Una célula primitiva

close all

clear all

clc

t\_start = tic;

kr =1 ;

gr = 0.2;

kp = 0.8;

gp = 0.01;

dt = 1/1000;

t\_max = 10 ;

t =0:dt:t\_max;

rna = zeros(length(t),1);

pr = zeros(length(t),1);

n = length(t) - 1;

for i = 1:n

rna\_t = rna(i);

pr\_t = pr(i);

k1 = kr \* dt;

k2 = gr \* rna\_t \* dt;

k3 = kp \* rna\_t \* dt;

k4 = gp \* pr\_t \* dt;

event = rand;

if event <= k1

rna(i+1) = rna\_t + 1;

else

event2 = rand;

if event2 <= k2

rna(i+1) = rna\_t - 1;

else

rna(i+1) = rna\_t;

end

end

event3 = rand;

if event3 <= k3

pr(i+1) = pr\_t + 1;

else

event4 = rand;

if event4 <= k4

pr(i+1) = pr\_t - 1;

else

pr(i+1) = pr\_t;

end

end

end

time.elasped=toc(t\_start);

figure

subplot(2,1,1)

plot(t, rna)

xlabel('Minutos');

ylabel('Cantidad mRNA');

subplot(2,1,2)

plot(t, pr)

xlabel('Minutos');

ylabel('Cantidad proteÌnas');

1000 células

close all

clear all

clc

t\_start = tic; %Start of runtime measuring

%Number of cells in the population and extent of the simulation

c = 1000; %no. of cell

dt = 1/1000; %Time step of the simulation (minute)

t\_max = 1/3; %Number of iterations to be run for the simulation in minutes

%% Initializing the molecule matrices and time vector

t =0:dt:t\_max; %Vector containing time of occurrence of events

n = length(t) - 1;

rna\_pop = zeros(c, length(t));

pr\_pop = zeros(c, length(t));

for j =1:c

rna = zeros(length(t),1); %Vector containing RNA molecules at any given time

pr = zeros(length(t),1); %Vector containing protein molecules at any given time

kr = 1 ;%RNA generation rate (molecules/minute)

gr = 0.2; %RNA destruction rate (molecules/minute)

kp = 0.8; %Protein generation rate (molecules/minute)

gp = 0.01; %Protein degradation rate (molecules/minute)

for i = 1:n

rna\_t = rna(i);

pr\_t = pr(i);

k1 = kr \* dt; %Assigning of the creation rate for the calculation of the time of RNA creation events

k2 = gr \* rna\_t \* dt; %Assigning of the degradation rate for the calculation of the time of RNA degradation events

k3 = kp \* rna\_t \* dt; %Assigning of the creation rate for the calculation of the time of protein creation events

k4 = gp \* pr\_t \* dt; %Assigning of the degradation rate for the calculation of the time of protein degradation events

event = rand;

if event <= k1

rna(i+1) = rna\_t + 1;

else

event2 = rand;

if event2 <= k2

rna(i+1) = rna\_t - 1;

else

rna(i+1) = rna\_t;

end

end

event3 = rand;

if event3 <= k3

pr(i+1) = pr\_t + 1;

else

event4 = rand;

if event4 <= k4

pr(i+1) = pr\_t - 1;

else

pr(i+1) = pr\_t;

end

end

end

%Time regularization and addition to regularized time matrix

rna\_pop(j,:) = rna;

pr\_pop(j,:) = pr;

end

%% End of runtime measuring

time.elasped=toc(t\_start);

%% Finding average values and noise of the RNA and protein distributions through time in the simulations

avg\_rna = mean(rna\_pop);

sd\_rna = std(rna\_pop);

noise\_rna = sd\_rna/avg\_rna;

avg\_pr = mean(pr\_pop);

sd\_pr = std(pr\_pop);

noise\_pr = sd\_pr/avg\_pr;

% Distribution of the RNA and protein molecules at steady state

dist\_rna = rna\_pop(:,length(rna\_pop(1,:)));

dist\_pr = pr\_pop(:,length(pr\_pop(1,:)));

%% Plotting the results

figure

subplot(2,2,1)

plot(t, avg\_rna)

xlabel('Minutos');

ylabel('Promedio mRNA');%RNA molecules against time

subplot(2,2,2)

histogram(dist\_rna)

xlabel('mRNA');

ylabel('mRNA/tiempo');%RNA molecules against time

subplot(2,2,3)

plot(t, avg\_pr)

xlabel('Tiempo');

ylabel('Promedio proteÌnas');%Protein molecules against time

subplot(2,2,4)

histogram(dist\_pr)

xlabel('Promedio proteÌnas');

ylabel('ProteÌnas/tiempo');%RNA molecules against time

10000 células

%%%%% Part c of Problem1%%%%%%%%%%%%%%%%%%%

close all

clear all

clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t\_start = tic; %Start of runtime measuring

%Number of cells in the population and extent of the simulation

c = 10000; %no. of cell

dt = 1/1000; %Time step of the simulation (minute)

t\_max = 1/3; %Number of iterations to be run for the simulation in minutes

%% Initializing the molecule matrices and time vector

t =0:dt:t\_max; %Vector containing time of occurrence of events

n = length(t) - 1;

rna\_pop = zeros(c, length(t));

pr\_pop = zeros(c, length(t));

for j =1:c

rna = zeros(length(t),1); %Vector containing RNA molecules at any given time

pr = zeros(length(t),1); %Vector containing protein molecules at any given time

kr = 1 ;%RNA generation rate (molecules/minute)

gr = 1/5; %RNA destruction rate (molecules/minute)

kp = 60; %Protein generation rate (molecules/minute)

gp = 1/30; %Protein degradation rate (molecules/minute)

for i = 1:n

rna\_t = rna(i);

pr\_t = pr(i);

k1 = kr \* dt; %Assigning of the creation rate for the calculation of the time of RNA creation events

k2 = gr \* rna\_t \* dt; %Assigning of the degradation rate for the calculation of the time of RNA degradation events

k3 = kp \* rna\_t \* dt; %Assigning of the creation rate for the calculation of the time of protein creation events

k4 = gp \* pr\_t \* dt; %Assigning of the degradation rate for the calculation of the time of protein degradation events

event = rand;

if event <= k1

rna(i+1) = rna\_t + 1;

else

event2 = rand;

if event2 <= k2

rna(i+1) = rna\_t - 1;

else

rna(i+1) = rna\_t;

end

end

event3 = rand;

if event3 <= k3

pr(i+1) = pr\_t + 1;

else

event4 = rand;

if event4 <= k4

pr(i+1) = pr\_t - 1;

else

pr(i+1) = pr\_t;

end

end

end

%Time regularization and addition to regularized time matrix

rna\_pop(j,:) = rna;

pr\_pop(j,:) = pr;

end

%% End of runtime measuring

time.elasped=toc(t\_start);

%% Finding average values and noise of the RNA and protein distributions through time in the simulations

avg\_rna = mean(rna\_pop);

sd\_rna = std(rna\_pop);

noise\_rna = sd\_rna/avg\_rna;

avg\_pr = mean(pr\_pop);

sd\_pr = std(pr\_pop);

noise\_pr = sd\_pr/avg\_pr;

% Distribution of the RNA and protein molecules at steady state

dist\_rna = rna\_pop(:,length(rna\_pop(1,:)));

dist\_pr = pr\_pop(:,length(pr\_pop(1,:)));

%% Plotting the results

figure

subplot(2,2,1)

plot(t, avg\_rna)

xlabel('Minutos');

ylabel('Promedio mRNA');%RNA molecules against time

subplot(2,2,2)

histogram(dist\_rna)

xlabel('mRNA');

ylabel('mRNA/tiempo');%RNA molecules against time

%title(strcat('Average',num2str(round(time.elasped,4)),' minutes'));

subplot(2,2,3)

plot(t, avg\_pr)

xlabel('Minutos');

ylabel('ProteÌnas');%Protein molecules against time

subplot(2,2,4)

histogram(dist\_pr)

xlabel('ProteÌna');

ylabel('ProteÌnas/tiempo');%RNA molecules against time

Retroalimentación negativa

close all

clear all

clc

t\_start = tic; %Start of runtime measuring

%Number of cells in the population and extent of the simulation

c = 1000; %no. of cell

dt = 1/1000; %Time step of the simulation (minute)

t\_max = 1/5; %Number of iterations to be run for the simulation in minutes

%% Initializing the molecule matrices and time vector

t =0:dt:t\_max; %Vector containing time of occurrence of events

n = length(t) - 1;

rna\_pop = zeros(c, length(t));

pr\_pop = zeros(c, length(t));

for j =1:c

rna = zeros(length(t),1); %Vector containing RNA molecules at any given time

pr = zeros(length(t),1); %Vector containing protein molecules at any given time

kr = 1; %RNA generation rate (molecules/minute)

gr = 1/5; %RNA destruction rate (molecules/minute)

kp = 60; %Protein generation rate (molecules/minute)

gp = 1/30; %Protein degradation rate (molecules/minute)

kh = 120; %Affinity K parameter of the Hill Equation

for i = 1:n

rna\_t = rna(i);

pr\_t = pr(i);

k1 = kr/(1+(pr\_t/kh)^2) \* dt; %Assigning of the creation rate for the calculation of the time of RNA creation events

k2 = gr \* rna\_t \* dt; %Assigning of the degradation rate for the calculation of the time of RNA degradation events

k3 = kp \* rna\_t \* dt; %Assigning of the creation rate for the calculation of the time of protein creation events

k4 = gp \* pr\_t \* dt; %Assigning of the degradation rate for the calculation of the time of protein degradation events

event = rand;

if event <= k1

rna(i+1) = rna\_t + 1;

else

event2 = rand;

if event2 <= k2

rna(i+1) = rna\_t - 1;

else

rna(i+1) = rna\_t;

end

end

event3 = rand;

if event3 <= k3

pr(i+1) = pr\_t + 1;

else

event4 = rand;

if event4 <= k4

pr(i+1) = pr\_t - 1;

else

pr(i+1) = pr\_t;

end

end

end

%Time regularization and addition to regularized time matrix

rna\_pop(j,:) = rna;

pr\_pop(j,:) = pr;

end

%% End of runtime measuring

time.elasped=toc(t\_start);

%% Finding average values and noise of the RNA and protein distributions through time in the simulations

avg\_rna = mean(rna\_pop);

sd\_rna = std(rna\_pop);

noise\_rna = sd\_rna/avg\_rna;

avg\_pr = mean(pr\_pop);

sd\_pr = std(pr\_pop);

noise\_pr = sd\_pr/avg\_pr;

% Distribution of the RNA and protein molecules at steady state

dist\_rna = rna\_pop(:,length(rna\_pop(1,:)));

dist\_pr = pr\_pop(:,length(pr\_pop(1,:)));

%% Plotting the results

figure

subplot(2,2,1)

plot(t, avg\_rna)

xlabel('Minutos');

ylabel('Promedio mRNA');%RNA molecules against time

subplot(2,2,2)

histogram(dist\_rna)

xlabel('mRNA');

ylabel('mRNA/tiempo');%RNA molecules against time

subplot(2,2,3)

plot(t, avg\_pr)

xlabel('Minutos');

ylabel('Promedio proteÌnas');%Protein molecules against time

subplot(2,2,4)

histogram(dist\_pr)

xlabel('ProteÌnas');

ylabel('ProteÌnas/tiempo');%RNA molecules against time

GILLESPIE

CÈLULA GILLESPIE

%%%Daniel Charlebois - May 2011 - MATLAB v7.11 (R2010b)

%Stochastic Simulation Algorithm (SSA) - Gillespie's Direct Method (Gillespie,J.Phys.Chem.,1977).

%Default reactions and analytical solutions correspond to a simple model of

%gene expression for a single cell (Kaern et al.,Nat.Rev.Genet.,2005):

%Pro-->M-->P, with mRNA (M-->0) and protein (P-->0) decay, when expression

%is assumed to be constitutive and cell volume is constant and set to unity.

clc;

clear all;

close all;

%% initialization

%%%general simulation parameters

t=0; %start time

t\_end=500; %end time

t\_sample=0.1; %sample interval for gathering data

k=1; %counter for waitbar update

alpha=10^2; %parameter for updating waitbar (increase for shorter runtime)

%%%model parameters

%initial numbers for each chemical species

Pro = 1; %promoter

M = 5; %mRNA

P = 100; %protein

%rate constants

kM =1; %mRNA production (transcription)

kP = 0.2; %protein production (translation)

dM = 0.8; %mRNA decay

dP = 0.01; %protein decay

%%%arrays to store results

j=1; %counter for arrays

t\_array(1,t\_end/t\_sample+1)=0; t\_array(1,j)=t; %time array and initial value

Pro\_array(1,t\_end/t\_sample+1)=0; Pro\_array(1,j)=Pro; %promoter array and initial value

M\_array(1,t\_end/t\_sample+1)=0; M\_array(1,j)=M; %mRNA array and initial value

P\_array(1,t\_end/t\_sample+1)=0; P\_array(1,j)=P; %protein array and initial value

%% SSA

tic %start timing the Gillespie loop

w=waitbar(0,'running SSA...');

while t < t\_end,

%calculate rxn propensities

h = [kM\*Pro kP\*M dM\*M dP\*P];

%combined rxn hazard

h0 = sum(h);

%calculate time to next event

r1=rand;

while r1 == 0,

r1=rand;

end

t\_next = ((1/h0)\*(log(1/r1)));

%update time

t = t + t\_next;

%determine next reaction

i=1; mu=0; amu=0; r2=rand;

while amu < r2\*h0,

mu = mu + 1;

amu = amu + h(i);

i = i + 1;

end

%reactions

if mu == 1 %transcription

M = M + 1;

elseif mu == 2 %translation

P = P + 1;

elseif mu == 3 %mRNA decay

M = M - 1;

elseif mu == 4 %protein decay

P = P - 1;

end

%store/output time and species

if t >= j\*t\_sample

j=j+1;

t\_array(1,j)=j;

Pro\_array(1,j)=Pro;

M\_array(1,j)=M;

P\_array(1,j)=P;

end

%update waitbar

if t >= k\*alpha\*t\_sample

k=k+1;

waitbar(t/t\_end)

end

end

close(w)

toc

%% analytical results for model

mean\_mRNA\_theoretical = kM/dM %steady-state mean mRNA

noise\_mRNA\_theoretical = sqrt(1/mean\_mRNA\_theoretical) %steady-state noise mRNA

mean\_protein\_theoretical = (kM\*kP)/(dM\*dP) %steady-state mean protein

noise\_protein\_theoretical = sqrt((1/mean\_protein\_theoretical)+ (1/(mean\_mRNA\_theoretical\*(1+(dM/dP))))) %steady-state noise protein

%% output simulation statistics

mean\_mRNA\_simulation = mean(M\_array)

stdev\_mRNA\_simulation = std(M\_array);

noise\_mRNA\_simulation = stdev\_mRNA\_simulation/mean\_mRNA\_simulation

mean\_protein\_simulation = mean(P\_array)

stdev\_protein\_simulation = std(P\_array);

noise\_protein\_simulation = stdev\_protein\_simulation/mean\_protein\_simulation

%% plots

figure;

subplot(2,2,1);

plot(t\_array,M\_array); %mRNA time series

xlabel('SEGUNDOS');

ylabel('mRNA');

subplot(2,2,2);

hist(M\_array); %mRNA histogram

xlabel('mRNA');

ylabel('CONTEO');

subplot(2,2,3);

plot(t\_array,P\_array); %protein time series

xlabel('SEGUNDOS');

ylabel('PROTEÕNA');

subplot(2,2,4);

hist(P\_array); %protein histogram

xlabel('PROTEÕNA');

ylabel('CONTEO');

Elapsed time is 2.204784 seconds.

mean\_mRNA\_theoretical =

1.2500

noise\_mRNA\_theoretical =

0.8944

mean\_protein\_theoretical =

25

noise\_protein\_theoretical =

0.2233

mean\_mRNA\_simulation =

0.4389

noise\_mRNA\_simulation =

2.2037

mean\_protein\_simulation =

10.6675

noise\_protein\_simulation =

1.9570

>>