reversal rate at which Cox_4 binds to Pe)	10		16

Table 3 shows us that none of the other proteins affect the expression of TetR, which is how the desired system is designed.

Table 3: The results of the sensitivity analysis for the C protein, see figure 17c and 18c.

TetR protein			
Parameter/Plasmid	A	B	AB
P_{pBAD} (the rate at which pBAD binds to RNA polymerase)	-	1	1
D_{TetR} (the degradation rate of TetR)	-	2	2
Ara (the concentration of arabinose)	-	3	3
O_{pBAD} (reversal rate of pBAD binding to RNA polymerase)	-	4	4
k_{pBAD} (the rate at which pBAD and RNA pol. produce mRNA)	-	5	5
k (the translation rate of mRNA)	-	6	6
D_{RNA} (the tdegradation rate of mRNA)	-	7	7
I_T (the rate at which TetR binds to the pTet operator)	-	8	8
R_T (the reversal rate at which TetR binds to pTet)	-	9	9

All in all, we did not notice any bistability, only impulses. This doesn’t bode well for the switch we are trying to create and an enhancement of our current system is needed.

Future work

In the future we hope to fit our models to the data from wet lab, and therefore we have studied different ways of doing so.

We need to be aware that any data we fit the model to will yield an estimated parameter value, due to data being noisy and error prone. One needs to fit the model to several data sets and the average of all the estimated parameter values will be the true parameter value - but often times you only have one data set. Bootstrapping overcomes the issue by resampling the data several times so we have additional parameter estimates and therefore find the true. Another method to overcome the lack of data is crossvalidation, where one part of the dataset the training set to which the model is fitted to and the other is the test set which evaluates the prediction error. Repeating this for different parts of the data set till judge how well the model works. A third method for parameter estimation is the Bayesian parameter estimation which is based on the parameter set not being a fixed value but rather a random variable so, by choosing its probability distribution we can state which parameter sets we regard in advance.[12]

By using these methods we should be more than able to fit our model and estimate our parameters accurately.

References

1. O'Neill, J. (2014).
2. World, Health & Organization (2018).
3. Doss, J., Culbertson, K., Hahn, D., Camacho, J. & Barekzi, N. A Review of Phage Therapy against Bacterial Pathogens of Aquatic and Terrestrial Organisms. Viruses 9, 50 (2017).
4. Nilsson, A.S. Phage therapy–constraints and possibilities. Upsala journal of medical sciences 119, 192-198 (2014).
5. Nilsson, A. (ed. E. Black) (iGEM Stockholm, 2019).
6. Renberg-Eriksson, S.K., Ahlgren-Berg, A., DeGrooth, J. & Haggård-Ljungquist, E. Characterization of the Developmental Switch Region of Bacteriophage P2 Hy dis. Virology 290, 199-210 (2001).
7. Maarleveld, T., Olivier, B. & Bruggeman, F. (2014).
8. Cao, Y., Gillespie, D.T. & Petzold, L.R. Efficient step size selection for the tau-leaping simulation method. The Journal of Chemical Physics 124, 044109 (2006).
9. MathWorks (2019).
10. Massad, T. et al. Crystal structure of the P2 C-repressor: a binder of non-palindromic direct DNA repeats. Nucleic Acids Res 38, 7778-7790 (2010).
11. van Dijk, M. & Bonvin, A.M.J.J. 3D-DART: a DNA structure modelling server. Nucleic Acids Research 37, W235-W239 (2009).
12. Klipp , E., Liebermeister , W., Wierling , C. & Kowald, A. Systems Biology: A Textbook, Edn. 2. (Wiley-Blackwell, USA; 2016).

GitHub link: https://github.com/AniKTH/iGEM_2019