Modeling the Arsenic Biosensor System

Yizhi Cai‡, Bryony Davidson§, Hongwu Ma‡, Chris French¶
From the ‡School of Informatics, §School of Engineering, ¶School of Biology
University of Edinburgh

Abstract This paper reports the modeling progress of an arsenic biosensor system, which is the iGEM project accomplished in the University of Edinburgh 2006. The arsenic biosensor system sought to address the fatal water pollution problem occurring in many poor countries/areas like Bangladesh by producing a calibratable pH changes in response to a range of arsenic concentrations. A computational model which contains 3 operons, 19 reactions and 17 species has been constructed to shed light on the wet-lab experimental design. By analyzing the sensitivity of each parameter/species, we identified their relative importance in the system which gives the theoretical guide when measuring the variable in wet-lab. The next step research is to refine the model by comparing it with the biological output.

Key words Modeling, Arsenic Biosensor, Edinburgh iGEM 2006

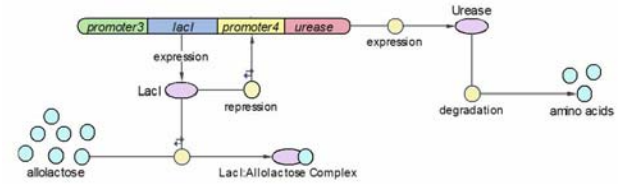
Methodology

The gene expression reactions are modeled using Michaelis-Menten equation while the other reactions are described in Mass-Action equations.

We divided the whole system into three operons, namely Urease operon, LambdaCI operon and LacZ operon.

The Urease operon:

Fig.1 Urease operon



LacI works as the repressor of Urease through occupying the free binding site of Urease promoter. However, when there is allolactose in the operon, LacI binds to allolactose to form complex with higher affinity. One assumption we made here is that there is sufficient allolactose in the system, so Urease is always being produced in this operon.

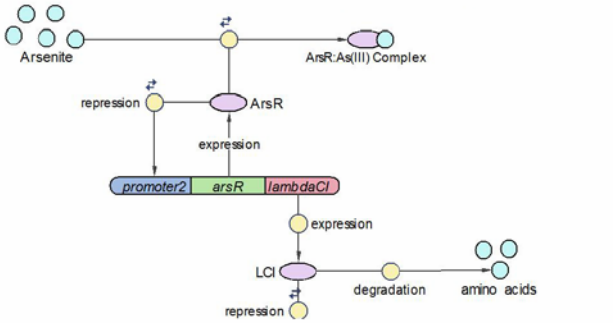
The reactions which involved in Urease operon is summed up as below.

Table 1. Urease operon reactions

No.	Name	Equation
1	LacI production	promoter3->promoter3+LacI
2	LacI binding to allolactose	LacI+allolactose=LacI-allolactose
3	LacI binding to promoter4	LacI+promoter4 = LacI-promoter4
4	Urease production	promoter4->promoter4+Urease
5	LacI degradation	LacI->null
6	Urease degradation	Urease->null

LambdaCI Operon:

Fig.2 LambdaCI operon



LambdaCI operon responses to low concentration of arsenic, e.g., 5ppb. Similar with Urease operon, ArsR represses both LambdaCI and itself (negative feedback loop) through binding site competition. When arsenic presents

in the system, ArsR binds to it with affinity higher than it binds to the promoter. That means with arsenic input, LambdaCI is produced and represses the Urease operon through occupying the free binding site of Urease promoter.

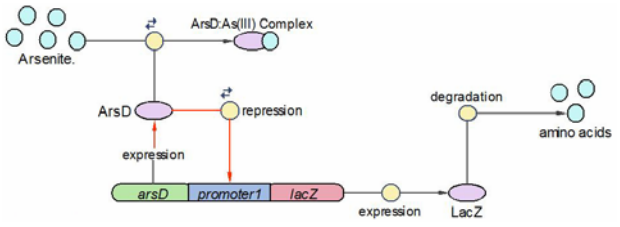
The reactions involved in LambdaCI operon are summarized as below:

Table 2. The LambdaCI operon reactions

No.	Name	Equation
7	ArsR production	promoter2->promoter2+ArsR
8	ArsR binding to Arsenic	ArsR+2As(III)=ArsR-2As(III)
9	ArsR binding to promoter2	2ArsR+promoter2=2ArsR-promoter2
10	LCI production	promoter2->promoter2+LCI
11	LCI binding to promoter 4	LCI+promoter4=LCI-promoter4
12	ArsR degradation	ArsR->null
13	LCI degradation	LCI->null

LacZ operon:

Fig.3 LacZ operon



The LacZ operon works in the similar mechanism as LambdaCI operon, but it responses to higher concentration of arsenic, e.g., 20pppb. ArsD represses LacZ and itself until high concentration of arsenic presents in the system. In other words, when there is high concentration of arsenic input, this operon will produce LacZ as output.

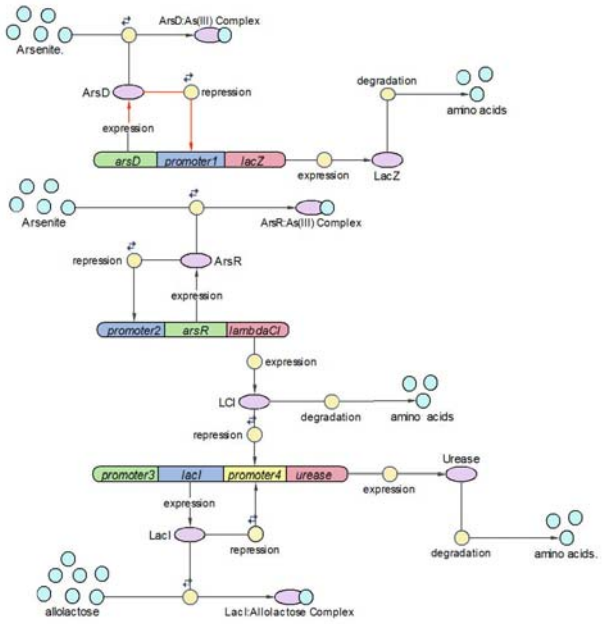
The reactions involved in this operon are summed up as below:

Table 3. The lacZ operon reactions

No.	Name	Equation
14	ArsD production	promoter1->promoter1+ArsD
15	ArsD binding to Arsenic	ArsD+2As(III)=ArsD-2As(III)
16	lacZ production	promoter1->promoter1+lacZ
17	ArsD binding to promoter1	2 ArsD+promoter1=2ArsD-promoter1
18	ArsD degradation	ArsD->null
19	lacZ degradation	lacZ->null

Assemble three operons into a whole system

Fig.4 The full system of arsenic biosensor



After three operons have been constructed and tested, they are assembled into the whole arsenic biosensor system. The system works follows the principles as:

1) When there is no arsenic input, only Urease operon works and produces Urease as the system output.

2) When there is low concentration of arsenic, Urease and LambdaCI operons work and

produce Urease and LambdaCI. Because LambdaCI represses the production of Urease, the concentration of Urease is lower than no arsenic condition.

3) When the concentration of arsenic rises to high level, all three operons work together. LacZ is produced together with Urease.

The complete reaction set is given with the initial condition bellow:

Table 4. Reaction Set[1]

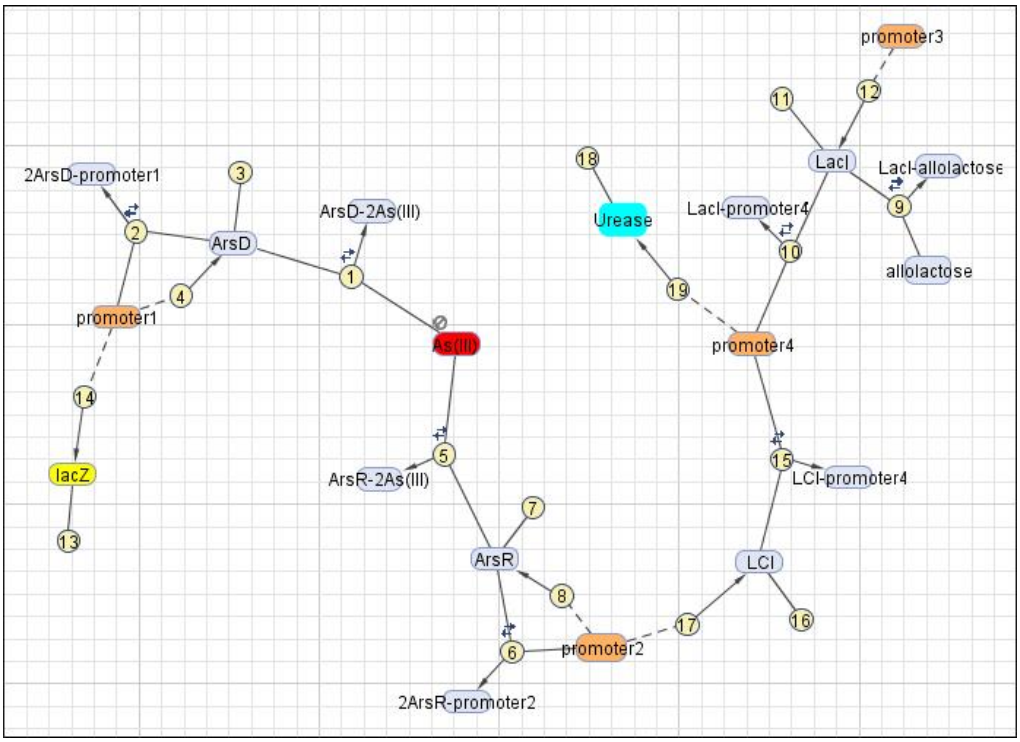
No.	Name	Equation	Rate Law	Parameters
1	LacI production	promoter3->promoter3+LacI	Michaelis-Menten	V12m=0.5, K12m=40
2	LacI binding to allolactose	LacI+allolactose=LacI-allolactose	Mass Action	K9=10000, K-9=0.1
3	LacI binding to promoter4	LacI+promoter4 = LacI-promoter4	Mass Action	K10=1000, K-10=0.5
4	Urease production	promoter4->promoter4+Urease	Michaelis-Menten	V19m=10, K19m=40
5	LacI degradation	LacI->null	Mass Action	K11=0.1
6	Urease degradation	Urease->null	Mass Action	K18=0.1
7	ArsR production	promoter2->promoter2+ArsR	Michaelis-Menten	V8m=10, K8m=25
8	ArsR binding to Arsenic (III)	ArsR+2As(III)=ArsR-2As(III)	Mass Action	K5=1000, K-5=0.65
9	ArsR binding to promoter2	2ArsR+promoter2=2ArsR-promoter2	Mass Action	K6=10000, K-6=0.65
10	LCI production	promoter2->promoter2+LCI	Michaelis-Menten	V17m=10, K17m=25
11	LCI binding to promoter 4	LCI+promoter4=LCI-promoter4	Mass Action	K15=10000, K-15=0.5
12	ArsR degradation	ArsR->null	Mass Action	K7=0.05
13	LCI degradation	LCI->null	Mass Action	K16=0.1
14	ArsD production	promoter1->promoter1+ArsD	Michaelis-Menten	V4m=0.5, K4m=75
15	ArsD binding to Arsenic (III)	ArsD+2As(III)=ArsD-2As(III)	Mass Action	K1=1000 K-1=0.65
16	lacZ production	promoter1->promoter1+lacZ	Michaelis-Menten	V14m=25, K14m=10
17	ArsD binding to promoter1	2 ArsD+promoter1=2ArsD-promoter1	Mass Action	K2=10000, K-2=0.65
18	ArsD degradation	ArsD->null	Mass Action	K3=0.05
19	lacZ degradation	lacZ->null	Mass Action	K13=0.1

Table 5. Initial Concentration

No.	Species	Initial Concentration (nMol)
	ArsD	25
	As(III)	40
	2ArsD-promoter1	25
	promoter1	5
	ArsR	25
	2ArsR-promoter2	25

[1] The units for first, second and third order rate constants are expressed in units of $second^{-1}$, $nMol^{-1}\times second^{-1}$ and $nMol^{-2}\times second^{-1}$ respectively.

	promoter2	5
	LCI	4
	LacI	0.1
	LacI-allolactose	0.1
	allolactose	1000
	LacI-promoter4	0.1
	promoter4	25
	LCI-promoter4	0.1
	Urease	0.1
	promoter3	5
	Other species	0



This reaction map is generated from the reaction set above using SimBiology Toolbox.

Software Used:

In order to model the quantitative behavior of the dynamic system, many modeling software systems have been developed in the past few years. In this project, three kinds of software systems are used for modeling. This section introduces their functions and features.

1. System biology toolbox for Matlab: SimBiology toolbox provides functions for modeling, simulating, and analyzing biochemical pathways on basis of the powerful computing engine of Matlab. Aside from the conventional typing reaction equation, SimBiology provides a user-friendly 'dragging-and-dropping' block diagram editor to build a new model. Thanks to the powerful computing ability of Matlab, SimBiology integrates a wide range of numerical solvers for both stochastic and deterministic simulations. Another strength of SimBiology is its analyzing ability which includes sensitivity analysis, parameter estimation and conservation of moieties. Last but not the least, SimBiology provides user-defined plotting function which was widely used in this research project. However SimBiology is commercial software, it costs the budgeted researchers extra expense to

purchase the license for this toolbox. Also, there is no GUI for sensitivity analysis and parameters scan functions, so users should obtain the skill of programming with Matlab.

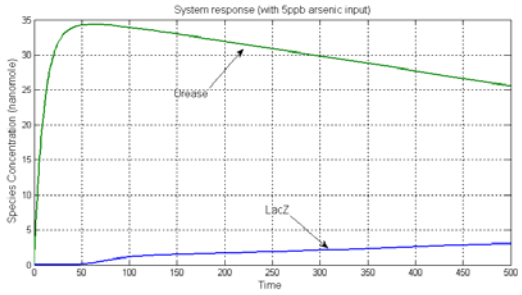
2. BIOCHAM: BIOCHAM stands for biochemical abstract machine which is a formal modeling environment for system biology. Because of its rule-based language and temporal logic based language, BIOCHAM can offer an automatically reasoning tool for querying the temporal properties of the system under all its possible behaviors. It is very useful for constructing models especially when numerical data is unavailable. BIOCHAM also provides a state-of-the-art symbolic model checker for handling the complexity of large highly non-deterministic models. Furthermore, it provides simulation results via its graphical interface. One disadvantage of BIOCHAM is that because it is initially developed under UNIX, it uses command line rather than GUI to build new models, requiring users to write scripts by hand. Another problem was that when transplanting the program from UNIX to Windows, the model checker of NuSMV does not work well. So this project used BIOCHAM in Fedora 5 rather than Windows XP.

3. COPASI: COPASI is freeware developed with collaboration of VBI and EMLR. It provides almost the same functions as SimBiology, though not quite powerful. But compared with SimBiology, it provides a friendly user interface for model analysis, such as parameter estimation, and parameter scan. But there is no parameters/species sensitivity analysis function in COPASI, and also it is not very stable in use, crashing without any responses.

To sum up, each software has its own pros and cons. A good strategy is to apply them for different purposes, for example, using SimBiology to analyze the sensitivity of the model, and using COPASI to scan the most sensitive parameters. When logical queries are needed, BIOCHAM should be the first choice. SimBiology is suitable for generating professional plots, however due to the license requirement, COPASI can be the alternative choice to to run simulations and export the results to such plotting software as SigmaPlot.

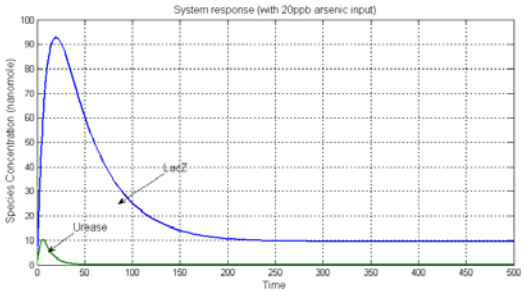
Simulation result:

Fig.5 System response with 5ppb arsenic input



When the concentration of arsenic is low, like 5ppb input, the production of Urease is at a high level above 25 units while lacZ production is at a basal level less than 5 units.

Fig.6 System response with 20ppb arsenic input



When the arsenic concentration rises to higher level like 20ppb, Urease decreases to almost zero because of the repression by LambdaCI and the concentration of lacZ increases to about 10 units because the ArsD has been captured by Arsenic molecules.

Because of the different responses of the system, a significant change of pH value is obtained.

Sensitivity Analysis

To provide suggestions to the experimental biologists which proteins or which reaction rates should be measured, it is important to find out the relative importance of each parameter in the system. Parameter sensitivity analysis is used to identify which parameters are more critical in effecting the output of the pathway, and help to gain a deeper insight of the structure and the function of the system.

The sensitivity is calculated in this way:

$$sensitivity = \frac{\Delta SystemOuput}{\Delta Parameter}$$

The Matlab scripts used for sensitivity analysis can be found in the appendix.

Fig.7 Sensitivity of lacZ with respect to parameter 1-16

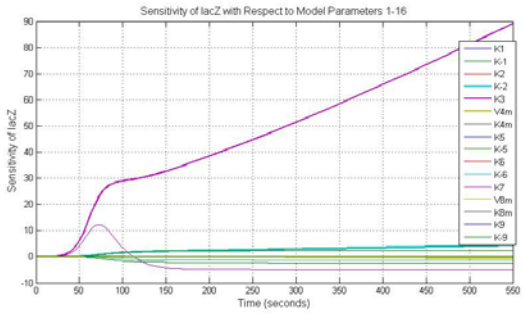


Fig.8 Sensitivity of lacZ with respect to parameter 17-32

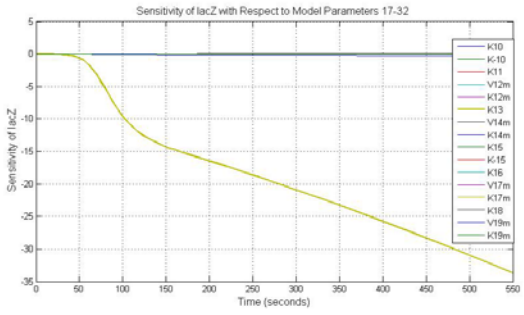


Table 6. The most sensitive parameters affecting lacZ

Name	Description	Peak	Value
K-1	ArsD-2AS(III) dissociate rate	-2	0.65/s
K-2	2ArsD-promoter1 dissociate rate	4	0.65/s
K3	ArsD degradation rate	90	0.05/s
K7	ArsR degradation rate	12	0.05/s
K13	LacZ degradation rate	-35	0.1/s

Fig.9 Sensitivity of Urease with respect to parameter 1-16

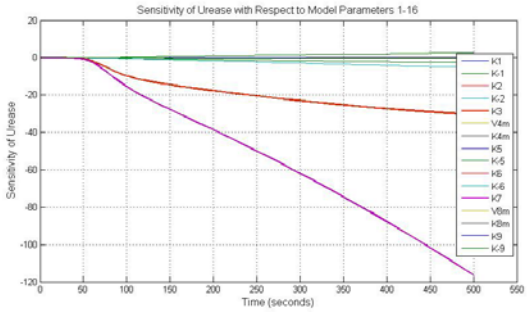


Fig.10 Sensitivity of lacZ with respect to parameter 17-32

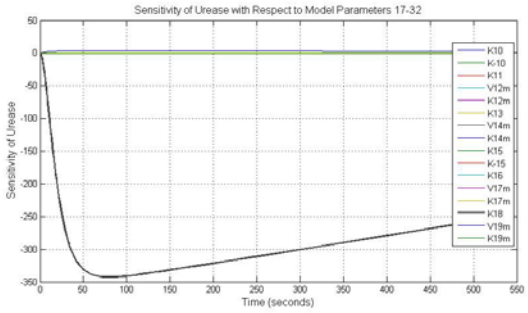


Table 7. The most sensitive parameters affecting Urease

Name	Description	Peak	Value
k3	ArsD degradation rate	-30	0.05/s
k-6	2ArsR-promoter2 dissociate rate	-5	0.65/s
k7	ArsR degradation rate	-120	0.05/s
K18	Urease degradation rate	-350	0.1/s
V19m	Urease production maximum rate	3.5	10 nMol/s

Fig.11 Sensitivity of lacZ with respect to model species

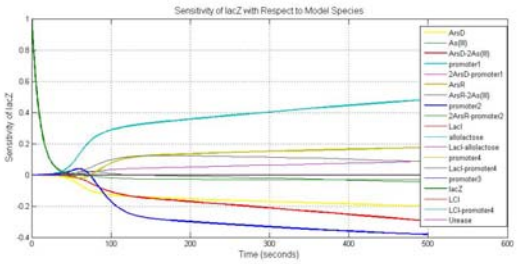


Table 8. The most sensitive species affecting lacZ

Name	Initial concentration (nMol)
promoter1	5.0
promoter2	5.0
ArsD-2As(III)	0

ArsD	25
ArsR	25

Fig.12 Sensitivity of lacZ with respect to model species

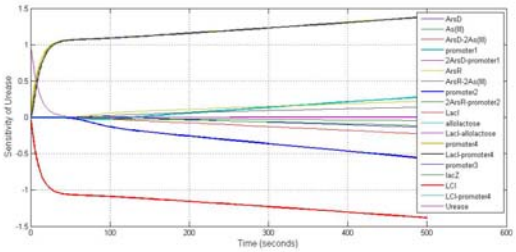


Fig.13 Varying the value of K-2 effect on lacZ

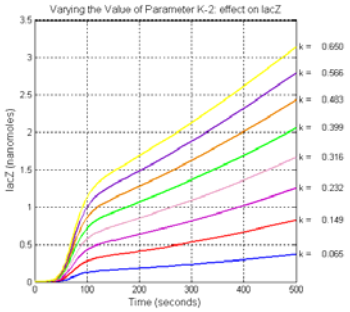


Fig.14 Varying the value of K3 effect on lacZ

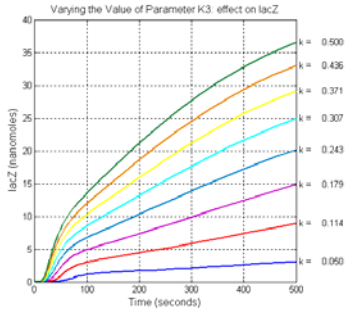


Fig.15 Varying the value of K7 effect on lacZ

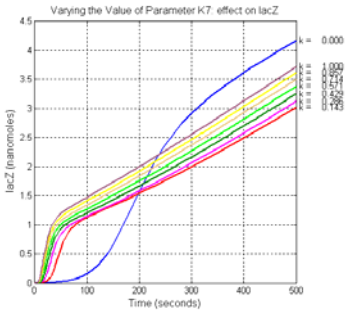


Fig.16 Varying the value of K13 effect on lacZ

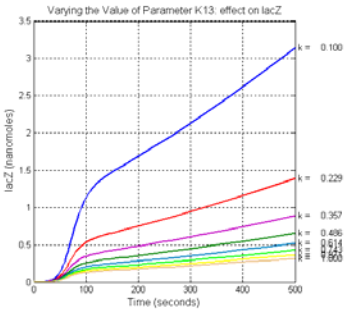


Fig.17 Varying the value of K-1 effect on lacZ

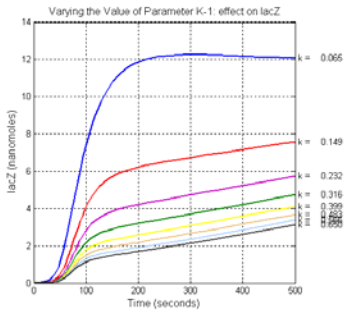


Fig.18 Varying the value of K3 effect on lacZ

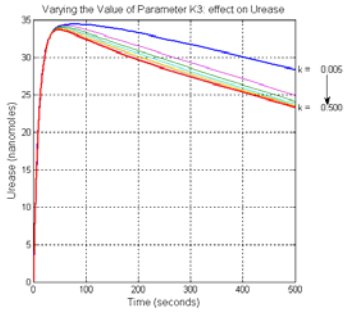


Fig.19 Varying the value of K7 effect on Urease

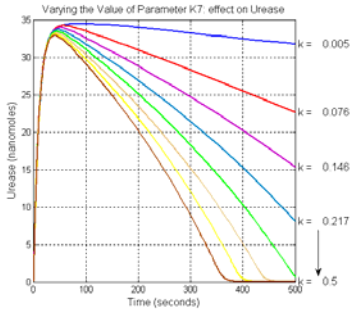


Fig.20 Varying the value of K_{18} effect on Urease

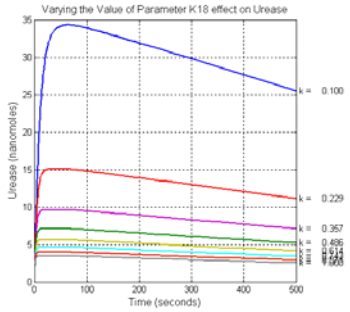


Fig.21 Varying the value of K_{-6} effect on Urease

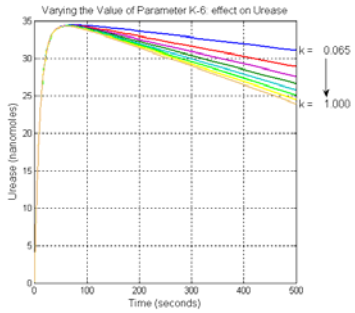


Fig.22 Varying the value of promoter2 effect on lacZ

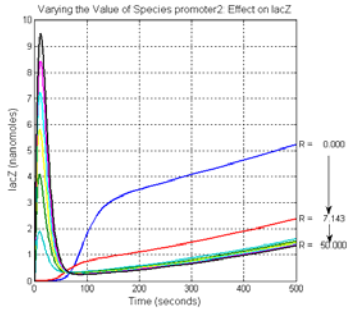


Fig.23 Varying the value of ArsD-2As(III) effect on lacZ

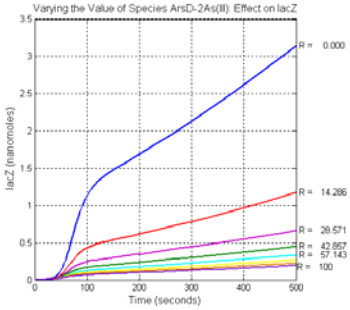


Fig.24 Varying the value of ArsD effect on lacZ

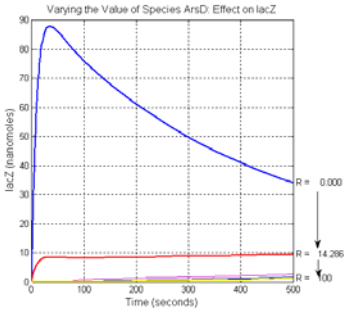


Fig.25 Varying the value of ArsR effect on lacZ

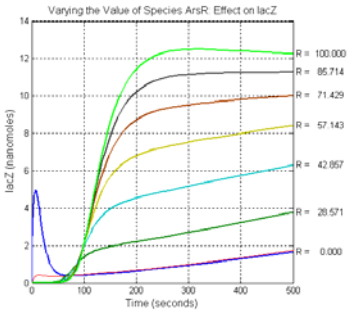


Fig.26 Varying the value of promoter1 effect on lacZ

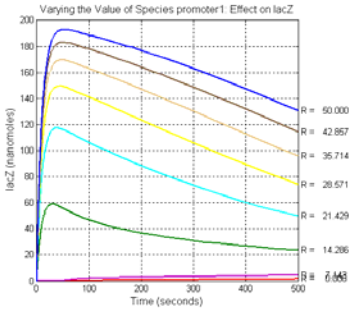


Fig.27 Varying the value of promoter4 effect on Urease

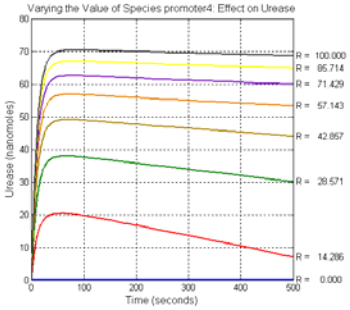


Fig.28 Varying the value of ArsR effect on Urease

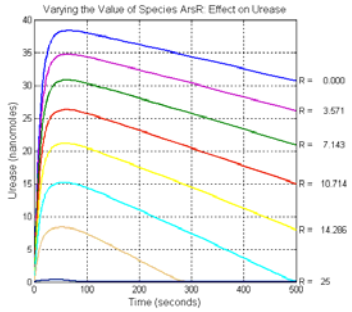


Fig.29 Varying the value of promoter4 effect on Urease

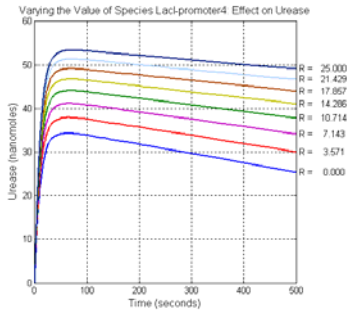
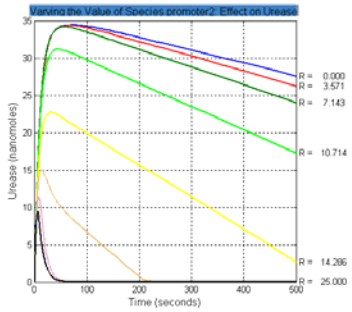


Fig.30 Varying the value of promoter2 effect on Urease



Appendix 1. The ordinary differential equations for the arsenic biosensor system

$$(1)\frac{d[ArsD]}{dt}=\frac{V_{4M}\times[promoter1]}{K_{4M}+[promoter1]}-K_3\times[ArsD]\times[As(III)]-K_1\times[ArsD]\times[As(III)]+K_{-1}\times[ArsD-As(III)]$$

$$(2)\frac{d[ArsD-2As(III)]}{dt}=K_1\times[ArsD]\times[As(III)]^2-K_{-1}\times[ArsD-2As(III)]$$

$$(3)\frac{d(Ar(III))}{dt}=-2\times(K_1\times[ArsD]\times[As(III)^2]-K_{-1}\times[ArsD-2As(III)])-2(K_5\times[ArsR]\times[As(III)]^2-K_{-5}\times[ArsR-2As(III)])$$

$$(4)\frac{d([2ArsD-promoter1])}{dt}=K_2\times[ArsD]^2\times[promoter1]-K_{-2}\times[2ArsD-promoter1]$$

$$(5)\frac{d([promoter1])}{dt}=-K_2\times[ArsD]^2\times[promoter1]+K_{-2}\times[2ArsD-promoter]$$

$$(6)\frac{d([lacZ])}{dt}=\frac{V_{14M}\times[promoter1]}{K_{14M}+[promoter1]}-K_{13}\times[lacZ]$$

$$(7)\frac{d([ArsR])}{dt}=-K_5\times[ArsR]\times[As(III)]^2+K_{-5}\times[ArsR-2As(III)]-2\times K_6\times[ArsR]^2\times[promoter2]$$

$$+2\times K_{-6}\times[2ArsR-promoter2])-K_7\times[ArsR]+\frac{V_{8M}\times[promoter2]}{K_{8M}+[promoter2]}$$

$$(8)\frac{d([ArsR-2As(III)])}{dt}=K_5\times[ArsR]\times[As(III)]^2-K_{-5}\times[ArsR-2As(III)]$$

$$(9)\frac{d([2ArsR-promoter2])}{dt}=K_6\times[ArsR]^2\times[promoter2]-K_{-6}\times[2ArsR-promoter2]$$

$$(10)\frac{d([promoter2])}{dt}=-K_6\times[ArsR]^2\times[promoter2]+K_{-6}\times[2ArsR-promoter2]$$

$$(11)\frac{d(LCI)}{dt}=-K_{16}\times[LCI]-K_{15}\times[LCI]\times[promoter4]+K_{-15}\times[LCI-promoter4]+\frac{V_{17M}\times[promoter2]}{K_{17M}+[promoter2]}$$

$$(12)\frac{d([lacI])}{dt}=\frac{V_{12M}\times[promoter3]}{K_{12M}+[promoter3]}-K_9\times[allolactose]\times[lacI]+K_{-9}\times[allolactose-lacI]$$
$$-K_{10}\times[lacI]\times[promoter4]+K_{-10}\times[lacI-promoter4]-K_{11}\times[lacI]$$

$$(13)\frac{d([lacI-allolactose])}{dt}=K_9\times[lacI]\times[allolactose]-K_{-9}\times[lacI-allolactose]$$

$$(14)\frac{d([allolactose])}{dt}=-K_9\times[lacI]\times[allolactose]+K_{-9}\times[lacI-allolactose]$$

$$(15)\frac{d([lacI-promoter4])}{dt}=K_{10}\times[lacI]\times[promoter4]-K_{-10}\times[lacI-promoter4]$$

$$(16)\frac{d([promoter4])}{dt}=-K_{15}\times[lacI]\times[promoter4]+K_{-15}\times[lacI-promoter4]$$

$$(17)\frac{d(LCI-promoter4)}{dt}=K_{15}\times[LCI]\times[promoter4]-K_{-15}\times[LCI-promoter4]$$

$$(18)\frac{d([Urease])}{dt}=\frac{V_{19M}\times[promoter4]}{K_{19M}+[promoter4]}-K_{18}\times[Urease]$$

Appendix 2. The Matlab scripts for multi-parameter sensitivity analysis

```
sbioloadproject Biosensor        %change the project name to replace “Biosensor” here
m1
m1.Species
m1.Reactions
csObj = getconfigset(m1);
% change stop time to the time you want the simulation to run for in the line below
set(csObj, 'StopTime', 500);
csObj
csObj.RunTimeOptions.StatesToLog
% in line below change urease for the output you want to monitor
csObj.RunTimeOptions.StatesToLog = sbioselect...
(m1, 'type', 'species', 'Where', 'Name', '==', 'Urease');
csObj.RunTimeOptions.StatesToLog
```

```
set(csObj.SolverOptions, 'SensitivityAnalysis', true);

pif = [sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K1');... %change the name of parameter here

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K-1')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K2')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K-2')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K3')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'V4m')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K4m')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K5')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K-5')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K6')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K-6')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K7')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'V8m')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K8m')
```

```
sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K9')

sbioselect(m1, 'Type', 'parameter', 'Where', 'Name', '==', 'K-9')];

set(csObj.SensitivityAnalysisOptions, 'ParameterInputFactors', pif);

set(csObj.SensitivityAnalysisOptions, 'Normalization', 'None');

warning('off', 'MATLAB:divideByZero');

tsObj = sbiosimulate(m1);

[T, R, snames, ifacs] = sbiogetsensmatrix(tsObj);

size(R)

R2 = squeeze(R);

figure;

plot(T,R2);

% in lines below change the labels to suit the graph you want to draw

title('Sensitivity of Urease with Respect to Model Parameters 1-16');

xlabel('Time (seconds)');

ylabel('Sensitivity of Urease');
```

Appendix 3. The Matlab script for Multi-species sensitivity analysis

```
sbioloadproject Biosensor

m1

m1.Species

m1.Reactions

csObj = getconfigset(m1);

% change stop time to the time you want the simulation to run for in the line below

set(csObj, 'StopTime', 500);

csObj

csObj.RunTimeOptions.StatesToLog

% in line below change urease for the output you want to monitor

csObj.RunTimeOptions.StatesToLog = sbioselect...

(m1, 'type', 'species', 'Where', 'Name', '==', 'Urease');

csObj.RunTimeOptions.StatesToLog
```

```
set(csObj.SolverOptions, 'SensitivityAnalysis', true);

sif = [sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'ArsD');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'As(III)');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'ArsD-2As(III)');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'promoter1');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', '2ArsD-promoter1');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'ArsR');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'ArsR-2As(III)');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'promoter2');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', '2ArsR-promoter2');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'LacI');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'allolactose');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'LacI-allolactose');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'promoter4');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'LacI-promoter4');...
```

```
sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'promoter3');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'lacZ');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'LCI');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'LCI-promoter4');...

sbioselect(m1, 'Type', 'species', 'Where', 'Name', '==', 'Urease')

];

set(csObj.SensitivityAnalysisOptions, 'SpeciesInputFactors', sif);

set(csObj.SensitivityAnalysisOptions, 'Normalization', 'None');

warning('off', 'MATLAB:divideByZero');

tsObj = sbiosimulate(m1);

[T, R, snames, ifacs] = sbiogetsensmatrix(tsObj);

size(R)

R2 = squeeze(R);

figure;

plot(T,R2);
```

```
% in lines below change the labels to suit the graph you want to draw

title('Sensitivity of Urease with Respect to Model Species');

xlabel('Time (seconds)');

ylabel('Sensitivity of Urease');
```

Appendix 4. The Matlab lab script for parameter scan

```
sbioloadproject Tuning

m1

m1.Species

m1.Reactions

csObj = getconfigset(m1);

% change stop time to the time you want the simulation to run for in the line below

set(csObj, 'StopTime', 5000);

csObj

csObj.RunTimeOptions.StatesToLog

% in line below change urease for the output you want to monitor

csObj.RunTimeOptions.StatesToLog = sbioselect...

(m1, 'type', 'species', 'Where', 'Name', '==', 'promoter1');
```

```
set(csObj.SolverOptions, 'SensitivityAnalysis', false);

set(csObj.SensitivityAnalysisOptions, 'Normalization', 'None');

h1 = figure;

ax1 = gca(h1);

% in next line change Vm4 to the name of the parameter you want to scan

p = sbioselect(m1, 'Type', 'parameter', 'Name', 'K2');

% in next line the first number is the lower limit and the second number the upper limit

% of your range and the last number is the number of scans in that range

s = linspace(1000, 100000, 8);

for k = s

set(p, 'Value', k);

[T,x,names] = sbiosimulate(m1);

str = sprintf( ' k = %8.3f',k);
```

```
plot(ax1,T,x(:,1));

figure(h1);

hold on;

text(T(end),x(end,1),str);

end

axis([ax1], 'square');

% Change these lines to suit your data

title(ax1, 'Varying the Value of Parameter K18 effect on Urease');

xlabel(ax1, 'Time (seconds)');

ylabel(ax1, 'Urease (nanomoles)');
```

Appendix 5. The Matlab Script for species scan

```
sbioloadproject Biosensor

m1

m1.Species

m1.Reactions

csObj = getconfigset(m1);

% change stop time to the time you want the simulation to run for in the line below

set(csObj, 'StopTime', 500);

csObj

csObj.RunTimeOptions.StatesToLog

% in line below change urease for the output you want to monitor

csObj.RunTimeOptions.StatesToLog = sbioselect...

(m1, 'type', 'species', 'Where', 'Name', '==', 'Urease');
```

```
set(csObj.SolverOptions, 'SensitivityAnalysis', false);

set(csObj.SensitivityAnalysisOptions, 'Normalization', 'None');

h1 = figure;

ax1 = gca(h1);

% in next line change Vm4 to the name of the parameter you want to scan

p = sbioselect(m1, 'Type', 'species', 'Name', 'promoter1');

% in next line the first number is the lower limit and the second number the upper limit

% of your range and the last number is the number of scans in that range

s = linspace(0, 50, 8);

for spR = s

set(p, 'InitialAmount', spR);

[T,x,names] = sbiosimulate(m1);

str = sprintf( ' R = %8.3f',spR);

plot(ax1,T,x(:,1));

figure(h1);
```

```
hold on;

text(T(end),x(end,1),str);

end

axis([ax1], 'square');

% Change these lines to suit your data

title(ax1, 'Varying the Value of Species promoter1: Effect on Urease');

xlabel(ax1, 'Time (seconds)');

ylabel(ax1, 'Urease (nanomoles)');
```