PSET1 Julia Code Sneha Kabaria-Copy1

February 5, 2019

```
In [16]: #Import Plotting Packages

         using Pkg; Pkg.add("Plots")
         using Plots
         Pkg.add("PyPlot")
```

```
 Resolving package versions...
  Updating `~/.julia/Project.toml`
 [no changes]
  Updating `~/.julia/Manifest.toml`
 [no changes]
 Resolving package versions...
  Updating `~/.julia/Project.toml`
 [no changes]
  Updating `~/.julia/Manifest.toml`
 [no changes]
```

```
In [17]: gr()
```

```
Out[17]: Plots.GRBackend()
```

```
In [18]: #PART A and B
```

```
In [19]: #Biological Constants
```

```
In [20]: #Closed to open complex = k2 = .024 s-1 (McClure Paper)
         kI = 0.024;
```

```
In [21]: # Elongation rate = e_X =42 nts/sec

         # Note for all links, they are too long to be pasted, so they are
         # put onto multiple lines. All are official cited in a separate
         # doc.

         #All citations also included in constants document

         # https://bionumbers.hms.harvard.edu/bionumber.aspx?id=108488&ve
```

```
# r=3&trm=elongation+rate+in+E.+coli&org=
e_X = 42;
Lj = 3075; #nt/gene
kej = e_X/Lj #s-1
```

Out[21]: 0.013658536585365854

```
In [24]: #Find K_x,j = McClureSlope*kI (see paper), converted to units of [mM]
         McClureSlope = 1.04*10^-3; #mM*s
         K_xj = McClureSlope*kI #mM
```

Out[24]: 2.496000000000001e-5

```
In [25]: #RNAP concentration = 30 nM
         # https://bionumbers.hms.harvard.edu/bionumber.aspx?id
         # =100194&ver=8&trm=rnap+e+coli+M&org=
         R_xt=30*10^-6; #mM
```

```
In [26]: #Find the gene concentration
         Gj_initial = 2500; # copies/cell
         Gj_2 = Gj_initial / (6.02*10^23); #mol/cell
         #Bionumbers: volume per cell = 6.7E-10 L/cell
         # https://bionumbers.hms.harvard.edu/bionumber.aspx?id
         # =108815&ver=1&trm=volume+of+e+coli+cell&org=
         Gj_3= Gj_2 / (6.7*10^-16) #mol/L = M
         Gj = Gj_3*1000 # mM
```

Out[26]: 3093.3004197832283

```
In [27]: tau = kej/kI
         #also PART B
```

Out[27]: 0.5691056910569106

```
In [28]: rxj = kej*R_xt*(Gj/(tau*K_xj+(tau+1)*Gj))
         #ANSWER TO rxj in PART A
```

Out[28]: 2.611398956088053e-7

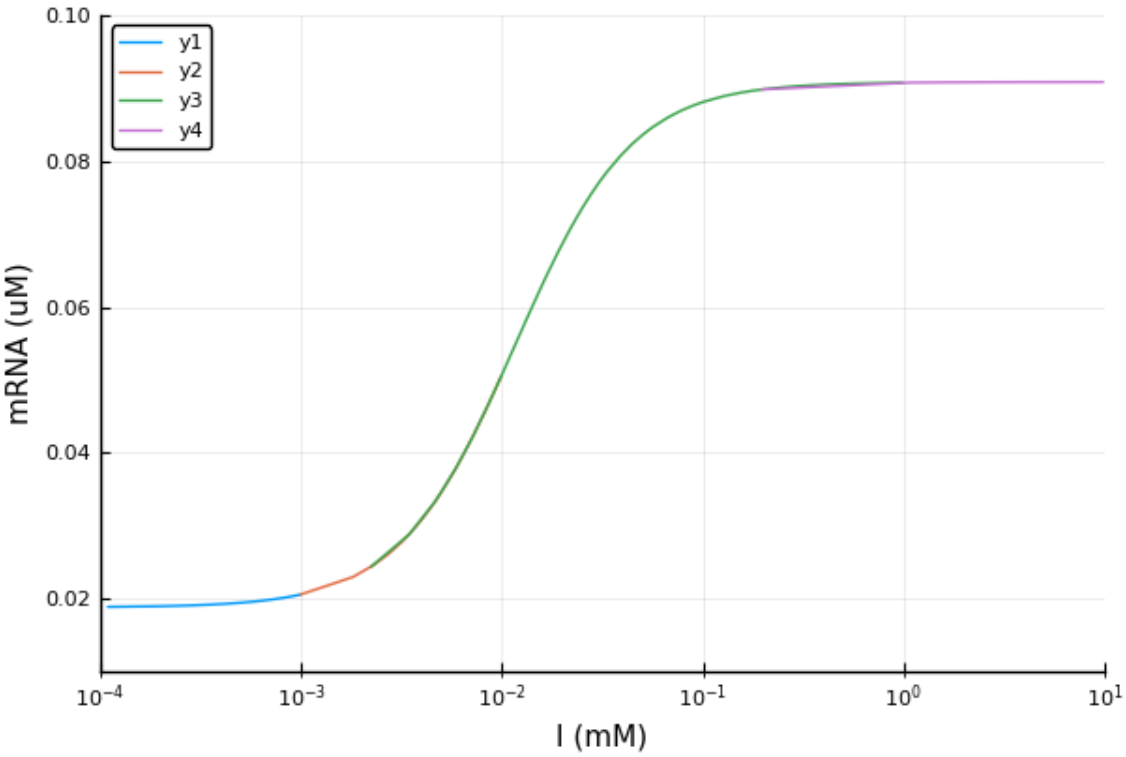
```
In [29]: #PART C
```

```
In [30]: pyplot()
```

Out[30]: Plots.PyPlotBackend()

```
In [31]: #Constants
         W1 = 0.26;
         W2 = 300;
         n = 1.5;
         K_c = 0.30;
         #find f(I)
         f(I) = (I^n)/(K_c+(I^n));
         #find u(I)
         u(I) = (W1 + W2*f(I))/(1 + W1 + W2*f(I))
```

```
Out[31]: u (generic function with 1 method)

In [32]: #growth rate, doubling time = 30 min
         grow = 30;
         #dilution is 1/grow, also convert to seconds
         B_term = 1/(30*60)

Out[32]: 0.0005555555555555556

In [33]: #global half life of mRNA = 5 min
         # https://bionumbers.hms.harvard.edu/bionumber.aspx?id=111927&ver
         # =2&trm=mrna+half+life+e+coli&org=
         halflife = 5;
         #convert to seconds
         halflife2 = 5*60;
         #convert to degradation rate, assuming first order kinetics
         kdeg = .693/halflife2

Out[33]: 0.00231

In [34]: m_j(I) = (rxj*u(I)/(kdeg+B_term)) #units of mM
         #convert to uM for the plot
         m_j_2(I) = m_j(I)*1000

Out[34]: m_j_2 (generic function with 1 method)

In [35]: #lower bound
         a = m_j_2(0.0001)

Out[35]: 0.01886208987347691

In [36]: #upper bound
         b = m_j_2(10)

Out[36]: 0.09082527844974031

In [37]: #Plotted the entire range incrementally
         #(will not plot all at once, possible due to Julia Box limitations)
         plot(m_j_2,0.0001,0.001,xaxis=:log,xlims=(0.0001,10),ylims=(1*10^-2,10*10^-2),
             xlabel = "I (mM)",ylabel="mRNA (uM)")
         plot!(m_j_2,0.0009,0.01,xaxis=:log,xlims=(0.0001,10),ylims=(1*10^-2,10*10^-2))
         plot!(m_j_2,0.001,1,xaxis=:log,xlims=(0.0001,10),ylims=(1*10^-2,10*10^-2))
         plot!(m_j_2,0.1,10,xaxis=:log,xlims=(0.0001,10),ylims=(1*10^-2,10*10^-2))

Out[37]:
```


y1
y2
y3
y4
0.10
0.08
0.06
0.04
0.02
mRNA (uM)
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
I (mM)