Warwick iGEM Model

December 14, 2017

1 What Actually is a Mathematical Model?

Before a proper introduction, modeling can seem like a very abstract concept with no apparent connection to lab work. It can quite easily come across as some gratuitous 'extra science' with little to no real world consequences. However, used correctly it can be extremely useful for predicting the behavior of a biological system, or optimizing such a system. This benefit allows for potentially massive amounts of time saved in a lab; if the modeling highlights a mechanism that will fail, this can be anticipated and edited before the lengthy process of ordering parts, PCR, gene insertion, growing the culture etc. Even once the biological mechanism is functioning in the lab, a sturdy model allows for rapid iteration and optimization that would not be possible given the time constraints involved with wet lab work. Used properly, a model is an additional tool in conjunction with the lab to add functionality to the design process, saving time and resources.

1.1 Simplicity is Key

I still have not actually described what a model is though. The key to approaching it is to keep in mind that a mathematical model is a fundamentally basic concept, but one may seem initially daunting if it is built up of lots of different parts to form something more complex. If its broken down and examined in the right way, each piece of a model will be simple, and will come together to perform one overall function. Much like the modular nature of synthetic biology itself.

Modeling follows the principle that the behavior of physical system can be described using mathematical equations. The system will be made up of variables, and these variables will have different mathematical relationships with each other. One variable might be a function of a number of other variables, etc.

1.2 Falling Ball Example

A simple example of mathematical model of a system could be a ball, falling from a height h, under the force of gravity g. For a simplistic model, the ball is governed by its position in space, y, its velocity, v, its acceleration, a, and the time elapsed from an initial event, t. We are then able to construct a set of equations to show how the properties of the system vary with time. We know that each second, its position in space y changes by its velocity v. We also know that with each second, its velocity v, changes by its acceleration, a. These equations would be as follows:

$$\frac{dy}{dt} = v \tag{1}$$

$$\frac{dv}{dt} = a \tag{2}$$

These two basic equations, now govern the relationships between the key variables identified for the behavior of the falling ball system. (This will all become relevant shortly!). We have outlined our time dependent relationships for the system, now we just need to establish some system parameters. Parameters are a fairly broad term when it comes to modeling. The best way to consider them is inputs to the system that do not vary with time or other time dependent variables. For our falling ball example, our input parameters will be the gravitational constant, g, and the initial height from which the ball is dropped, h. We will program into our model that at $t = 0$, $y = h$; also that $a = g$. This simulates the ball being released at the start of the simulation from the height h, and will have a constant acceleration of g.

So where am I going with all of this? Well what we have outlined here is all the basic components for a mathematical model. Once you have identified the variables of your system, the mathematical relationships that connect those variables and the input parameters for that system, you can then write these into a script form for a programming platform such as MATLAB. I'm not going to go into the details of how to write that script here, but learning code syntax should not be considered a significant barrier if this is something you want to explore. Once the model has been built (the model is not a physical thing, just a system of equations to approximate a system), the differential equations need to be solved. The reason we do computational modeling is that for more complex systems, solving the differential equations by hand is not a trivial task (where in the case of our falling ball example it would be). Platforms such as MATLAB often come with built in solvers for systems of differential equations that will process them using numerical approximations. All you do is create your system, send it to the solver and then the solver will return a solution. Whilst a lot of this may seem abstract, such as 'send it to the solver', it really is that simple, and understanding that you create the mathematical description for the system, then use built in functions to process this information is sufficient to have a good idea of what is going on.

Now we have a basic model for a falling ball that can predict the position, velocity and acceleration of the ball at any point in time. Testing this against a real ball falling from a set distance, we might find some differences between the model and reality. From there we can develop the sophistication of the model to improve its ability to predict the ball's behavior. Perhaps we could add equations to take air resistance into account, or add x and z positional components; maybe rotational components too. You would add all of these additional relationships in the same way we started with the governing principles of the system. The point is that it is just a matter of simple steps to build up a sophisticated model.

1.3 How does this Relate to Biology?

Well a biological system is just the same as a mechanical system in many respects. It has a number of variables, such as reactant and product concentrations, it has parameters, such as reaction rates, and it has governing equations, such as reaction kinetics. The issue is, in comparison to the falling ball situation, a biological system is incredibly complex. In the context of iGEM, engineers, computer scientists and biologists must work together to decide what level of system representation (how many governing equations are needed) is sufficient to accurately predict the behavior without becoming over complex. Often the system can be massively simplified and still yield surprisingly accurate results.

2 Our Model

2.1 Switching Mechanism

Our model focused on a principal called 'Cellular Economics'. Cellular Economics places attention on the fact that the cell has a finite amount of resources which it uses to carry out host functions. These resources are quantities such as energy, metabolites, ribosomes etc. When synthetic genes are inserted into the cell genome, some of the finite resources will be spent on the synthetic genes and will impede the performance of the cell. For example, if the cell has to use ribosomes to produce an inserted gene, it will not be able to use those ribosomes at that point to produce its metabolic enzymes and therefore the cell has less available energy. As you can see there is a knock on effect and sometimes a positive feedback effect can be established when adding foreign genes to the genome. Our project aimed to insert a light activated mechanism into E.Coli to stimulate the production of cellulose. The dynamic model for the cell was adapted from WEIßE. Y. et al. *Mechanistic Links between cellular trade-offs, gene expression, and growth.* PNAS, 2014. Using this dynamic model as a template, we added the relevant genes for our light activated circuit. From there we were able to test the model to see if there were any obvious failure mechanisms for our circuit and analyze the performance of the light switching mechanism in more detail. From the figure below, it is clear that at 5000 seconds (where we simulated light activation) the cell responds dramatically. Most importantly out of all these changes the cell produces gn, which is the notation for cellulose in our model. It is clear from these results that our theoretical cell is very responsive to the light activation an functions as intended.

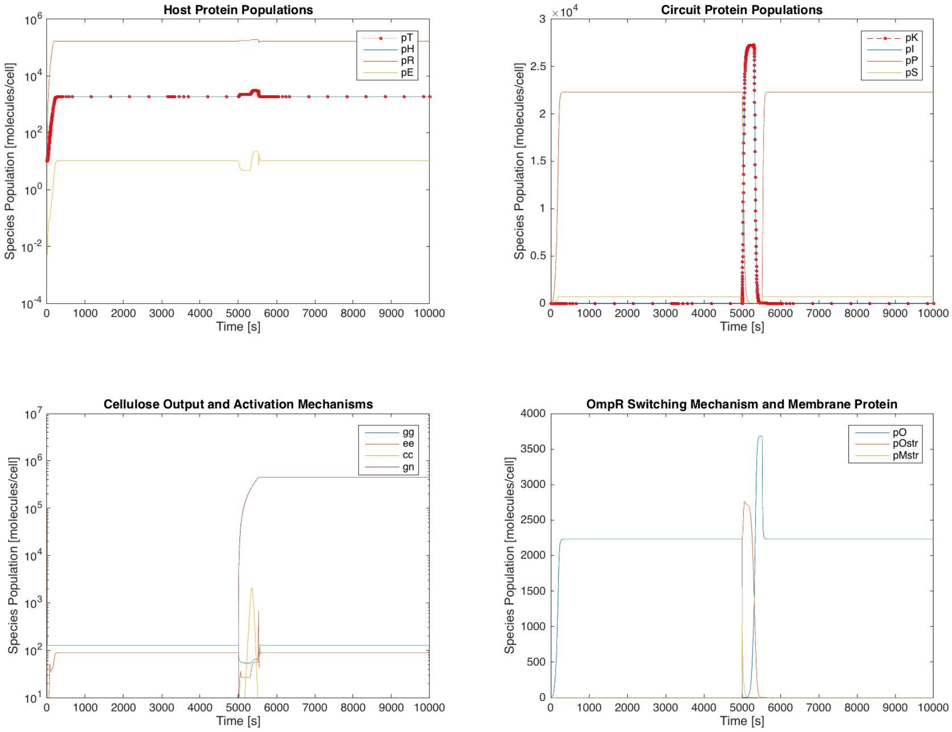

Figure 1: Graphs to show the behaviour of the cell when light activated.

2.2 Multi Objective Genetic Algorithm

The next step in our modelling process was to implement a multi objective genetic algorithm as outlined in Deb, Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*to create a set of 'host mindful' design parameters for our genetic circuit. These parameters are the ribosome binding site strengths of our inserted genes and their respective transcription rates. The model demonstrates a trade off between cellulose production and the cell's growth rate; the multi objective genetic algorithm optimises this trade off and allows us to select an operating point that suits the design needs of the system. The way this works is that the genetic algorithm creates a population of individual cells with unique parameter values. It simulates the cells performances and ranks them on their growth rate and cellulose production (the two objective functions). It then iterates the populations with this in mind varying the parameter sets; this produces an output known as a Pareto front, an optimised graph showing the trade off between our two objective functions (shown in the figure below).

3 Pareto Front

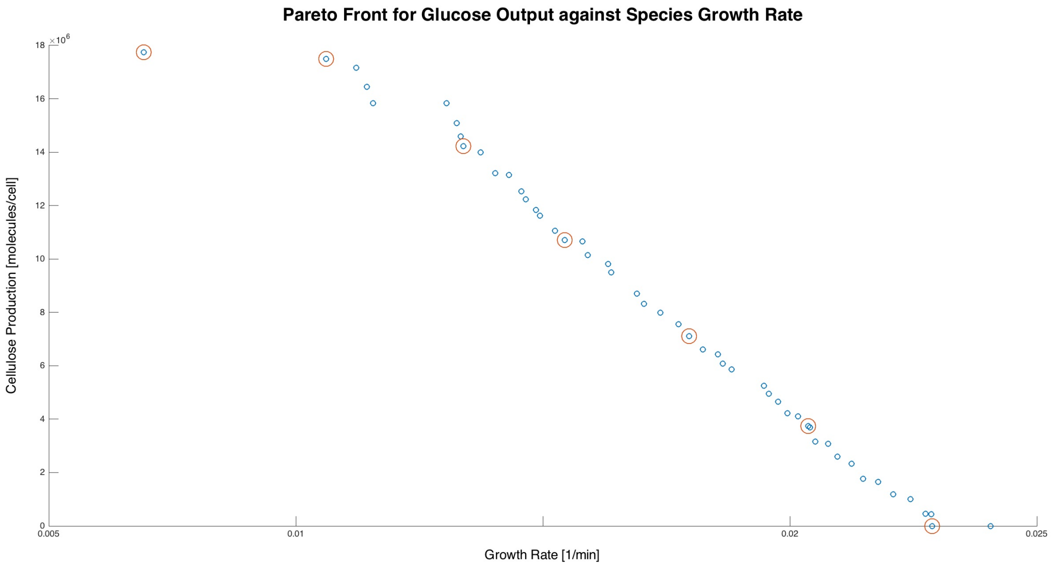

Figure 2: A graph showing the optimised trade off of the two objective functions.

Using the Genetic Algorithm to perform a multi objective optimisation, we were able to produce a figure showing the cellulose output to growth rate trade off for different operational parameters. From this figure we selected seven individual operational points, as shown on the graph, that could be used to ensure the cell's performance suited its application. This figure is an interesting result as there is a shelf where growth rate can be improved without affecting the cellulose output. This is a feature that the modeling process has been able to identify and could be exploited in the future.

4 Operation Points

Table 2.1 shows the operation points for the cell and the corresponding parameter sets for transcription rate and ribosome binding strengths of the inserted gene circuit. For lab implementation, the desired trade off point between cell growth rate and cellulose output can be selected from the table and then required parameters can be read off.

Table 1: Model Operation Parameters

Operating Point	wM	wO	wK	wP	wI	wS	bM	bO	bK	bP	bI	bS	λ	Cellulose Output
1	11.56	5.41	10.84	5.65	80.66	121.94	0.42	0.68	0.36	0.56	0.25	0.51	0.0069	1.77e+07
2	16.71	5.17	16.94	4.83	80.34	82.78	0.41	0.62	0.42	0.54	0.45	0.65	0.011	1.75e+07
3	4.91	3.08	62.78	1.44	179.45	32.39	0.45	0.92	0.78	0.25	0.91	0.93	0.013	1.42e+07
4	8.32	3.45	56.31	2.37	214.21	19.32	0.51	0.72	0.78	0.23	0.82	0.91	0.015	1.07e+07
5	4.49	2.73	29.14	1.98	124.71	14.19	0.14	0.78	0.59	0.42	0.89	0.79	0.017	7.11e+06
6	3.16	2.26	19.56	1.71	29.42	7.59	0.18	0.64	0.68	0.36	0.89	0.61	0.020	3.73e+06
7	2.76	1.00	2.69	1.00	1.00	47.40	0.00	0.00	0.65	0.45	1.00	0.015	0.023	1.65e-08

5 Appendix - Mathematical Description

For this model outline we will define all the species present in the model, but some reactions will be omitted as they are included in the model referenced in the description.

5.1 Species Definitions

gg = Glucose
ee = Energy
cc = Cyclic-di-GMP
gn = Cellulose
m_T = Glucose Importer mRNA
c_T = G-Importer Ribosome Complex
p_T = Glucose Importer Protein
m_E = Metabolism Reaction mRNA
c_E = Metabolism Ribosome Complex
p_E = Metabolism Reaction Protein
m_H = Host Protein mRNA
c_H = Host Protein Ribosome Complex
p_H = Host Protein
m_R = Ribosome mRNA
c_R = Ribosome Ribosome Complex
p_R = Ribosome
m_M = Membrane Protein mRNA
c_M = Membrane Protein Ribosome Complex
p_M = Membrane Protein
p_{M^*} = Activated Membrane Protein
m_O = ompR mRNA
c_O = ompR Ribosome Complex
p_O = ompR Protein
p_{O^*} = Activated ompR
m_K = C-di-GMP Producer mRNA
c_K = C-di-GMP Producer Ribosome Complex
p_K = C-di-GMP Producer Protein
m_P = C-di-GMP Decayer mRNA
c_P = C-di-GMP Decayer Ribosome Complex
p_P = C-di-GMP Decayer Protein
m_I = tetR Protein mRNA
c_I = tetR Protein Ribosome Complex
p_I = tetR Protein
m_S = Cellulose Machinery mRNA
c_S = Cellulose Machinery Ribosome Complex
p_S = Cellulose Machinery Protein
p_{S^*} = Activated Cellulose Machinery

5.2 Host Parameter Definitions

v_T = Glucose Import Rate
k_T = Michaelis Menton Constant for G-Importer Protein
v_E = Rate of Catalysis for Metabolic Protein

k_E = Michaelis Menton Constant for Metabolic Protein
w_X = Default maximum transcription rate
w_H = Host Protein maximum transcription rate
w_R = Ribosome maximum transcription rate
o_X = Default transcription threshold energy for half maximal rate
o_R = Ribosome transcription threshold energy for half maximal rate
dym_X = Default decay rate
b_X = Default RBS strength
u_X = Default ribosome unbinding rate
n_X = Default protein length
n_R = Ribosome protein length
max_G = Maximal elongation length
k_G = Michaelis Menton constant for cellulose production
M_0 = Cell mass
k_H = Host Protein Hill function constant
h_H = Host Protein Hill function constant

5.3 Circuit Parameter Definitions

w_M = Membrane Protein transcription rate
w_O = ompR Protein transcription rate
w_K = C-di-GMP Producer transcription rate
w_P = C-di-GMP Decayer transcription rate
w_I = tetR protein transcription rate
w_S = Cellulose machinery transcription rate
b_M = Membrane protein RBS strength
b_O = ompR protein RBS strength
b_K = C-di-GMP producer protein RBS strength
b_P = C-di-GMP decayer protein RBS strength
b_I = tetR protein RBS strength
b_S = Cellulose machinery RBS strength
k_O = ompR Hill function constant
h_O = ompR Hill function constant
k_I = tetR Hill function constant
h_I = tetR Hill function constant
v_K = C-di-GMP producer enzymatic rate
k_K = C-di-GMP producer Michaelis Menton constant
v_P = C-di-GMP decayer enzymatic rate
k_P = C-di-GMP decayer Michaelis Menton constant
s_S = Cellulose enzymatic parameter
v_S = Cellulose enzymatic parameter
k_S = Cellulose enzymatic parameter
f_s = cc to pS binding rate
r_s = p_{S*}-cc unbinding rate
k_{M^*} = Membrane protein activation reverse reaction constant

kr_O = ompR* degradation to ompR reaction constant

5.4 Translation Rate Parameters

$$ribosomes_{translating} = c_T + c_E + c_H + c_R + c_M + c_O + c_K + c_P + c_I + c_S$$

$$\gamma = \frac{max_G \times ee}{k_G + ee}$$

$$\lambda = \frac{1}{M_0} \times \gamma \times ribosomes_{translating}$$

Note: This section neglects transcription.

5.5 Membrane Protein

$$m_M \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_M \xrightleftharpoons[u_X]{b_M} c_M$$

$$c_M \xrightarrow{\frac{\lambda}{n_M}} p_R + p_M + m_M$$

$$c_M \xrightarrow{\lambda} \emptyset$$

$$p_M \xrightarrow{\lambda} \emptyset$$

$$p_{M^*} \xrightarrow{k_{M^*}} p_M$$

$$p_{M^*} \xrightarrow{\lambda} \emptyset$$

5.6 ompR Protein

$$m_O \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_O \xrightleftharpoons[u_X]{b_O} c_O$$

$$c_O \xrightarrow{\frac{\lambda}{n_O}} p_R + p_O + m_O$$

$$c_O \xrightarrow{\lambda} \emptyset$$

$$p_O + p_{M^*} \rightarrow p_{O^*} + p_{M^*}$$

$$p_O \xrightarrow{\lambda} \emptyset$$

$$p_{O^*} \xrightarrow{kr_O} p_O$$

$$p_{O^*} \xrightarrow{\lambda} \emptyset$$

5.7 C-di-GMP Producer

$$m_K \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_K \underset{b_K}{\overset{u_K}{\leftrightharpoons}} c_K$$

$$c_K \xrightarrow{\frac{\gamma}{n_K}} m_K + p_K + p_R$$

$$c_K \xrightarrow{\lambda} \emptyset$$

$$p_K \xrightarrow{\lambda} \emptyset$$

5.8 C-di-GMP Decayer

$$m_P \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_P \underset{b_P}{\overset{u_P}{\leftrightharpoons}} c_P$$

$$c_P \xrightarrow{\frac{\gamma}{n_P}} m_P + p_P + p_R$$

$$c_P \xrightarrow{\lambda} \emptyset$$

$$p_P \xrightarrow{\lambda} \emptyset$$

5.9 tetR Gene Protein - yhjh Gene Inhibitor

$$m_I \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_I \underset{b_I}{\overset{u_I}{\leftrightharpoons}} c_I$$

$$c_I \xrightarrow{\frac{\gamma}{n_I}} m_I + p_I + p_R$$

$$c_I \xrightarrow{\lambda} \emptyset$$

$$p_I \xrightarrow{\lambda} \emptyset$$

5.10 Cellulose Producing Proteins

$$m_S \xrightarrow{\lambda + dym_X} \emptyset$$

$$p_R + m_S \underset{u_S}{\overset{b_S}{\leftrightharpoons}} c_S$$

$$c_S \xrightarrow{\lambda} \emptyset$$

$$c_S \xrightarrow{\frac{\lambda}{n_S}} m_S + p_S + p_R$$

$$p_S \xrightarrow{\lambda} \emptyset$$

$$p_S + c_c \underset{r_S}{\overset{f_S}{\leftrightharpoons}} p_{S^*}$$

$$p_{S^*} \xrightarrow{\lambda} \emptyset$$

5.11 C-Di-GMP

$$ee \xrightarrow{p_K} cc$$

$$cc \xrightarrow{p_P} \emptyset$$

$$cc \xrightarrow{\lambda} \emptyset$$

5.12 Cellulose Production

$$gg \xrightarrow{p_{S^*}} gn$$

5.13 Transcription Rates

$$g2m_M = \frac{w_M \times ee}{o_X + ee}$$

$$g2m_O = \frac{w_O \times ee}{o_X + ee}$$

$$g2m_K = \frac{w_K \times ee}{o_X + ee} \times \frac{(\frac{p_{O^*}}{k_O})^{h_O}}{1 + (\frac{p_{O^*}}{k_O})^{h_O}}$$

$$g2m_P = \frac{w_P \times ee}{o_X + ee} \times \frac{1}{1 + (\frac{p_I}{k_I})^{h_I}}$$

$$g2m_I = \frac{w_I \times ee}{o_X + ee} \times \frac{(\frac{p_{O^*}}{k_O})^{h_O}}{1 + (\frac{p_{O^*}}{k_O})^{h_O}}$$

$$g2m_S = \frac{w_S \times ee}{o_X + ee}$$

5.14 Translation Rates

$$m2p_M = \frac{\gamma}{n_M} \times c_M$$

$$m2p_O = \frac{\gamma}{n_O} \times c_O$$

$$m2p_K = \frac{\gamma}{n_K} \times c_K$$

$$m2p_P = \frac{\gamma}{n_P} \times c_P$$

$$m2p_I = \frac{\gamma}{n_I} \times c_I$$

$$m2p_S = \frac{\gamma}{n_S} \times c_S$$

5.15 Membrane Protein

$$\frac{dm_M}{dt} = g2m_M - (\lambda + dym_X) \times m_M + m2p_M - b_M \times p_R \times m_M + u_X \times c_M$$

$$\frac{dc_M}{dt} = -\lambda \times c_M - m2p_M + b_M \times p_R \times m_M - u_X \times c_M$$

$$\frac{dp_M}{dt} = m2p_M - \lambda \times p_M + k_{M^*} \times p_{M^*}$$

$$\frac{dp_{M^*}}{dt} = -\lambda \times p_{M^*} - k_{M^*} \times p_{M^*}$$

5.16 ompR Protein

$$\frac{dm_O}{dt} = g2m_O - (\lambda + dym_X) \times m_O + m2p_O - b_O \times p_R \times m_O + u_X \times c_O$$

$$\frac{dc_O}{dt} = -\lambda \times c_O - m2p_O + b_P \times p_R \times m_O - u_X \times c_O$$

$$\frac{dp_O}{dt} = m2p_O - \lambda \times p_O - p_O \times p_{M^*} + kr_O \times p_{O^*}$$

$$\frac{dp_{O^*}}{dt} = p_O \times p_{M^*} - kr_O \times p_{O^*} - p_{O^*} \times \lambda$$

5.17 c-di-GMP Producer

$$\frac{dm_K}{dt} = g2m_K - (\lambda + dym_X) \times m_K + m2p_K - b_K \times p_R \times m_K + u_X \times c_K$$

$$\frac{dc_K}{dt} = -\lambda \times c_K - m2p_K + b_K \times p_R \times m_K - u_X \times c_K$$

$$\frac{dp_K}{dt} = m2p_K - \lambda \times p_K$$

5.18 c-di-GMP Decay

$$\frac{dm_P}{dt} = g2m_P - (\lambda + dym_X) \times m_P + m2p_P - b_P \times p_R \times m_P + u_X \times c_P$$

$$\frac{dc_P}{dt} = -\lambda \times c_P - m2p_P + b_P \times p_R \times m_P - u_X \times c_P$$

$$\frac{dp_P}{dt} = m2p_P - \lambda \times p_P$$

5.19 tetR Protein - yhjh Inhibitor

$$\frac{dm_I}{dt} = g2m_I - (\lambda + dym_X) \times m_I + m2p_I - b_I \times p_R \times m_I + u_X \times c_I$$

$$\frac{dc_I}{dt} = -\lambda \times c_I - m2p_I + b_I \times p_R \times m_I - u_X \times c_I$$

$$\frac{dp_I}{dt} = m2p_I - \lambda \times p_I$$

5.20 Cellulose Producing Proteins

$$\frac{dm_S}{dt} = g2m_S - (\lambda + dym_X) \times m_S + m2p_S - b_S \times p_R \times m_S + u_X \times c_S$$

$$\frac{dc_S}{dt} = -\lambda \times c_S - m2p_S + b_S \times p_R \times m_S - u_X \times c_S$$

$$\frac{dp_S}{dt} = m2p_S - \lambda \times p_S - f_S \times cc \times p_S + r_S \times p_{S^*}$$

$$\frac{dp_{S^*}}{dt} = f_S \times cc \times p_S \times -r_S \times p_{S^*} - \lambda \times p_{S^*}$$

5.21 C-Di-GMP

$$\frac{dcc}{dt} = \frac{v_K \times ee \times p_K}{k_K + ee} - \frac{v_P \times cc \times p_P}{k_P + cc} - \lambda \times cc$$

5.22 Cellulose Production

$$\frac{dgn}{dt} = \frac{v_S \times gg \times p_{S^*}}{k_S + p_{S^*}}$$

6 Contributors

Postgraduate Modelling Instructor - Alexander Darlington
Warwick iGEM Modelling Student - Ben Cox