Project 3 – Gene Expression

Modelling mRNA transcription.

Use the program **expression.nlogo.**

Each patch on the lattice represents one cell. mRNAs are drawn as red dots close to the centre of each patch. The number of turtles on a patch represents the number of mRNAs in a cell.

Transcription of mRNAs occurs at a constant rate k. In one time step, there is a probability $k\delta t$ that a new mRNA is made in each cell. There is a finite rate of breakdown of mRNAs γ. In one time step, there is a probability $\gamma\delta t$ that each mRNA is destroyed.

Let the total number of mRNAs be M and the number of cells be N_{cells} (= number of lattice sites in this model). The program plots the mean number of mRNAs per cell $\langle m \rangle = M / N_{cells}$.

As M is large, we can treat it as a deterministic variable and write down an ODE:

$$\frac{dM}{dt} = kN_{cells} - \gamma M$$

Dividing this equation by N_{cells}, we also get the deterministic ODE for $\langle m \rangle$.

$$\frac{d\langle m \rangle}{dt} = k - \gamma \langle m \rangle .$$

From this, it can be seen that $\frac{d\langle m \rangle}{dt} = 0$ when $\langle m \rangle = k / \gamma$. This is the stationary (or equilibrium) value of $\langle m \rangle$.

Q1 (10 marks) - What is $\langle m \rangle$ as a function of time, assuming that we start with 0 mRNAs in each cell at time 0? Add the exact solution to this ODE into the model and plot it on the same graph, so that you can see the simulation follows the expected solution.

Although the mean number of mRNAs per site follows a smooth deterministic function, the number of mRNAs in any one cell follows the stochastic dynamics of births and deaths. Use the following line of code to count the turtles in one particular patch (*e.g.* the patch with co-ordinates 10 10)

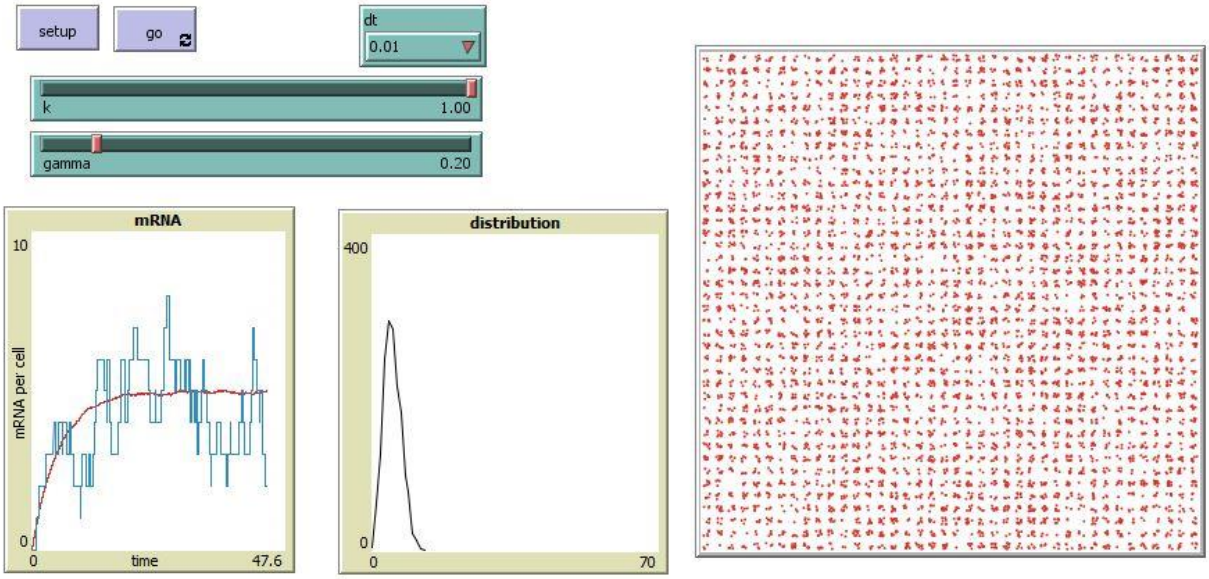
```
count turtles-on patch 10 10
```

Plot the number m on this patch on the same graph as $\langle m \rangle$ from the simulation, and the deterministic solution for $\langle m \rangle$. Comment on what is seen.

Q2 (10 marks) – Use the patches-own feature to declare a variable *m* that is owned by each patch. After each birth and death step, ask each patch to set *m* to be the number of mRNAs on that patch.
Add an extra graph and plot the distribution of *m* – *i.e.* the number of cells that have *m* mRNAs. The following lines calculate the histogram:

```
set-current-plot "distribution"
ask patches [set m count turtles-here]
histogram [m] of patches
```

After adding the features in Q1 and Q2, my program looks like this. Include a screen shot of your program to show that you have got this far.



Mathematical solution for the probability distribution of *m*

The histogram function gives N_m, the number of cells with *m* mRNAs. It is useful to define

$$P_m = \frac{N_m}{N_{cells}}.$$

This is the fraction of cells that have *m* mRNAs, or the probability that one randomly chosen cell has *m* mRNAs. P_m is a normalized probability distribution that sums up to 1.

$$\sum_{m=0}^{\infty} P_m = 1.$$

As long as N_{cells} is large enough, there should not be much fluctuation in P_m. It should be independent of the size of the lattice (N_{cells}). The stationary distribution for P_m can be calculated in the following way.

The probability that a cell increases from *m* to *m*+1 in one time step is $P_m k\delta t$.

The probability that a cell decreases from *m*+1 to *m* in one time step is $P_{m+1}\gamma(m+1)\delta t$, because each of the *m*+1 copies is destroyed at rate γ. Balancing these two rates gives

$$P_{m+1}\gamma(m+1) = kP_m,$$

$$P_{m+1} = \frac{\lambda}{m+1}P_m,$$

where we have defined $\lambda = k/\gamma$. This means that

$$P_1 = \lambda P_0, \qquad P_2 = \frac{\lambda}{2}P_1 = \frac{\lambda^2}{2}P_0, \qquad \text{and } P_m = \frac{\lambda^m}{m!}P_0 \ ,$$

where the factorial is m! = m(m-1)...3.2.1

Now we need to set P_0 so that the distribution is normalized. We note that

$\sum_{m=0}^{\infty} P_m = P_0 \sum_{m=0}^{\infty} \frac{\lambda^m}{m!} = P_0 e^{\lambda} = 1$. Therefore we need to set $P_0 = e^{-\lambda}$.

The final solution is:

$$P_m = \frac{\lambda^m}{m!}e^{-\lambda}.$$

This is called a Poisson distribution. A Poisson distribution is defined by a single parameter λ, which is the mean of the distribution:

$$\langle m \rangle = \sum_{m=0}^{\infty} mP_m = \lambda .$$

Optional - *Check the math on this! How would you prove it? You can also use Excel to calculate the P_m distribution for different values of λ and show that it is normalized to 1 and the mean is λ.*

The mean of m^2 is $\langle m^2 \rangle = \sum_{m=0}^{\infty} m^2 P_m$

The variance of the distribution is $Var(m) = \langle m^2 \rangle - \langle m \rangle^2$. *For a Poisson distribution, it turns out that* $Var(m) = \lambda$. *Can you prove this? In other words, the variance of a Poisson distribution is equal to its mean:* $Var(m) = \langle m \rangle$. *So the standard deviation is* $SD(m) = \sqrt{\langle m \rangle}$. *This means that the width of the distribution gets narrower as a fraction of the mean when the mean increases:* $\frac{SD(m)}{\langle m \rangle} = \frac{1}{\sqrt{\langle m \rangle}}$.

The biological point is that if the transcription rate is fast compared to the breakdown rate, then the mean number $\langle m \rangle = \lambda$ *is large, and fluctuations in m are small. All cells should have a similar number of mRNAs, close to λ. In reality, this is only true for the most highly expressed genes where transcription is 'on' continuously. Most genes have small transcription rates. In this case λ is small, and there are significant fluctuations among cells.*

Q3 (10 marks) – Export the histogram N_m into Excel, and convert it to the probability distribution P_m. Plot the Poisson distribution on the same graph, and show that the theory agrees with the simulation. Do this for three cases:

(i) $k = 1$, $\gamma = 0.2$, $\lambda = 5$
(ii) $k = 1$, $\gamma = 0.05$, $\lambda = 20$
(iii) $k = 1$, $\lambda = 1$, $\lambda = 1$

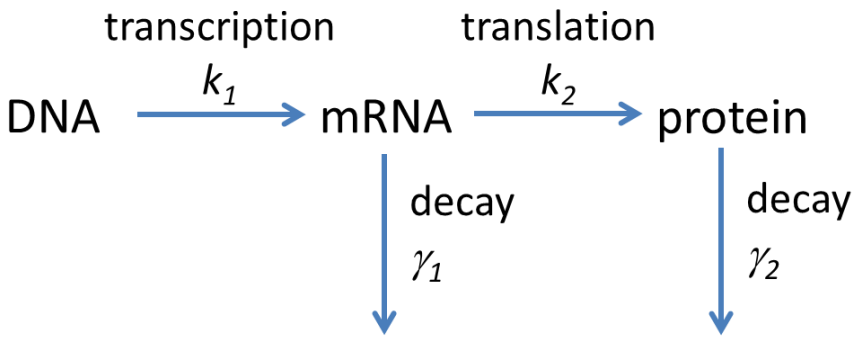
Comment on the shapes of these three different Poisson distributions.

Modelling expression of mRNAs *and* proteins.

Read the paper by Taniguchi *et al.*

They are able to label an mRNA and the corresponding protein with different fluorophores, and to measure the level of both mRNA and protein at the same time in individual cells. They are interested in the fluctuations in the numbers of mRNAs and proteins over time within one cell and the variation between different cells at any one time.
They use the following parameters:

k_1 = transcription rate (mRNA production rate per cell)
k_2 = translation rate (protein production rate per mRNA)
γ_1 = mRNA decay rate
γ_2 = protein decay rate



We will now build a model of what is measured in the Taniguchi *et al.* paper. Start with your latest version of the expression program and save it as a new file called **mrna+protein.nlogo**
Let m be the number of mRNA in one cell and n be the number of proteins. The mean values of these quantities satisfy the following ODEs

$$\frac{d\langle m\rangle}{dt} = k_1 - \gamma_1\langle m\rangle$$

$$\frac{d\langle n\rangle}{dt} = k_2\langle m\rangle - \gamma_2\langle n\rangle$$

The stationary values are

$$\langle m \rangle = \frac{k_1}{\gamma_1} \equiv \lambda_1$$

$$\langle n \rangle = \frac{k_2 \langle m \rangle}{\gamma_2} = \frac{k_2 k_1}{\gamma_2 \gamma_1} \equiv \lambda_2$$

We already know that the distribution of *m* is a Poisson distribution with mean λ_1:

$$P_m = \frac{\lambda_1^m}{m!} \exp(-\lambda_1).$$

We might guess that the distribution of *n* will be a Poisson distribution with a mean λ_2:

$$P_n = \frac{\lambda_2^n}{n!} \exp(-\lambda_2).$$

However, this guess is wrong, as Taniguchi *et al.* show, and as we will show in our model. Poisson distributions occur when there are independent random events. The transcription events that create the mRNAs are random in our model, *i.e.* they are stochastic events that occur with a small probability at each time step. However, the translation events are not random in the same way. A burst of protein production occurs whenever an mRNA is present, and stops whenever an mRNA decays. The distribution of proteins is more complex than a Poisson distribution because of these bursts of translation.

Let's use the following parameters as an example.

$k_1 = 1$, $\gamma_1 = 0.5$ - therefore $\lambda_1 = 2$

$k_2 = 10$, $\gamma_2 = 0.1$ - therefore $\lambda_2 = 200$

Translation is fast when mRNAs are present, so the mean number of proteins is much larger than the mean number of mRNAs. The mean lifetime of an mRNA is shorter than the mean lifetime of a protein. For these parameters $1/\gamma_1 = 2$ time units and $1/\gamma_2 = 10$ time units. The mRNAs come and go frequently within the lifetime of a protein. The number of proteins present depends on how many mRNAs there were in the past, not just on how many there are now.

As the number of proteins is likely to be large in this model, we will treat the protein number *n* as a variable that is a property of a cell, rather than creating turtles to represent each protein. We will treat the mRNAs as individual turtles, as we already did so far. Declare *n* and *m* as belonging to the patches - patches-own [nmrna nprot]. Set *n* and *m* to be zero on all patches initially.

There are four stages in the birth-death routine.

1. Birth of mRNAs. Ask each patch to create a new mRNA with probability $k_1 \delta t$.
2. Birth of proteins. The mean number of proteins translated from one mRNA is $k_2 \delta t$. If k_2 is large, this number may be more than 1. The actual number of new proteins generated is an integer *p* drawn from a Poisson distribution with mean $k_2 \delta t$. Thus for each turtle, we generate a

random number and add it to the number of proteins already on that patch. This is simple in Netlogo:

```
ask-turtles [set nprot (nprot + random-poisson (k2 * dt))]
```

Note that each turtle does this, but nprot is a property of the patch on which the turtle sits. So if there is more than one turtle on a patch (*i.e.* more than one mRNA in the same cell), all the proteins go onto the same patch.

3. Death of mRNAs. Ask each turtle to die with probability $\gamma_1 \delta t$.

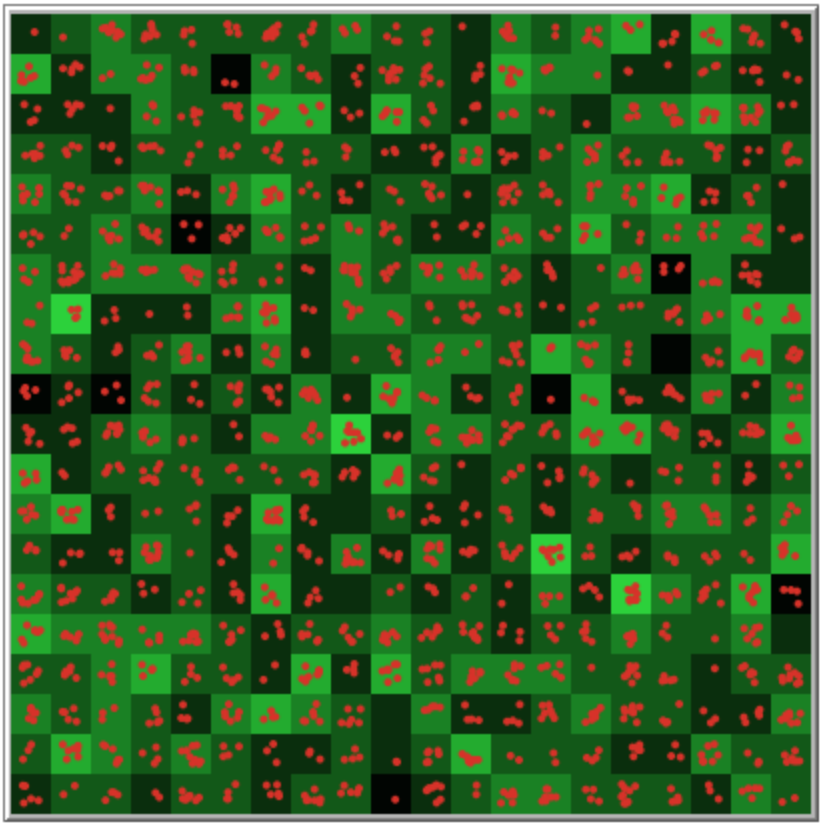
4. Death of proteins. The mean number of proteins that are destroyed in one time step is $n\gamma_2 \delta t$, where n is the current number of proteins. The number of proteins destroyed p is an integer drawn from a distribution with mean $n\gamma_2 \delta t$. So in the program, we write

```
ask-patches [
        set nprot (nprot – random-poisson (nprot * gamma2 * dt))
        if nprot < 0 [set nprot 0]
]
```

Note that the actual distribution is binomial, not Poisson, but these are close, and there is no built in binomial distribution in Netlogo. So we will use Poisson. For this reason, we need the line to check whether we have accidentally destroyed more proteins than were there in the first place. In this case, we set nprot to 0. We would not need this line if we used the binomial distribution (There is more information on this in the Optional Box on the next page.)

Now add the following features to visualize the results of the model.

- Measure the mean value of n for all patches and plot it on the same graph as the mean m. As $n >> m$, it is convenient to scale each one by its theoretical mean. You can plot $\langle m \rangle / \lambda_1$ and $\langle n \rangle / \lambda_2$. These quantities should both tend to 1 when the program reaches equilibrium. This is a good check to see if you did it right.
- Add a new plot to show the histogram of n. The range of n is different to the range of m so it is difficult to plot them on the same graph.
- Visualize the level of protein fluorescence by colouring the patches according to the number of proteins. The colors 60-65 give a range from black to bright green. So figure out a way to set pcolor in the range 60 (for low n) to 65 (for high n). If the lattice is small (20 x 20 in the example below), it is possible to see individual patches lighting up and going dark, and individual mRNAs (red dots) appearing and disappearing.



Example of a 20 x 20 lattice using a green colour scheme to show protein concentration.

Q4 (10 marks) – Include the Netlogo code of your proteins+mrna program and a screen shot of the result in order to show that you have added all the features of the model described on pages 4-6.

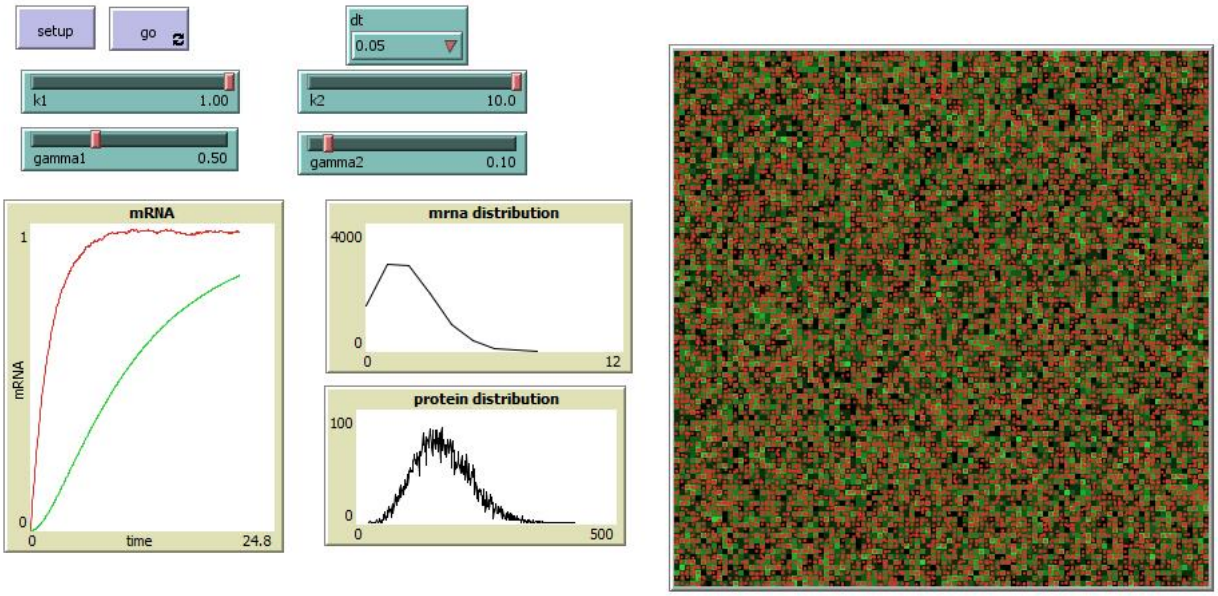
Optional Box – No marks for this bit.

Suppose there are n = 20 proteins, $\gamma_2 = 0.1$ *and* $\delta t = 0.05$*. The probability each protein is destroyed is 0.005. The mean number of proteins destroyed is 0.1.* P_p *is the probability that p proteins are destroyed. If we use a Poisson distribution,* $P_p = \dfrac{(0.1)^p \exp(-0.1)}{p!}$

If we use a binomial distribution,

$$P_p = (0.005)^p (0.995)^{20-p} \frac{20!}{p!(20-p)!}$$

Plot both of these distributions using Excel. There are built-in functions POISSON.DIST and BINOM.DIST in Excel that will calculate this. They are almost the same, but p has a very small probability of being greater than 20 in the Poisson case, whereas it cannot be greater than 20 in the binomial case. So it is OK to use the Poisson function as an approximation in our program, but we need to check for the rare event where the number of proteins becomes negative by mistake.



Screenshot for a 100 x 100 lattice

Q5 (10 marks) – Once the program is working, make the lattice size large (100 x 100 patches) so that you get good statistics on the distributions of *m* and *n.*
Run the program with the example parameters

$k_1 = 1$, $\gamma_1 = 0.5$, $k_2 = 10$, $\gamma_2 = 0.1$

Once the program reaches equilibrium, stop it, and export the histograms of *m* and *n* to Excel. Convert the histograms to probability distributions by dividing by the total number of patches (10000). Plot the measured P_m together with the Poisson distribution with mean $\lambda_1 = 2$. Show that it is a good fit. Plot the measured P_n together with the Poisson distribution with mean $\lambda_2 = 200$. This should not be a good fit. The Poisson distribution is a narrow peak close to the mean, whereas the measured distribution is much broader.

Q6 (10 marks) – In the Taniguchi paper, it explains that an approximate theory for the P_n distribution can be obtained by assuming that a burst of protein production occurs every time a mRNA is produced. The result depends on the parameters $a = k_1 / \gamma_2$ (the mean number of mRNAs produced during the lifetime of a protein) and $b = k_2 / \gamma_1$ the mean number of proteins translated from one mRNA during its lifetime). The formula for the distribution is

$$P_n = \frac{n^a \exp(-n/b)}{\Gamma(a) b^a}.$$

This is called a gamma distribution. There is a built in function GAMMA.DIST in Excel. Add the gamma distribution onto the graph with the measured P_n and the Poisson distribution. Show that this fit is quite good (although it is still an approximate theory, which is not an exact solution

of the model that we simulated). Try a few alternative sets of parameters to see how well the theory works in different cases.

Is there a correlation between mRNA level and protein level?

The levels of proteins and mRNAs differ by many orders of magnitude among the different genes in an organism. Biologists often talk about 'expression level' of a gene assuming that mRNA and proteins vary in proportion. Our theory says that, on average the mean *n* is proprtional to the mean *m*.

$$\langle n \rangle = \frac{k_2 \langle m \rangle}{\gamma_2}$$

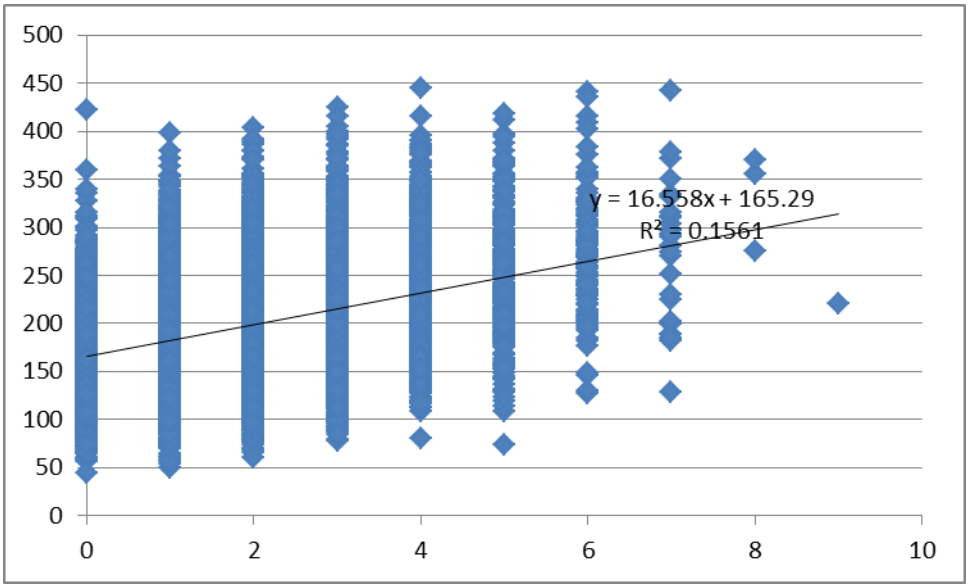
This would be a straight line, if k_2 and γ_2 were the same for every gene. In reality, there is a lot of variation in these 'constants', so we expect a scattered relationship between *n* and *m*, but with a clear linear trend. This is what Taniguchi *et al.* see in Fig. 3C

They also measure a correlation between *n* and *m* across individual cells for one gene in Fig 4 They show that there is no correlation in this case. In other words, individual cells which have more mRNAs at a particular time do not necessarily have more proteins at the same time. Thus there is correlation between average *n* and *m* across genes, even though there is no correlation between *n* and *m* across individual cells for one gene. These two things are different.

Q7 (10 marks) – The following piece of Netlogo code will print out the *m* and *n* values for all the patches into the command centre at the bottom of the screen.

```
ask patches [
        type nmrna
        type " "
        print nprot
]
```

Add a button to your program called 'correlations' that will execute this code when you click on the button. Run the program until it reaches equilibrium, stop it, and then click the correlations button. You can export the text in the command centre to Excel and plot the scatter plot of *n* and *m*. For the example parameters we have been using, I get something like this:



Draw this scatter plot for your data and calculate the correlation coefficient. There is a Trend-line feature on the Excel graph that will do this.

In my case, R^2 = 0.156 – which is a weak positive correlation.

Additional Thoughts

The fact that the correlation in the experimental data is even weaker than this (essentially zero correlation) suggests that we have missed out a few sources of variation between cells that would decrease the correlation even further. I am guessing the following:

- The paper says the distribution of the mRNA is broader than Poisson, whereas it is Poisson in our case. Thus there might be bursts of mRNA transcription as well as bursts of translation from each mRNA. How would you model this factor?
- The paper talks about extrinsic sources of noise as well as intrinsic sources of noise. For example, variation in the number of ribosomes per cell, or the number of RNA polymerases, or transcription factors. How could you add some of these things to the model?
- Our model ignores cell division. In reality, cells are growing and dividing at the same time as the mRNAs and proteins are being produced and destroyed. How could you model this?

Total Marks 70 for this sheet.

FILES TO SUBMIT - Please submit one pdf file with the graphs and answers to questions, and one Netlogo file with your program that answers Q4.