CHEME 7770 – Problem Set 1
Robert Dunleavy

1a) Modifying RNAP to account for the closed complex

We assumed that changing the W_gene1_RNAP parameter to 1.0 effectively turns on transcription of gene 1. This assumes that all available RNAP is present in the open complex. However, RNAP can exist in equilibrium between the closed and open complex,

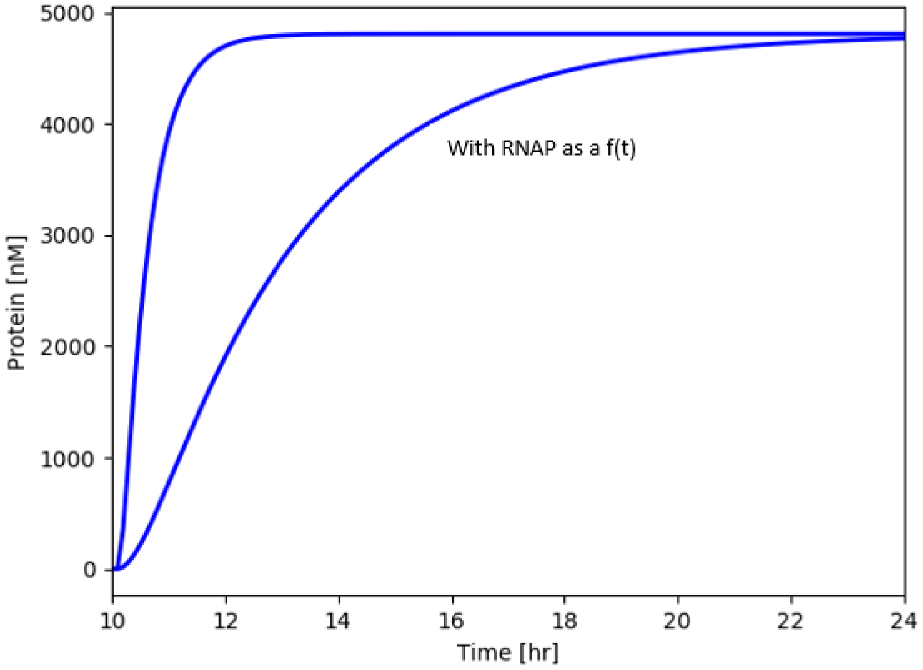
R + P <--> RP_closed <--> RP _open
Where R = Free RNAP and P= promoter concentration

The expression that [RP_open]=[RNAP_total]*(1-exp(-t/k_obs)) can be used to account for the time dependence for RNAP to equilibrate to the open position.

Changes were made to the Kinetics.jl file so that
rnapII_concentration = rnapII_concentration*(1-exp(-k_obs*(t-10)))
where k_obs was estimated to be 0.36 [1/hr] according to Figure 2.

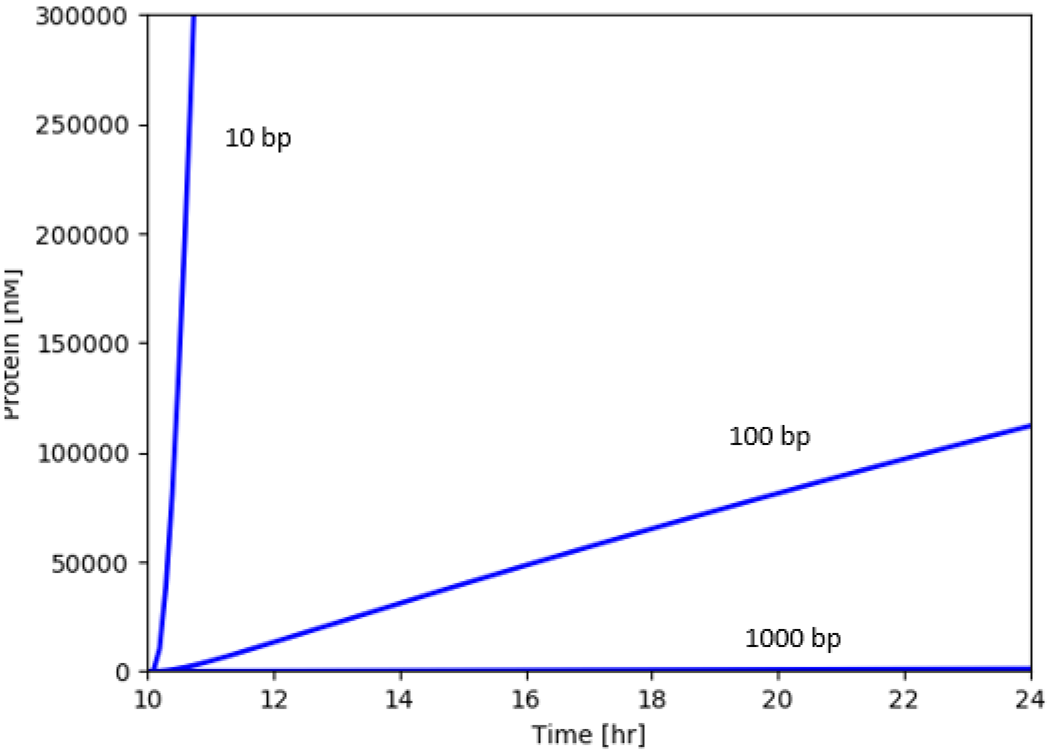
1b) Comparison of Constant versus Non-constant RNAP

Having RNAP be a function of time shifts the amount of protein produced to the right, as less RNAP is immediately available for transcription.



1c) Effect of Gene Length

Gene read length drastically affects the amount of protein produced, shorter genes are transcribed much quicker. However, in practice a 10 bp gene (perhaps 2 amino acids) would not be transcribed.



Problem 1 – Code Specifics

The Three Gene Example was modified to plot only Protein 1.

The figure in 1b) was created by running Washout.jl and commenting line 39 in the Kinetics.jl file that has RNAP as a function of time.

The figure in 1c) was prepared by manually changing the length of gene 1 in data dictionary.jl.

It would be better to produce 1b) and 1c) in a for loop instead of preparing the figures manually.

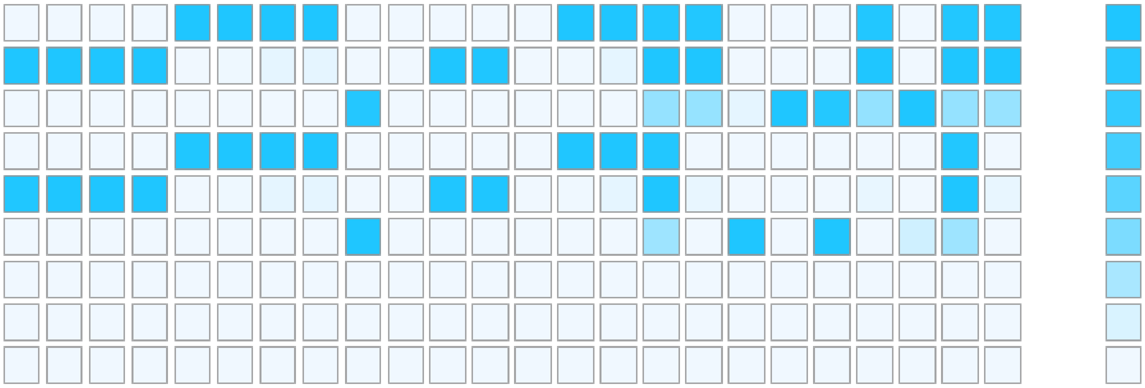
2a) Average Scaled Sensitivity Array

To get an estimate of the parameters involved an average scaled sensitivity array was calculated

	Parameter Number	Parameter Name
Protein 3	5	n_gene_3_gene_1
	6	K_gene_3_gene_1
	7	n_gene_3_gene_2
	8	K_gene_3_gene_2
	14	W_gene_3_gene_1
	15	W_gene_3_gene_2
	16	rnapII_concentration
	17	ribosome_concentration
	21	kcat_translation
	23	saturation_constant_transcription
	24	saturation_constant_transcription
Protein 2	1	n_gene_2_gene_1
	2	K_gene_2_gene_1
	3	n_gene_2_gene_3
	4	K_gene_2_gene_3
	11	W_gene_2_gene_1
	12	W_gene_2_gene_3
	16	rnapII_concentration
	17	ribosome_concentration
	21	kcat_translation
	23	saturation_constant_transcription
	24	saturation_constant_translation
Protein 1	9	W_gene_1_RNAP
	16	rnapII_concentration
	17	ribosome_concentration
	19	degradation_constant_protein
	20	kcat_transcription
	21	kcat_translation
	22	maximum_specific_growth_rate
	23	saturation_constant_transcription
	24	saturation_constant_translation

As expected, the control parameters (W, n, and K) affect their target. Global parameters (roughly 16 through 24) affect most proteins present.

(States are on the y-axis, from Protein 3,2,1 then mRNA 3,2,1 then gene 3,2,1)
(Parameters are on the x-axis, from parameter 1 (far left) to parameter 24 (far right))



2a) SVD Analysis

SVD was performed using the julia command `(U,S,V) = svd(SA,thin=false)`

From looking at V, the global parameters (rnapII_concentration, ribosome_concentration) control the steady state abundance of the three proteins.

I wasn't sure how to construct a dynamic sensitivity analysis, but I would guess that saturation_constant_transcription and saturation_constant_translation control the dynamics of protein induction.

2b) Identifiable Parameters

Changing sampling frequency from time_skip=1 to time_skip=100 did not influence the number of measurable parameters, there is probably an error in the code someplace.

Using the function `estimate_identifiable_parameters(SA,0.01)` on the full sensitivity array allows 5 parameters to be estimated (12, 16, 17, 18, 22)

When the sensitivity array only included Protein 3 and mRNA1, only 2 parameters could be estimated (17, 18).

Problem 2 - Code Specifics

The AdjDriver.jl script was modified to call an adjoint_washout_simulation.jl which is identical to the Washout.jl script expect it calls SolveAdjBalances instead of just SolveBalances.

The AdjDriver script takes about an hour to run and it solves for the states and sensitivity coefficients while changing 24 parameters.

The time data for each parameter is saved to a file in Problem_2/src/sensitivity2. The file Sensitivity.jl calculates the average sensitivity array, plots this array using PlotSensitivity.jl, performs SVD on the time averaged sensitivity array (which outputs U, S, and V), and estimates the number of identifiable parameters.

3a) Strategy for Model Discrimination

Generate Three Models for the Experimental Data

Model 1	Inducer induces Gene 1 Gene 1 activates Gene 2 Gene 2 inhibits Gene 3 Gene 1 inhibits Gene 3
Model 2	Inducer induces Gene 1 Gene 1 activates Gene 2 Gene 2 inhibits Gene 3
Model 3	Inducer induces Gene 1 Gene 1 activates Gene 2 Gene 2 inhibits Gene 3 Gene 3 inhibits Gene 2

1. Use JuGRN to generate model code for each Hypothesis
2. Create an Error function (Error.jl) that given a vector of model parameters
    a. Runs Washout.jl and returns the average value for P1, P2, and P3
    b. Calculates the error between the average values and the data given for Figure 2.
3. Create a function (Final.jl) that minimizes Error.jl by changing parameters that appear in data dictionary. Using the NLopt package, a minimum for the Error can be found.

3b) Results

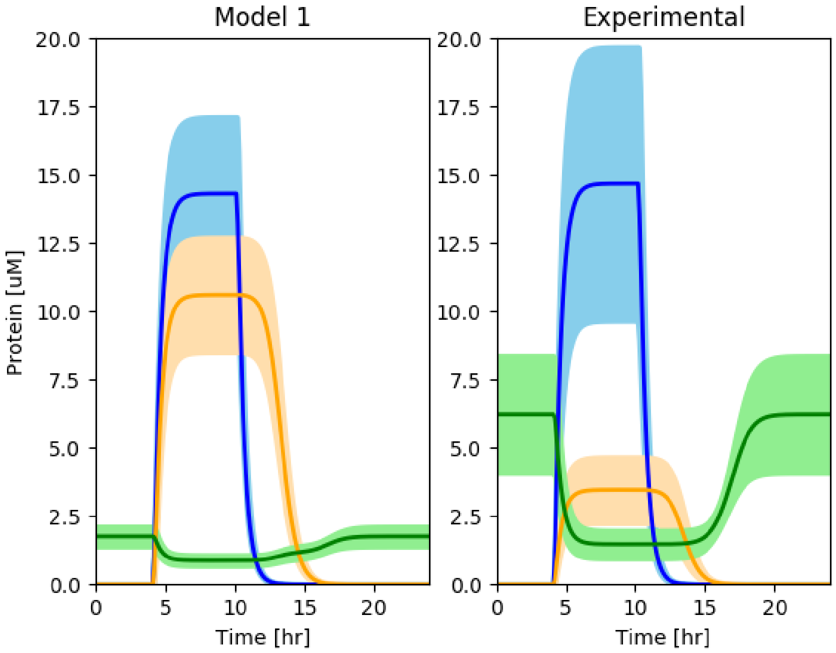
In general, the minimization function Final.jl, while able to be run, does not minimize Error.jl completely. It would be better to perform a detailed sensitivity analysis to only change parameters that affect P1, P2, and P3 the most.

Perhaps specifying relative tolerances or allowing the minimization to continue for a longer time would help.

In summary, Model 3 appears to fit the shape of the experimental data the best.

Model 1

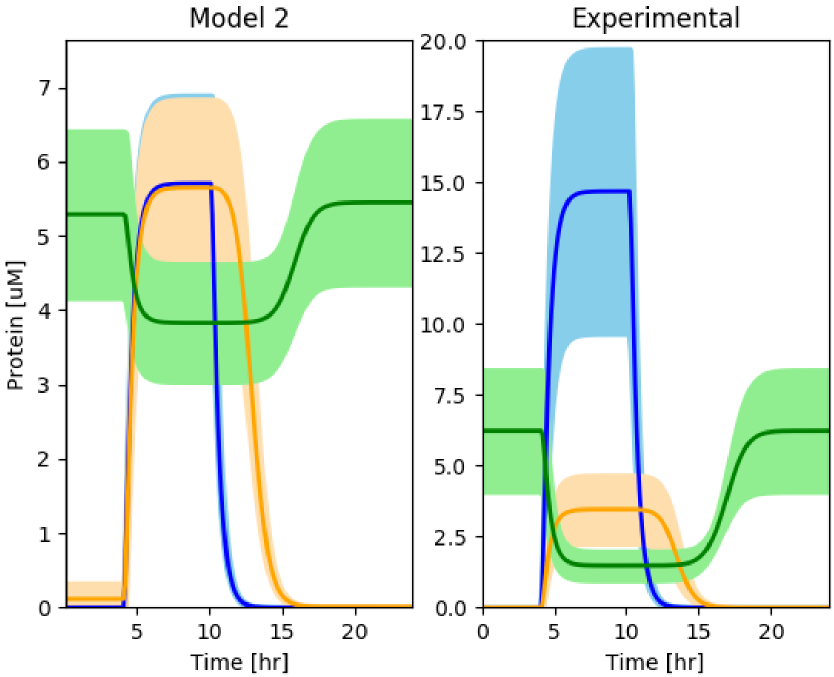
Neglecting the magnitude of the proteins produced, there is a small change in P3 when P1 falls.



Parameter	Initial	Final
n_gene_2_gene_1	1	0.933755
K_gene_2_gene_1	120	122.847
n_gene_3_gene_2	1	0.983646
K_gene_3_gene_2	120	120.401
n_gene_3_gene_1	1	0.989281
K_gene_3_gene_1	120	122.189
W_gene_1_RNAP	0	0.002175
W_gene_2_RNAP	0	0.017389
W_gene_2_gene_1	0.5	0.497263
W_gene_3_RNAP	1	0.975153
W_gene_3_gene_2	1	0.971979
W_gene_3_gene_1	1	0.964657
rnapII_concentration	10964	5307.75
ribosome_concentration	1.20E+06	5.69E+05
Error	25156	4643

Model 2

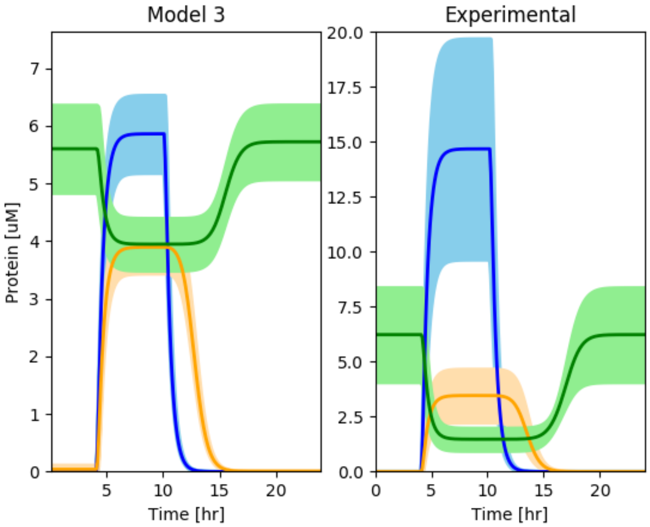
Neglecting the magnitude of the proteins produced, this appears to match the experimental values better than Model 1.



	Initial	Final
n_gene_2_gene_1	1	0.352577
K_gene_2_gene_1	120	91.9121
n_gene_3_gene_2	1	0.566859
K_gene_3_gene_2	120	109.7
W_gene_1_RNAP	0	0.086042
W_gene_2_RNAP	0	0.276489
W_gene_2_gene_1	0.1	0.231164
W_gene_3_RNAP	1	0.286561
W_gene_3_gene_2	1	0.58469
rnapII_concentration	10964	55755.6
ribosome_concentration	1.20E+06	6.19E+06
Error	8355	4839

Model 3

Neglecting the magnitude of the proteins produced, this appears to best match the experimental values.



Parameter	Initial	Final
n_gene_2_gene_1	1	1.93506
K_gene_2_gene_1	120	183.209
n_gene_2_gene_3	1	1.94307
K_gene_2_gene_3	120	183.621
n_gene_3_gene_2	1	2.21159
K_gene_3_gene_2	120	169.313
W_gene_1_RNAP	0	0.316946
W_gene_2_RNAP	0	0.311712
W_gene_2_gene_1	1	0.692489
W_gene_2_gene_3	1	0.683372
W_gene_3_RNAP	1	0.682877
W_gene_3_gene_2	1	0.682443
rnapII_concentration	10964	62491.1
ribosome_concentration	1.20E+06	6.19E+06
Error	8360	4498

Problem 3 - Code Specifics

Each model is listed under Problem_3/Model_#/src

For each model, under the source folder, run Final.jl to create the figures produced, return final parameters and the initial/final error. Depending on the constraint for maximum time provided, each Final.jl make take up to 5 minutes to run.

The difference between Error.jl and Error_final.jl is that Error.jl outputs the RMS error while Error_final.jl outputs the states for plotting given the final parameter vector.

Decrease `t` in `maxtime!(opt, t)` to have the minimization run in shorter time.