THE UNIVERSITY *of* EDINBURGH
School of Biological Sciences

Final Assignment

Practical Systems Biology

Student Exam Number: B132062

Introduction

Many gram-negative bacteria synthesize N-acylhomoserine lactones (AHLs) and then use transcription factors of the LuxR family to sense and respond to AHL accumulation in the environment. This phenomenon is termed quorum sensing and it is a mechanism that a lot of bacteria use to regulate gene expression in response to fluctuations in cell-population density. (You. L, et. al 2004).

In the paper **Programmed population control by cell–cell communication and regulated killing**, Lingchong You and collaborators harness these mechanisms of quorum sensing to build and characterize a 'population control' circuit that autonomously regulates the density of Escherichia coli population, based on the LuxI/LuxR system from the marine bacterium Vibrio fischeri. (See figure 1). The goal of their work was to see if it was possible to programme the dynamics of a population despite variability in the behaviour of individual cells, and to create a mathematical model that depicted this behaviour.

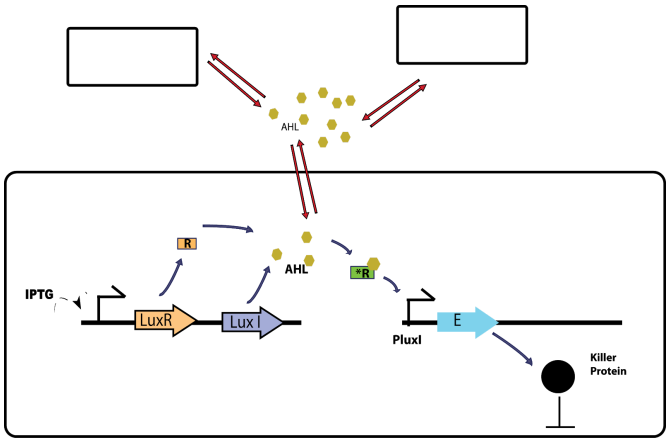

Figure1. Population Control Circuit proposed by You L, et al. (2004). Upon IPTG induction LuxR and LuxI start being transcribed resulting in the production of LuxR protein (orange) and AHL molecules (yellow hexagons). As cell density increases the concentration of AHL also increases inside and outside of the cell due to the diffusion of the molecules through the membrane. At high enough concentrations it binds and activates Lux R. The activated LuxR*(Green) then binds to the LuxI promoter which in turn starts releasing the Ccdb toxin (E) that causes cell death at high enough concentrations.

Mathematical Model:

In order to better understand their system the authors modelled the major kinetic events dictating the function of their circuit with a set of 3 equations and certain assumptions , constants and parameters obtained from experimental data.

The equations describing the model are given as:

$$\frac{dN}{dt} = kN(1 - \frac{N}{N_m}) - dEN. \quad (1)$$

$$\frac{dN}{dt} = k_E A - d_E E. \quad (2)$$

$$\frac{dN}{dt} = \nu aN - d_A A. \quad (3)$$

Where **(1)** is a modified logistic curve with an added death rate term (d*E*N) that describes the growth and death of the cells, (2) is an equation that describes the production and degradation of the toxin E and (3) is an equation that describes production of the AHL and its degradation.

Some of the Assumptions made in the model are:

- In cells that don't have the circuit expressed, changes in viable cell density (**N**, ml^{-1}) follow logistic kinetics with an intrinsic per capita growth rate of **k** (h^{-1}) and a carrying capacity of **Nm**(ml^{-1})
- For circuit-regulated growth, the cell death rate is proportional to the intracellular concentration of the killer protein **E** (nM) with a rate constant of **d** ($nM^{-1}\ h^{-1}$)
- The production rate of E is proportional to AHL(**A** nM) with a rate constant of **kE** (h^{-1})
- AHL production rate is proportional to **N** with a rate constant of **vA** ($nM^{-ml}\ h^{-1}$).
- Degradation of the killer protein E and AHL follows first-order kinetics with rate constants of **dE** (h^{-1}) and **dA** (h^{-1}).
- The production term (**kEA**) in equation (2) lumps several intermediate steps together activation of LuxR by AHL, binding of activated LuxR to pluxI, and expression of the killer gene
- Cooperativity of AHL action to be 1

Reproducibility of the Model:

The feasibility of the model was verified by reproducing the model and then comparing it with the results obtained from the paper. In this case **Figure 2a from the paper** was reproduced using the parameters from the paper (shown in **Table 1)** and compared with the results from the paper **(See Figure 2)**.

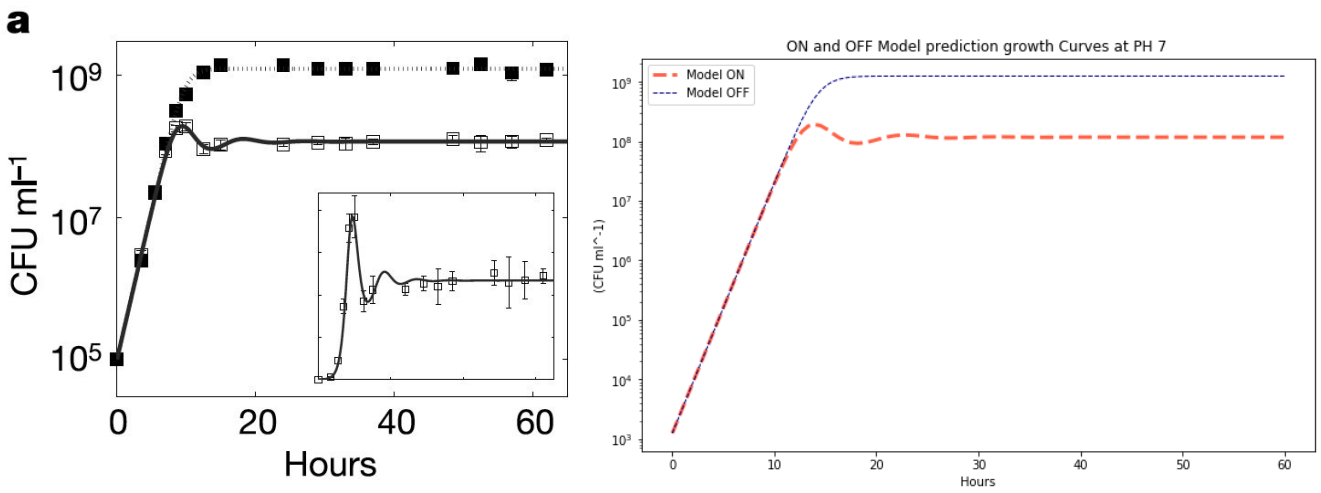

Figure 2. Comparison of Prediction Models in pH 7 from paper by You. et, al. and the manually reproduced model. (LEFT) *Figure* ***a*** *shows the experimentally measured growth curves on a media with pH 7.0. Filled squares represent the growth curve for the uninduced population control circuit (OFF) and open squares represent the the growth curve for the population-control circuit when it is induced (ON). Model Predictions are shown in solid (ON) and dotted (OFF) lines.****(RIGHT)*** *The manually reproduced model is seen. Predicted grown curved fro the model are shown in blue for the uninduced (OFF) and Orange for the induced ON circuit.*

Table 1: Summary of variables and parameters used to replicate the model. All Values were taken from You et.,al (2004)

Variable	Description	Value	Unit
Nm	Carrying capacity	1.24×10^9	ml^{-1}
k	Growth rate constant	0.970	h-1
d	Death Rate constant	4.3×10^9	$nM^{-1}\ h^{-1}$
kE	**Production rate of Toxin**	5	h^{-1}
dE	Degradation rate of Toxin	2	h^{-1}
vA	**Production rate of AHL**	4.8×10^{-7}	$nM\ ml\ h^{-1}$
dA	Degradation rate of AHL	0.639	h^{-1}

Model predictions in both LEFT and RIGHT show that when the cells are expressing the gene circuit at a sufficiently high cell density of around 10^8CFU the Killer protein responds to the activated LuxR and starts causing cell death at approximately hour 15 causing the population concentration to decrease. However, the system doesn't reach steady state immediately, but rather goes through a cycle of damped oscillations before reaching the steady state. These damped oscillations are due to the fact that the level of killer protein starts to decrease due to the decline of AHL, which in turn is due to the decreasing amount of cells. This delay and lack of killer protein then allows the population to increase until it reaches a steady state.
The results of the model match the experimental data shown on the black and white boxes thus validating that the model is able to predict the behavior of the system and that the system indeed has negative feedback and that cell density can be regulated by using cell-cell communication.

Novel Hypothesis

Although the model manages to predict the dynamics of the population, one of the limitations of the proposed circuit, is that the stable value of the cell population N is not appreciably low. This could be a limitation if this system was used as a way to regulate the presence of pathogenic bacteria through the use of quorum sensing.
In order to examine whether the amount of bacteria can be lowered, the mathematical model was used to test the following hypotheses.

1) Since the AHL production rate is proportional to the amount of bacteria, letting the population grow until they reach a steady state without being uninduced should create enough AHL (once it is induced) to lower the overall amount of bacteria since there would be more cells to kill. **(See Figure 3)**

2) Redesigning the circuit proposed in the paper where the IPTG promoter is replaced by a stronger promoter with a higher affinity to RNA polymerase should be able to produce more AHL molecules faster, which would create more killer protein at an earlier period of growth and thus reducing the steady state concentration. **(See Figure 4)**

Results

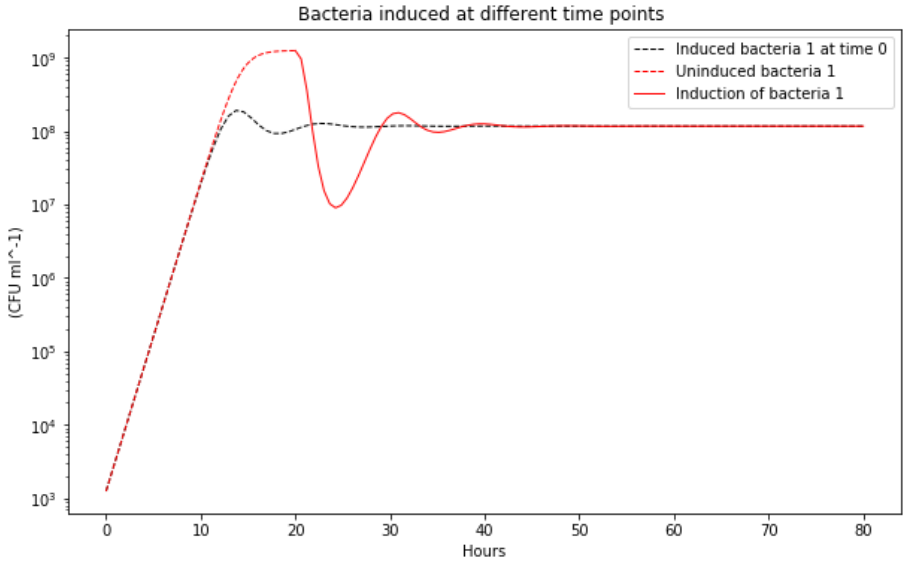

Figure 3. Bacteria induced at different time points. *Figure shows the growth curves of two populations. The population induced at time 0 (black lines) shows the expected behavior of a population with the switch ON as seen on Figure 2. The uninduced bacteria (orange line) are first grown to reach a population size similar to the carrying capacity Tm. Then once it reaches the threshold the bacteria are induced at around time 15 showing a bigger decrease in population and managing to reach a population lower than the steady state of the bacteria induced at time 0. Nonetheless the population size bounces back to normal and reaches the same steady state. * To model this an extra argument Iu was added to the equation (3) turning it into y=((va*Iu)*N)-da*A. This Iu was used as 0 or 1 , where a value of 0 turned the production rate va into 0 and turned the circuit OFF, and a value of 1 maintained the circuit ON.*

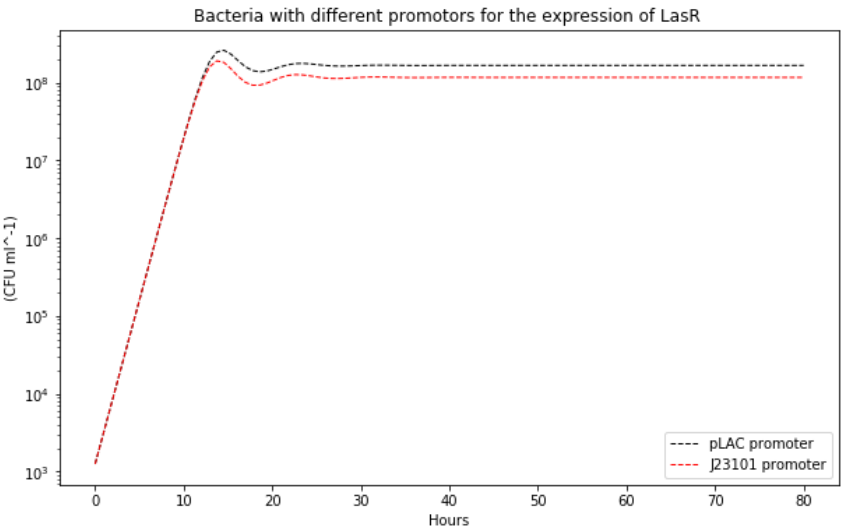

Figure 4. Growth of bacteria with different promoter strengths. Growth curves show the behavior of the bacteria when two promoters of different strength are used. *To model this an extra argument **pr** was added to the equation (3) turning it into y=((va*pr)*N)-da*A. This pr is a number that describes the relative activity of a promoter which is defined as the ratio of the absolute activity of the promoter to the absolute activity of a reference standard promoter. (Kelly,J et al,2009). In this case the relative activity of the pLac promoter which is used in the original genetic construct has a smaller activity than the reference promoter J23101.This activity is assumed to have an effect on the production rate **va** of the AHL molecules and consequently an effect on the amount of killer protein produce. The curve shows that the bacteria grown with a stronger promoter J 23101 has a slight decrease in the final steady state concentration.*

Table 2. Relative Promoter Strength.

Promoter	Efficiency on E.coli DH5 alpha
Plac	0.67
J23101	1

Source:http://parts.igem.org/Part:BBa_R0011:Experience

Conclusions:

The results from figure 3 show that letting the bacteria grow more and then inducing it at a later time does not reduce significantly the population of the bacteria . It also does not reduce at all the final steady state concentration meaning that the first hypothesis was wrong. Nonetheless the results shown on figure 4 suggest that by experimenting with the rate at which the AHL is produced, in this case by changing promoters, the concentration of the bacteria does stabilize at a lower steady state value. Although the reduction is not significant, a solution to this could be to go back to the design of the genetic construct and modify the transcription rates via promoter strength, translation rates via rbs strength and copy numbers.

Overall although their genetic design might not be useful for a real life application such as a biosensor to detect and kill a pathogen, the mathematical model used does prove to be a great tool for trying to understand the behaviour of populations despite the variability in the behaviour of individual cells. This proves to be a great tool for synthetic biology which aims to design systems that can behave in a predictable and robust manner.

References:

You, L., Cox, R. S., Weiss, R., & Arnold, F. H. (2004). Programmed population control by cell–cell communication and regulated killing. Nature, 428(6985), 868-871. doi:10.1038/nature02491

Kelly, J. R., Rubin, A. J., Davis, J. H., Ajo-Franklin, C. M., Cumbers, J., Czar, M. J., . . . Endy, D. (2009). Measuring the activity of BioBrick promoters using an in vivo reference standard. Journal of Biological Engineering, 3(1), 4. doi:10.1186/1754-1611-3-4

CODE:

Figure 2.

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

Nm=1.24e9   # nm is the carrying capacity
k=0.97  #Growth rate constant
ke=5 #Production rate of toxin
de=2 #Degradation rate of toxin
ka=4.8E-07 #Production rate of AHL
da=0.639 #Degradation rate of AHL
d=4.0e-03

def ahlON(y, t):
    N=y[0] #Cell density at given point of time
    E=y[1] #Concentration of toxin killer protein
    A=y[2] #Concentration of AHL
    dydt=np.empty(len(y))
    dydt[0]=k*N*(1-N/Nm)-(d*E*N)
    dydt[1]=ke*A-de*E   #cont
    dydt[2]=ka*N-da*A
    return dydt

def ahlOFF(y1,t):
    N=y1[0]
    ahlOFF=k*N*(1-N/Nm)
    return ahlOFF
```

```
initial_conditions=[1250,0,0]
initial_conditions2=[1250]

t=np.linspace(0,60,100)
y=odeint(ahlON, initial_conditions, t)
y1=odeint(ahlOFF,initial_conditions2,t)
plt.figure(figsize= (10, 6))
plt.plot(t,y[:,0],color='tomato',label='Model ON',linestyle='--',linewidth=3)
plt.yscale('log')
plt.plot(t,y1,color='navy',label='Model OFF',linestyle='--',linewidth=1)
plt.yscale('log')
plt.xlabel('Hours')
plt.ylabel('(CFU ml^-1)')
plt.title('ON and OFF Model prediction growth Curves at PH 7 ')
plt.legend()
plt.show()
```

Figure 3:

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

da=0.639 #Degradation rate of AHL
Nm=1.25e9  #Carrying capacity
k=0.97  #Growth rate constant
d=4.0/1000
ke=5.0 #Production rate of toxin
de=2.0 #Degradation rate of toxin
va=4.8*10**-7 #Production rate of AHL

def ahl(y,t,Iu):
    N=y[0] #Cell density at given point of time
    E=y[1] #Concentration of toxin killer protein
    A=y[2] #Concentration of AHL

    dydt=np.empty(len(y))
    dydt[0]=k*N*(1-N/Nm)-(d*E*N)
    dydt[1]=ke*A-de*E   #cont
    dydt[2]=((va*Iu)*N)-da*A
    return dydt

plt.figure(figsize= (10, 6))
```

```
#bacteria 1 , induced at time 0
initial_conditions=[1250,0,0]
t=np.linspace(0,80,100)
Iu=1
y1=odeint(ahl, initial_conditions, t,args=(Iu,))
plt.plot(t,y1[:,0],color='black',label='Induced bacteria 1 at time 0',linewidth=1,linestyle='--')
# uninduced bacteria 2
initial_conditions=[1250,0,0]
t=np.linspace(0,20,100)
Iu=0
y2=odeint(ahl, initial_conditions, t,args=(Iu,))
plt.plot(t,y2[:,0],color='red',label='Uninduced bacteria 1',linewidth=1,linestyle='--')

#bacteria 2 induced once the population reaches its maximum carrying capacity (Nm)
t=np.linspace(20,80,100)
initial_conditions2=y2[-1] #initial conditions are taken as the last values from the previous graph y2
Iu=1
y3=odeint(ahl, initial_conditions2, t,args=(Iu,))
plt.plot(t,y3[:,0],color='red',label='Induction of bacteria 1 ',linewidth=1)

plt.xlabel('Hours')
plt.ylabel('(CFU ml^-1)')
plt.title( 'Bacteria induced at different time points ')
plt.yscale('log')
plt.legend()
plt.show()
```

Figure 4

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

da=0.639 #Degradation rate of AHL
Nm=1.25e9  #Carrying capacity
k=0.97  #Growth rate constant
d=4.0/1000
ke=5.0 #Production rate of toxin
de=2.0 #Degradation rate of toxin
va=4.8*10**-7 #Production rate of AHL

def ahl(y,t,pr):
    N=y[0] #Cell density at given point of time
    E=y[1] #Concentration of toxin killer protein
    A=y[2] #Concentration of AHL

    dydt=np.empty(len(y))
    dydt[0]=k*N*(1-N/Nm)-(d*E*N)
    dydt[1]=ke*A-de*E
    dydt[2]=((va*pr)*N)-da*A
    return dydt

plt.figure(figsize= (10, 6))
```

```
#bacteria with pLAC promoter
initial_conditions=[1250,0,0]
t=np.linspace(0,80,100)
pr=0.67
y1=odeint(ahl, initial_conditions, t,args=(pr,))
plt.plot(t,y1[:,0],color='black',label='pLAC promoter',linewidth=1,linestyle='--')

#bacteria with promoter J23101
initial_conditions=[1250,0,0]
t=np.linspace(0,80,100)
pr=1
y2=odeint(ahl, initial_conditions, t,args=(pr,))
plt.plot(t,y2[:,0],color='red',label='J23101 promoter',linewidth=1,linestyle='--')


plt.xlabel('Hours')
plt.ylabel('(CFU ml^-1)')
plt.title( 'Bacteria with different promotors for the expression of LasR ')
plt.yscale('log')
plt.legend()
plt.show()
```