Modelling a Kill Switch for Probiotics

Overview

NUSgem have successfully completed modelling of a kill switch for engineering probiotics. The aim of modelling a kill switch for probiotics is demonstrate how a kill switch for any application (in this case engineered probiotics), can be easily elucidated using our framework and E2 chassis. Our modelling results serve as a guide for experimenters to better understand the genetic circuit as well as offer a useful tool for optimising and debugging the genetic circuit. The following report will outline the methodology and justifications for our Specifications. The report will then discuss and evaluate the insights gained from the modelling results of the Simple Circuit and Complex Circuit.
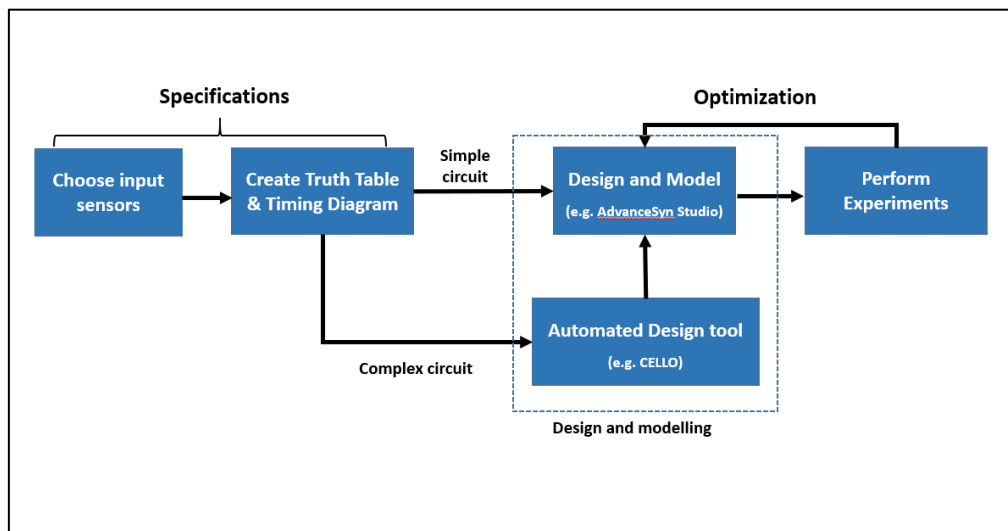

Figure 1 NUS Workflow for engineering a customised kill switch

Specifications Methodology

Choosing Input Sensors

Using our workflow, the first step to develop any kill switch is to specify the input sensors. For the purpose of developing a kill switch for engineered probiotics, we need to choose sensors that are able to differentiate between the environments that an engineered probiotic may be travel through in its real life application. With this aim in mind, we established that in application, the engineered probiotic will enter four distinct stages: 1. Transportation Stage (before consumption; 2. Ingestion Stage; 3. Colon Stage; 4. Wastewater Stage. Again, it is essential that the sensors chosen are able to differentiate between all of these environments to avoid accidental killing, and ensure effective killing. From these specifications, we decided to choose two sensors that when used together, are able to detect all the stated environments. The sensors we chose were a phosphate sensor and a temperature sensor. We decided to choose a phosphate sensor because phosphate concentrations are different in the human body than in wastewater: phosphate concentration in the human intestine (before absorption) ranges from 25-45mg/L while the phosphate concentration in standard wastewater is about 5mg/L [1, 2, 3]. Initially we used the phosphate sensor module, pPhoB, from the NYMU-Taipei iGEM registry [4], and later, we successfully improved upon this sensor to make feasible our kill switch. Similarly, we decided to choose a temperature sensor that was capable of differentiating between 37℃ body temperature and cooler outdoor temperatures. Fortunately, we were able to locate an effective temperature sensor, in the form of the repressible temperature promoter, pTlpA, which is able to detect temperatures above and below 36℃ [4]. Combining these two sensors, we are able to detect all the environments that an

engineered probiotic may enter. The state diagram in Figure 2 illustrates the changing phosphate and temperature conditions across each of the different environments.
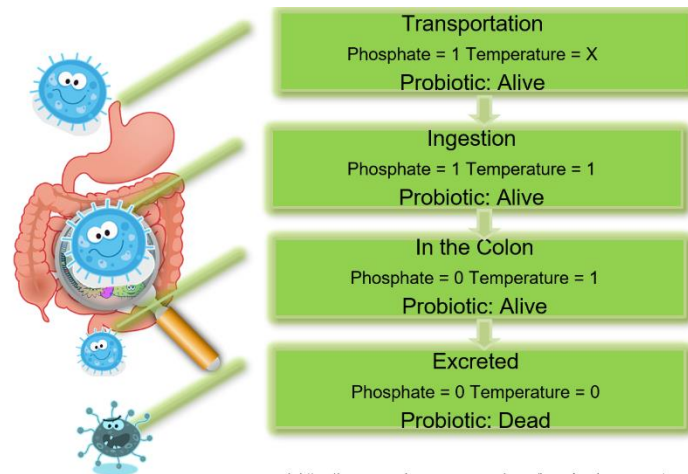
Figure 2 The State Diagram of a kill switch for an engineered probiotic in application. Phosphate = 1 refers to a high concentration of phosphate, Phosphate =0 refers to a low concentration of phosphate. Temperature = 1 refers to greater than 36℃ conditions, Temperature = 0 refers to lower than 36℃ conditions, Temperature = X refers to any temperature condition.

Truth Table and Timing Diagram

Once we have specified our input phosphate and temperature sensors, the next step is to interface these inputs with our output, anti-toxin IM2. The process to interface inputs with outputs is known as logic implementation and requires creating a timing diagram and truth table. To develop the truth table, we first determined the timing diagram. The timing diagram is a graphical representation of the state diagram with the exception that it includes a time domain. After we create the timing diagram, we can segment the timing diagram to each of timing states, or different environments, and record the changing environmental conditions, input promoter states and output anti-toxin expression states. We then organise this information into a table such that the logic implemntation between inputs and output is clear. This relationship between inputs and outputs is called the logic table and it is essential for determining how the kill switch will function in all environments.
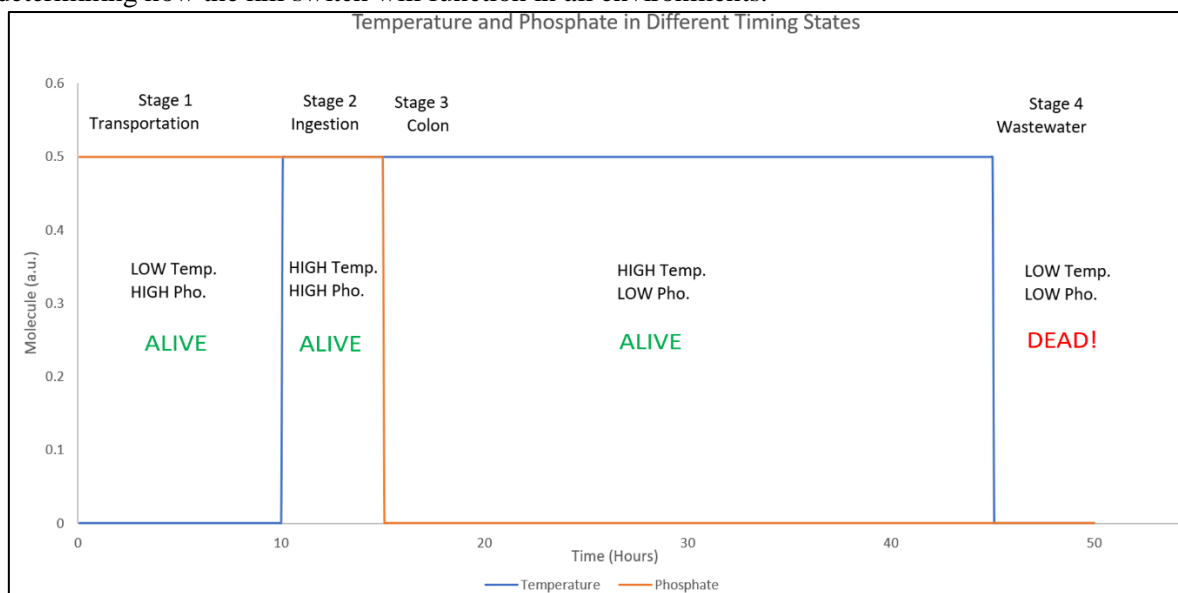

Figure 3 Timing Diagram of the kill switch for engineered probiotics. In the final state we want to stop producing antitoxin IM2 so we can kill the engineered probiotic

Stages	Environmental Conditions (HIGH = 1, LOW = 0)		Simple Circuit Logic Table Inputs (ON = 1, OFF = 0)		Automated Circuit Logic Table Inputs (ON = 1, OFF = 0)		Output (HIGH = 1, LOW, = 0)	Output
	Phosphate	Temperature	pPhoB	pTlpA	pPhoB	pTlpA	IM2	Status
Transportation (0 - 10 Hrs)	1	0/1	0	1	0	0/1	1	Alive
Ingestion (10 – 15 Hrs)	1	1	0	1	0	1	1	Alive
Colon (15 – 45 Hrs)	0	1	1	1	1	1	1	Alive
Wastewater (45 - 50 Hrs)	0	0	1	0	1	0	0	Dead

Figure 4 Logic Table of the kill switch for engineered probiotics. Because of the different circuit designs, the logic table for the Simple Circuit and Automated Circuit slightly varies. However, both circuits obey the timing diagram and logic table and function the same way.

Simple Circuit Design

Using the logic table shown in Figure 4, we generated a cascaded dual input genetic circuit as our simple design. The Simple Circuit Design is shown in Figure 5. In order to model this genetic circuit, we translate it into a dynamic modelling software such MATLAB or AdvanceSyn. For our purpose, we modelled the Simple Circuit in AdvanceSyn. From AdvanceSyn, we completed simulation of the functional model, applied sensitivity, and used combinatorial analysis to analyse three of the most sensitive part kinetics. These simulations are important because from the results we are able to gain better insight about our Simple Circuit and improve and optimise design.

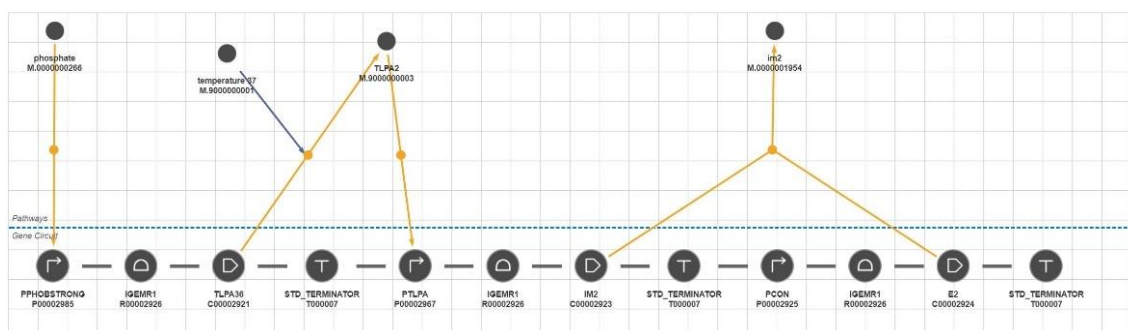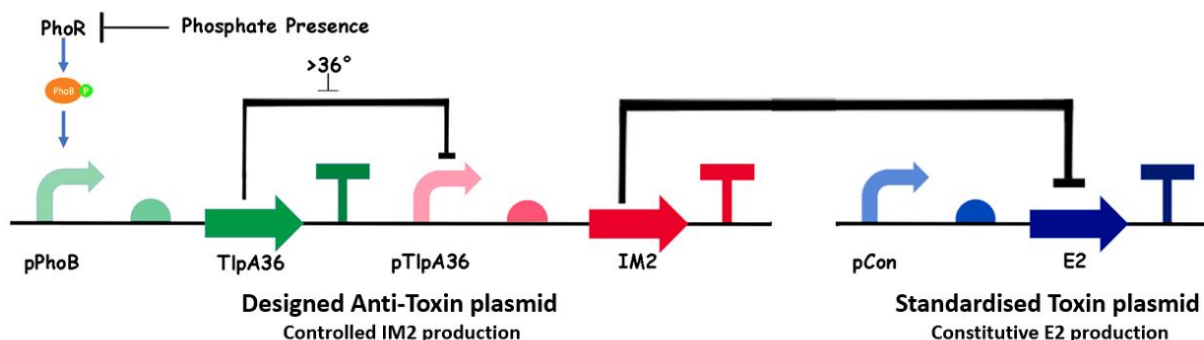All simulations are run for 50 hours at a 300s timestep.

Figure 5 (Above) The genetic circuit for the Simple Circuit Design. (Below) The Simple Circuit as displayed in the AdvanceSyn Studio.
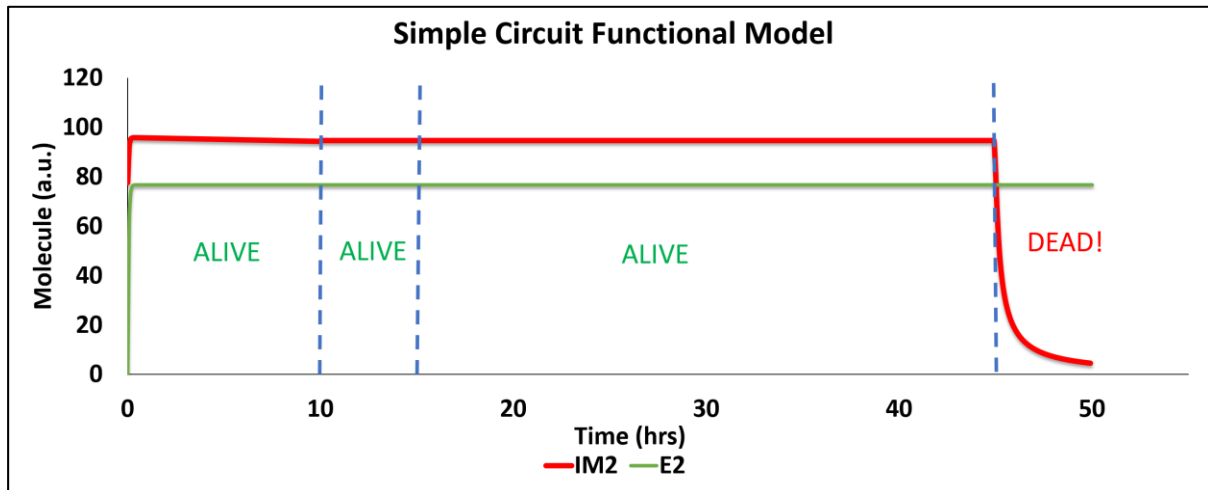
Results & Discussion
Functional Model

Figure 6 The functional model illustrates the ideal performance of the Simple Circuit in all timing states. This serves as a proof of a concept that the Simple Circuit can successfully be used as a kill switch for engineered probiotics!

The functional model for the Simple Circuit was generated by increasing the binding affinity of the temperature promoter and by decreasing the promoter strength of pCon by 20%. The role of the functional model is to have a

Sensitivity Analysis

Table 1 Sensitivity Analysis of most sensitive kinetics in the Simple Circuit. Modifying pTlpA Km and pTlpA Vmax will have the biggest effect on the output response, anti-toxin IM2.

Most Sensitive Parts (in order of Descending Sensitivity)
pTlpA Km
pTlpA Vmax
Initial Condition of TlpA36
pPhoB Vmax

Using the results of the sensitivity analysis, we can pose scenarios and use our model to simulate these scenarios. In addition, we can observe undesired behaviour in our scenario cases and offer solutions and optimisation by changing the most sensitive kinetics. In our analysis of Simple Circuit, we posed two problem cases that were inspired by feedback from experimenters in the lab:

1. **What happens if the repressor TlpA is leaky? How can this be fixed?**
2. **What happens if the toxin E2 is too strong? How can this be fixed?**

Combinatorial Analysis

IM2 across Phosphate Promoter Strengths (Leaky)

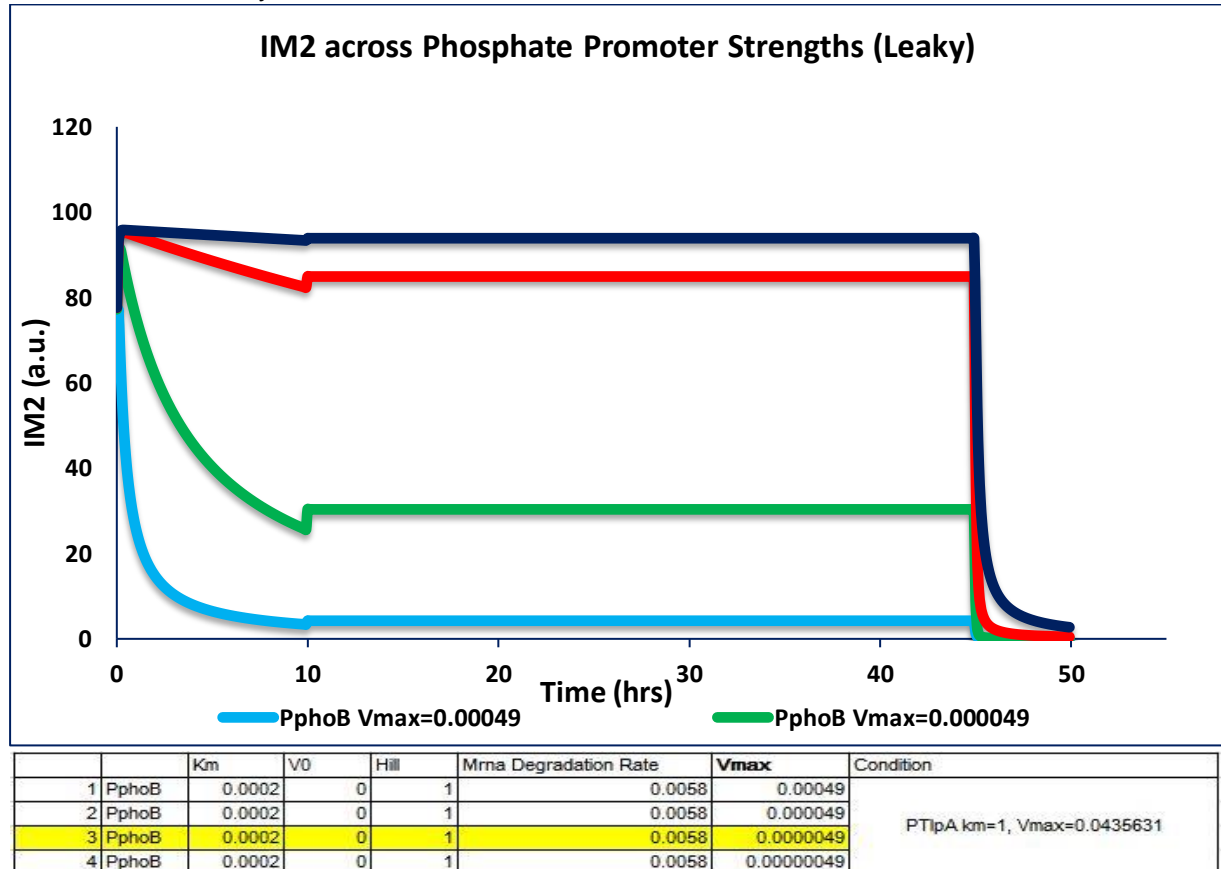|		Km	V0	Hill	Mrna Degradation Rate	Vmax	Condition
1	PphoB	0.0002	0	1	0.0058	0.00049	
2	PphoB	0.0002	0	1	0.0058	0.000049	
3	PphoB	0.0002	0	1	0.0058	0.0000049	PTlpA km=1, Vmax=0.0435631
4	PphoB	0.0002	0	1	0.0058	0.00000049	

Figure 7Decreasing the promoter strength of the phosphate promoter reduces the effect of that leaky TlpA has on the output response, anti-toxin IM2

IM2 across Binding Affinties of Temperature Promoter (Leaky)

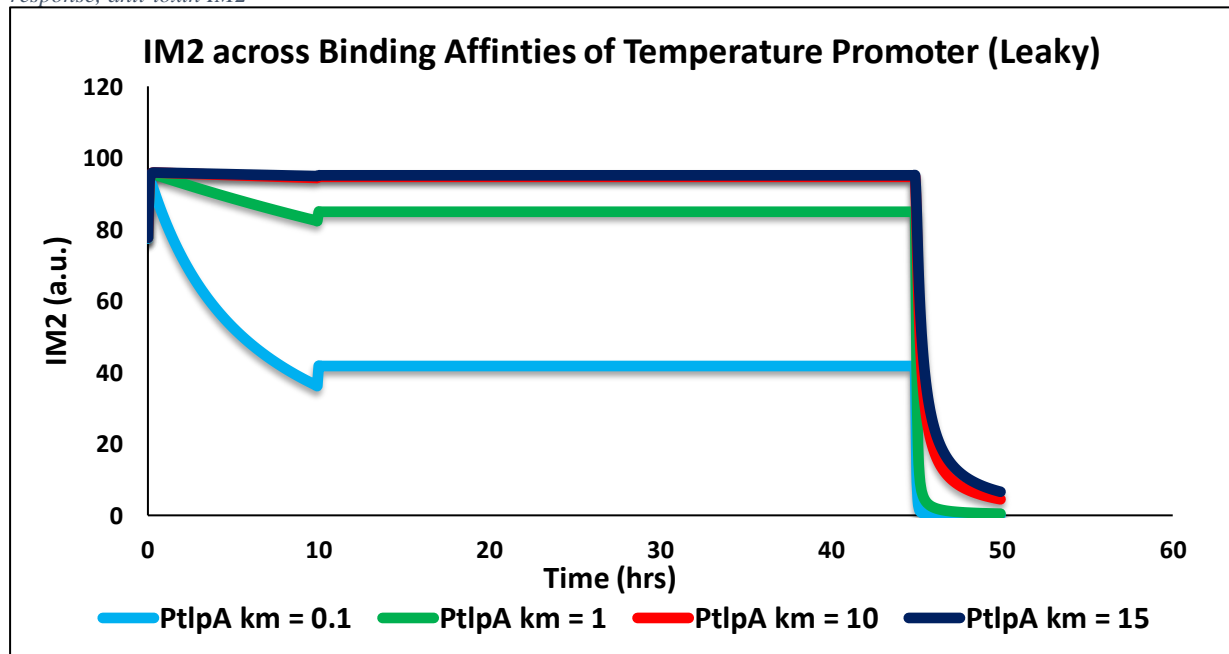|		Km	V0	Hill	Mrna Degradation Rate	Vmax	Condition
1	PTLPA	0.1	0	1	0.0058	0.0435631	
2	PTLPA	1	0	1	0.0058	0.0435631	PphoB km=0.0002, Vmax=0.0000049
3	PTLPA	10	0	1	0.0058	0.0435631	
4	PTLPA	15	0	1	0.0058	0.0435631	

Figure 8Iincreasing the binding affinity of the Temperature sensor also proved an effective counter-measure in reducing the effect of leaky TlpA repressor. However, due to its implementation difficulties, it was not a reasonable solution for fixing TlpA leakiness.

1. What happens if the repressor TlpA is leaky? How can this be fixed?

Using modelling (as shown in Figure 7 and Figure 8), we illustrated that if TlpA repressor is leaky (unwantedly produced), there is a decrease in antitoxin IM2 expression in the initial transportation stage. This unwanted decrease in expression is caused by TlpA repressing the pTlpA temperature promoter which in turn decreases antitoxin IM2 expression. This scenario could be detrimental because it would cause accidental killing of the engineered probiotic.

To solve the issue of leaky TlpA repressor issue, we used modelling to offer two effective solutions. The first solution required decreasing the phosphate promoter strength which was responsible for repressor TlpA expression (Figure 7). The second solution required increasing the binding affinity of the temperature promoter pTlpA (Figure 8). Our models showed that both these solutions were effective counter-measures to reduce the effect of leaky production of TlpA repressor. However, after consulting with our experimenters, we noted that changing the binding affinity of a promoter is a difficult process and therefore, not the most reasonable solution to solve leaky expression. Instead, changing the promoter strength of the phosphate promoter was much easier to implement from an experimental perspective. Therefore, with the help of our models, we successfully proposed an alternative solution to fixing leaky repressor TlpA: decreasing the phosphate promoter strength.

IM2 Production across Different Temperature Promoter Strengths

Y-axis: Molecule (a.u.), X-axis: Time (hrs)

Legend: PtlpA Vmax=0.01, PtlpA Vmax=0.04, PtlpA Vmax=0.1, PtlpA Vmax=0.5, E2

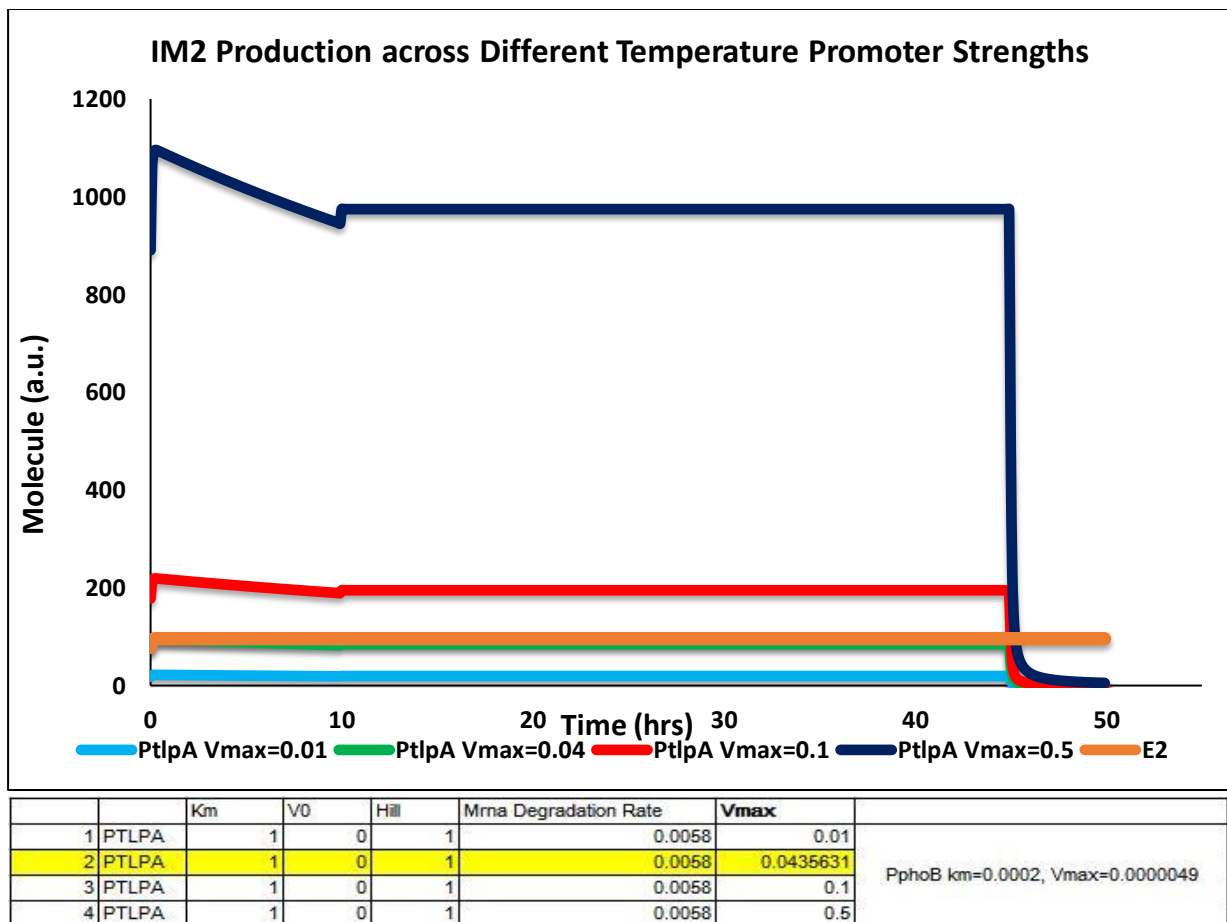|		Km	V0	Hill	Mrna Degradation Rate	Vmax	
1	PTLPA	1	0	1	0.0058	0.01	
2	PTLPA	1	0	1	0.0058	0.0435631	PphoB km=0.0002, Vmax=0.0000049
3	PTLPA	1	0	1	0.0058	0.1	
4	PTLPA	1	0	1	0.0058	0.5	

Figure 9 If E2 toxin is too strong, we recommend increasing the strength of pTlpA. Increasing pTlpA increases IM2 expression. However, if IM2 is produced in excess, this can cause increased metabolic stress for the host.

2. What happens if the toxin E2 is too strong? How can this be fixed?

If the E2 toxin is too strong, the engineered probiotic will die accidentally. This accidental killing of the kill switch can be detrimental because it renders the kill switch ineffective. Therefore to rectify this issue, we first performed a combinatorial analysis of temperature promoter strength. This required simulating the temperature promoter strength across a range of different kinetics. What we found was that the assuming 1:1 binding, the temperature promoter strength should be strong enough such as to produce more antitoxin IM2 than toxinE2. However, producing an excess of antitoxin IM2 is not ideal because it places unnecessary metabolic stress on the engineered probiotic. Likewise a weak temperature strength meant that not enough antitoxin IM2 would be produced and the engineered probiotic would undergo accidental death. Using modelling, we recommended to our lab that if toxin E2 is too strong, increasing the temperature promoter strength accordingly should solve this issue.

Automated Circuit Design

Methodology

To create the genetic circuit, we first translate the logic table proposed earlier to Verilog code. We then enter the Verilog code in the CAD software, Cello, which will then generate genetic circuits that fit the logic table. The genetic circuit generated by Cello is shown in Figure 9. We the then transfer the genetic circuit generated from cello and transfer it to the dynamic modelling software AdvanceSyn. From AdvanceSyn we can generate the functional model, apply sensitivity analysis, apply combinatorial analysis and implement timing states.

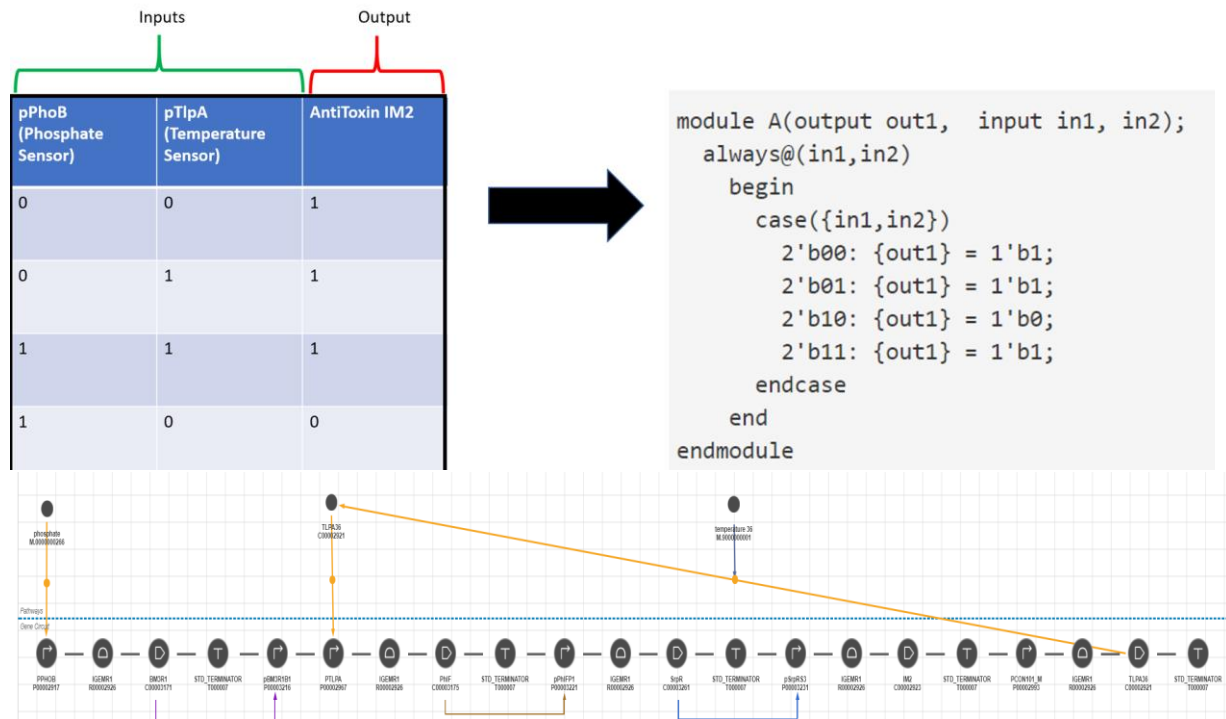All simulations are run for 50 hours at a 300s timestep.

Figure 10 (Above) Converting the logic table to Verilog Code so that Cello can generate a circuit design. (Below) The Automated Circuit generated from Cello as displayed using AdvanceSyn

Results & Discussion
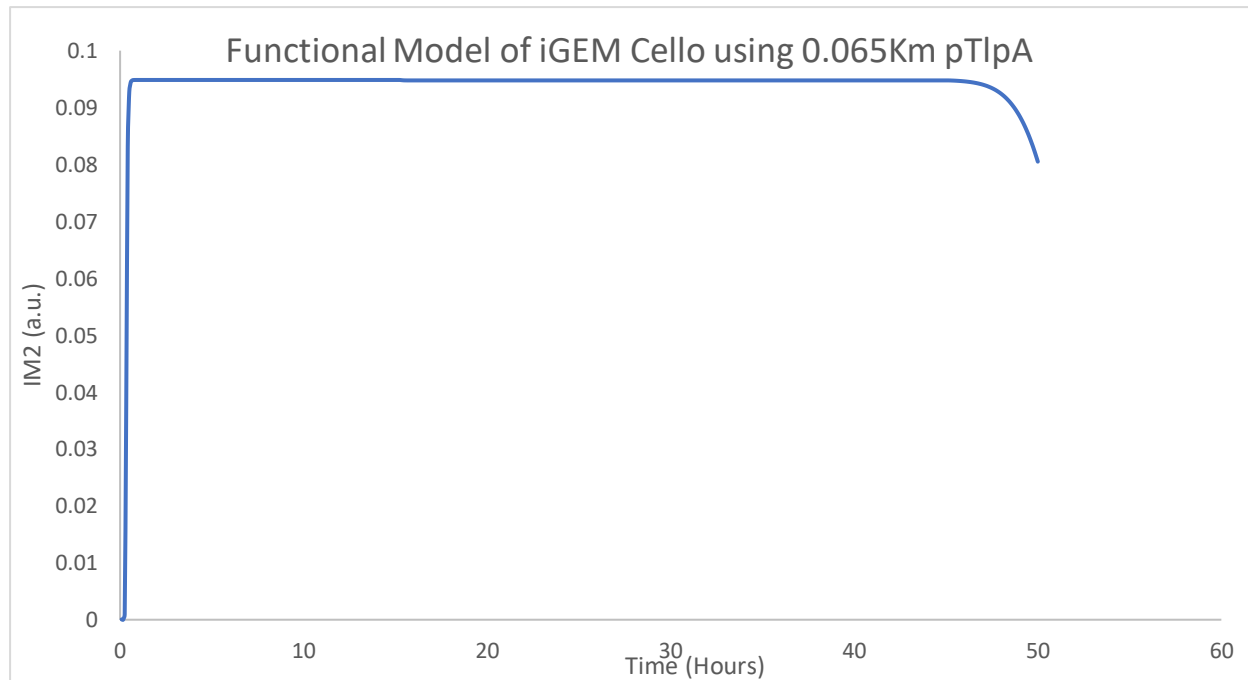### Functional Model

Figure 11 The Functional Model of the Automated Circuit. Note the time taken to repress IM2 in Wastewater (45-50hrs) is slower than in the Simple Circuit.

The functional model generated from Cello and modelled in AdvanceSyn fits the timing diagram most accordingly. The only changes made to generate the functional model required increasing the binding affinity of the temperature promoter. It should be noted that in the Automated Circuit functional model, the repression of antitoxin IM2 is significantly slower compared to the Simple Circuit Functional model. The reason there is a delay in killing is due to propagation delay. In the Automated Circuit, propagation delay occurs because of the complexity of the circuit. With each added part and its respective interaction, the time taken for changes in the environment to be recognised in the output response increases because of the increased processes. In other words, the more complex the genetic circuit is, the longer it takes to generate output. To rectify this issue, it is possible to increase the promoter strength and RBS of the parts in the circuit, however, this comes at the risk of increasing metabolic stress.

Sensitivity Analysis

Most Sensitive Parts (in order of Descending Sensitivity)
pTlpA Km
pTlpA Vmax
pTlpA Hill Coefficient
pPhoB Vmax

Using the results of the sensitivity analysis, we can pose scenarios and use our model to simulate these scenarios. In addition, we can observe undesired behaviour in our scenario cases and offer solutions and optimisation by changing the most sensitive kinetics. In our analysis of Automated Circuit, we propose three scenarios that were inspired by feedback from our sensitivity analysis and our modelling of the Simple Circuit:

1. **What happens if the repressor TlpA is leaky? How can this be fixed?**
2. **What happens if pPhoB is leaky? How can this be fixed?**
3. **What happens if pTlpA leaky? How can this be fixed?**

Combinatorial Analysis

Figure 12 Changing the binding affinity of the sensitive binding affinity of pTlpA kinetic parameter to address leaky TlpA repressor. The ideal binding affinity is one that is high enough to reduce the effect of undesired antitoxin repression in the Colon Stage, yet low enough that can enact fast killing in the final stage.
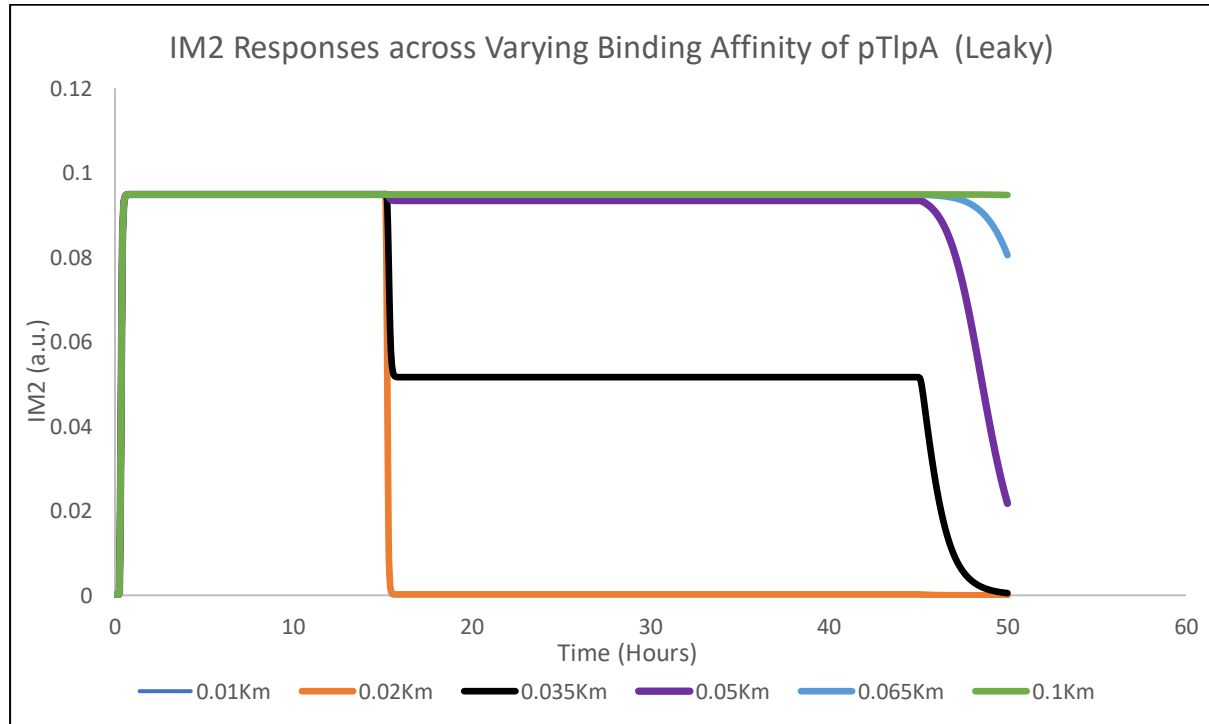
1) What happens if the repressor TlpA is leaky? How can this be fixed?

Leaky TlpA repressor affects the output response only in the Colon and Wastewater stage. This occurs because in the third and fourth stage, the genetic circuit is solely dependent on the promoter. We can observe that at lower binding affinities where the temperature promoter is sensitive to the repressor molecule, antitoxin IM2 is prematurely repressed. Likewise, at high binding affinities, the circuit no longer is sensitive to any environmental change, and continuously produces anti toxin IM2, or represses of antitoxin IM2 at a slower rate. Therefore, for the kill switch to be successful when TlpA repressor is leaky, our models show the Automated Circuit requires a temperature promoter to have a "goldilocks" binding affinity such that the binding affinity is high enough so leaky TlpA repressor doesn't prematurely repress antitoxin IM2 significantly; yet, low enough to generate a fast repression of antitoxin IM2 in the wastewater stage.
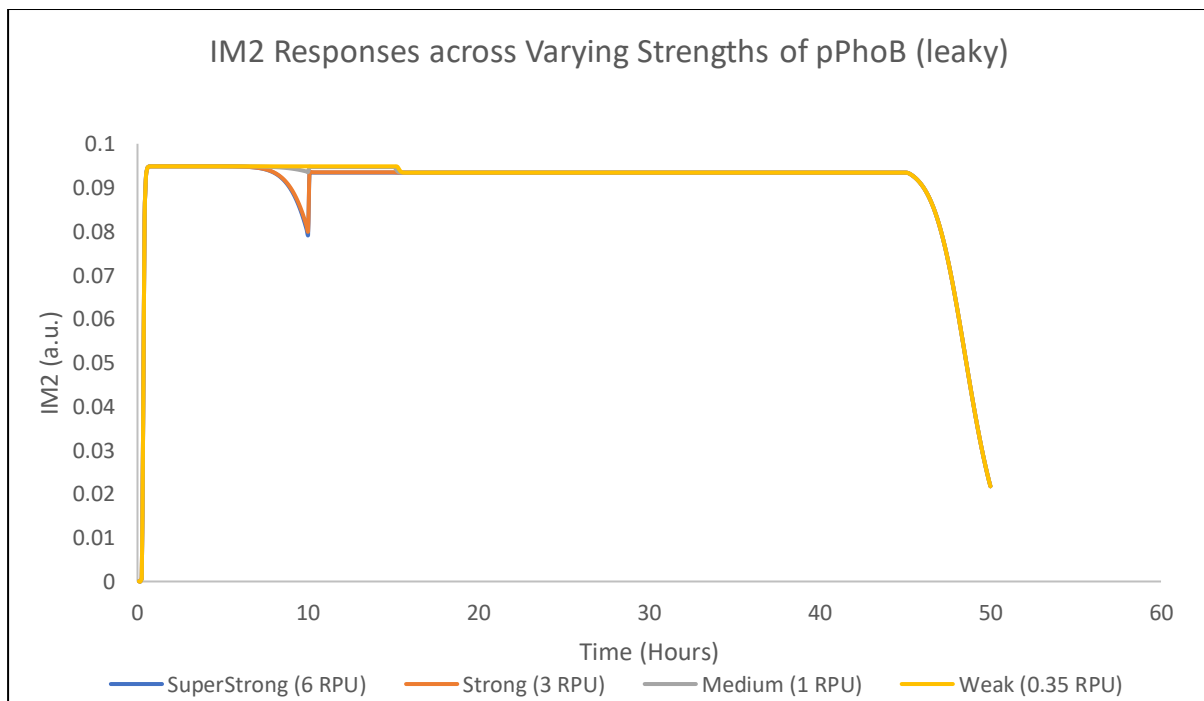
Figure 13 Changing the promoter strength of pPhoB. Changing the promoter strength of pPhoB does not have significant impact on the results due to the intermediate parts acting as buffers and time constraint. However, if the transportation stage is increased, we may see observe increased repression of antitoxin IM2 with higher strengths of pPhoB.

2) What happens if pPhoB is leaky? How can this be fixed?

If the phosphate promoter is leaky, the promoter strength does not significantly impact the output response. Phosphate promoter strength is most important in Transportation Stage and Ingestion Stage. Our model indicates that at a high phosphate promoter strength, there is a slight decrease in antitoxin IM2 expression. This is because the phosphate promoter is responsible for the repression of antitoxin IM2 and when leaky, there is accidental repression of antitoxin IM2. The reason antitoxin IM2 expression slightly decreases is because of delay caused by the time constraint of the transportation stage and the complexity of the Automated Circuit. These two factors combine limit the effect of accidental repression of antitoxin IM2 caused by a leaky phosphate promoter. However, if we increase the Transportation Stage to a longer duration, the effect of accidental repression of antitoxin may be greater. Therefore, we suggest in the event of a leaky phosphate promoter, a user can decrease phosphate promoter strength to mitigate accidental repression of antitoxin IM2.
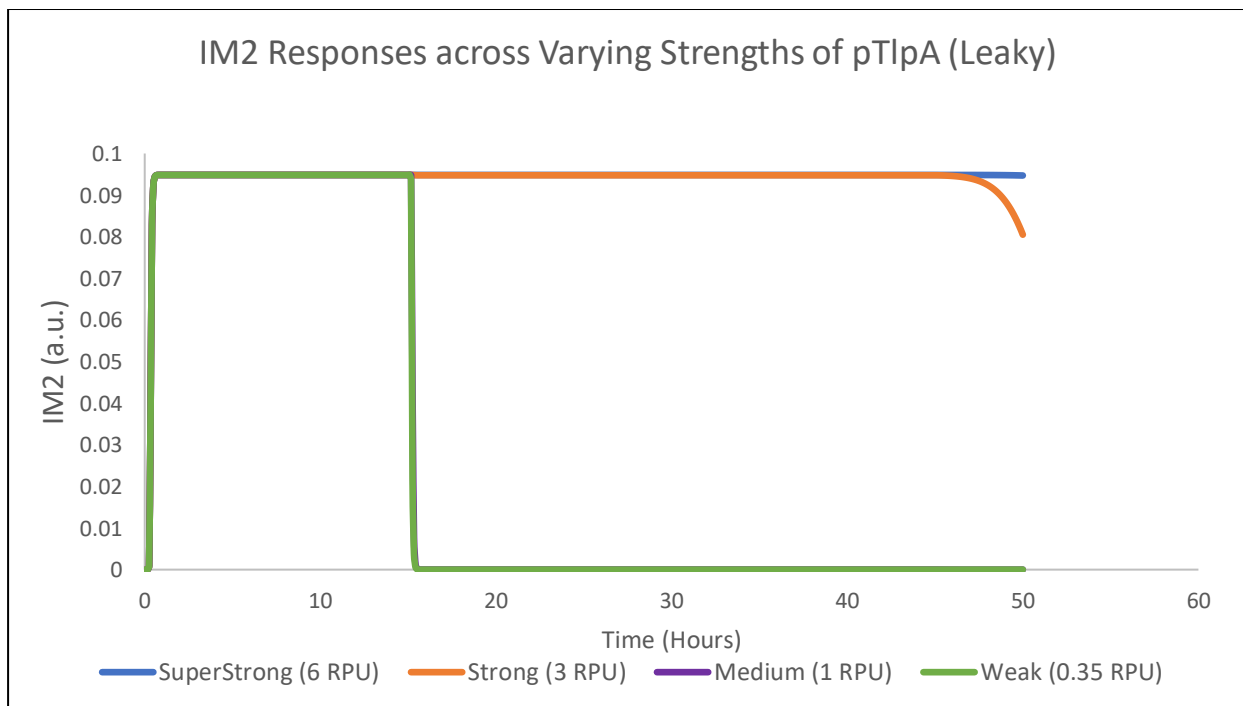
Figure 14 Changing pTlpA Promoter Strength under leaky conditions. As our sensitivity analysis pointed out, pTlpA is the most sensitive part of the entire Automated Circuit. We can observe that decreasing the promoter strength of pTlpA results in the kill switch functioning incorrectly. Likewise, we can observe that increasing the promoter strength of pTlpA makes the circuit ignore any parameter change and therefore, killing is not observed. The ideal pTlpA promoter strength to correct leaky pTlpA performance, is one that is strong enough to drive the circuit without hindering performance.

1) How does Temperature Promoter Strength affect the output response?

The strength of temperature promoter has a significant impact on antitoxin IM2 expression. Like the binding affinity scenario, the ideal temperature promoter strength is one that is high enough to drive output through the intermediate parts of the circuit and yet, weak enough such that the circuit is still sensitive to changes and can activate killing in the correct environment. We can observe in Figure 14, that at low temperature promoter strengths, the temperature promoter fails to drive the intermediate parts of the circuit, and therefore the circuit is unable to produce anti-toxin IM2. Likewise, too high of a temperature promoter strength, overly expresses IM2 production such that the kill switch is rendered ineffective. Therefore, the temperature strength should be in a "goldilocks zone" where the strength is high enough to drive the circuit, but not too high that performance is lost.

Considering these results, we wanted to generate a functional model of Automated Circuit when the temperature promoter strength is weak. We completed a functional model of the Automated Circuit that uses a weak temperature promoter by changing the kinetic values of the intermediate cello parts. The results can be seen in Figure 15.
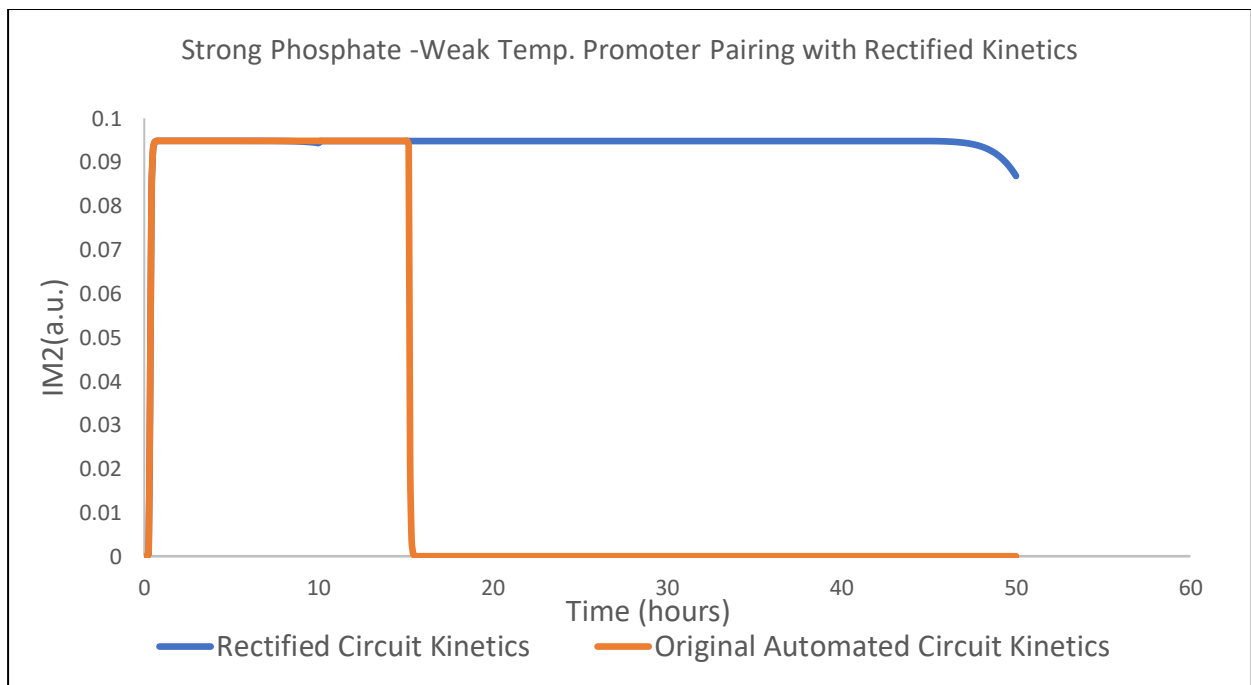
Figure 15 By changing the kinetic values of intermediate parts in the original Automated Circuit, we successfully obtained functional model using the same phosphate promoter paired to a weak temperature promoter. This improvement, shows an optimisation of how one can the weak promoter in the automated circuit.

Comparison between Designs

Comparing the two designs, we can observe that both have their own advantages and disadvantages. The Automated Circuit offers increased response control, modularity, and buffering against some mutations and error. The Automated Circuit offers increased response control because each of its input sensors is independent of each other. Therefore, a user can not only develop complex logic to control output, but can also rely less on ensuring that all sensors work 100% since the output can be reliant on multiple input sensors. For example, unlike the Simple Circuit where the input promoters are cascaded, in the automated circuit, the input promoters are independent and therefore in the case of a leaky promoter, the effect it has on the output response is less in Automated Circuit than in the Simple Circuit. In addition, the Automated Circuit offers modularity because a user can simply replace the input sensors with other input sensors and make slight modifications to the kinetics involved. However, in the Simple Circuit, a user were to use different sensors, they may need to overhaul the genetic circuit design depending on the promoter types. Finally, the automated circuit offers buffering against some mutations and small errors. Because of the intermediate processes involved between input and output, some small errors caused by mutation are not significant in changing the output response of the genetic circuit. On the other hand, if mutations occurred in the Simple Circuit, because of the cascaded configuration and high dependency on parts, errors will have a large effect on the response output and can potentially cause failure.

However, there are limitations that exist in the Automated Circuit that are made up for in the Simple Circuit. One of the biggest drawbacks in the Automated Circuit is complexity. With increasing complexity comes issues surrounding propagation delay, metabolic stress, and construction difficulty. For example, when compared to the functional model of the Automated Circuit, the functional model of the Simple Circuit performed 33% faster killing thanks to its simple design. However, as mentioned earlier, because of its simple design, the Simple Circuit sacrifices modularity and control for faster response and less metabolic stress. This trade-off, as shown through modelling, is a reasonable justification as to why we construct the simple model in the lab. After all, we aim to generate a kill switch as a secondary function that can kill the probiotic host quickly and effectively too. Therefore, we can propose that for simple kill switches (less than two input sensors), a user can employ the Simple Circuit design, and for more complex systems (more than two input sensors), a user can employ the Automated Circuit design.

Conclusion

Using our E2 Chassis and modelling workflow, we successfully demonstrate in silico how a customised genetic circuit for engineered probiotics can be made easier. Our results from modelling can be used by the experimenter to guide them during construction. We have mentioned examples of using modelling to recommend solutions to problems that may be faced during the construction of the kill switch. Making the engineering of customised kill switches for probiotics easier enables anyone to use their own biosensors to develop their own kill switch for engineered probiotics. We hope that with the proliferation of different sensors and killing mechanisms, anyone will be able to utilise our workflow to develop kill switches and in turn, meet regulatory standards.

References

[1] M. Levi and J. P. Knochel, "Disorders of Phosphate Metabolism," in *Threapy of Renal Diseases and Related Disorders*, Springer Science & Business Media, 2012, pp. 121-122.

[2] M. Levi and M. Popovtzer, "Disorder of phosphate balance," in *Atlases of Diseases of the Kidney*, 1999, pp. 7-1.

[3] National Environment Agency , "Allowable Limits For Trade Effluent Discharge To Watercourse/ Controlled Watercourse," Singapore Government, Singapore, 2017.

[4] iGEM08_NYMU-Taipei, "external phosphate sensing reporter," iGEM, 29 10 2008. [Online]. Available: http://parts.igem.org/Part:BBa_K116404.

[5] D. I. Piraner, M. H. Abedi, B. A. Moser, A. Lee-Gosselin and M. G. Shapiro, "Tunable thermal bioswitches for in vivo control of microbial therapeutics," *Nature Chemical Biology,* vol. 13, pp. 75-80, 2017.