MCB137L/237L: Physical Biology of the Cell
Spring 2024
Homework 8
(Due 3/19/24 at 2:00pm)

Hernan G. Garcia

"How can the events in *space and time* which take place within the spatial boundary of a living organism be accounted for by physics and chemistry?" - Erwin Schrödinger **What is Life?**

1 What Living Organisms Must Fight

In class we talked about how systems will tend towards the state of maximum entropy. In this problem, you are going to flesh out the details of the calculations leading to the graphs we showed in class and will provide your own graphs.

(a) Equilibrium with respect to mass transport. Consider a system partitioned equally into two parts, each of which contains Ω lattice sites. We want to write the total entropy as $S_{tot}(L) = S_L(L) + S_R(L_{tot} - L)$. Show that these contributions to the entropy can be written as

$$S_L(L) = k_B \log \frac{\Omega^L}{L!} \tag{1}$$

for the left side and

$$S_R(L_{tot} - L) = k_B \log \frac{\Omega^{L_{tot}-L}}{(L_{tot} - L)!} \tag{2}$$

for the right side. Using the Stirling approximation, derive the expression

$$S_{tot}(L) = -k_B L_{tot} \left[\frac{L}{L_{tot}} \log \frac{L}{L_{tot}} + \left(1 - \frac{L}{L_{tot}}\right) \log \left(1 - \frac{L}{L_{tot}}\right) + \left(\log \frac{L_{tot}}{\Omega} - 1\right) \right] \tag{3}$$

for the total entropy. Plot the entropy of the left part, the right part and the total entropy as a function of the number of ligands in the left side of the container which can run from $L = 0$ to $L = L_{tot}$. To make this plot, you will need to assume a certain number of lattice sites. Imagine a container with $\Omega = 10^9$ lattice sites. If each such lattice site has a volume

of 1 nm^3, then the total volume of each side is 1 μm^3.

(b) We next consider the case in which the partition between the two sides is mobile. In this case, we are interested in how the entropy on the left side and the right side play against each other, conspiring to give a total entropy of the form

$$S_{tot}(x) = S_L(x) + S_R(x), \tag{4}$$

where x is the label used to characterize the position of the interface. As usual, the entropy is given by the Boltzmann formula which in this case takes the form

$$S_L(x) = k_B \log W_L(x) \tag{5}$$

and

$$S_R(x) = k_B \log W_R(x). \tag{6}$$

To make progress, we now need to reckon the number of states as a function of the position x of the partition. When the partition is at the midpoint, each of the subcompartments has a volume V. The volume swept out by the motion of the partition by a distance x is xA, where A is the cross-sectional area of that partition. As a result, show that the number of lattice sites added or subtracted due to the motion of the partition is xA/v where v is the volume corresponding to a single lattice site, leading to the results

$$W_L(x) = \frac{(\frac{V+xA}{v})^{L_L}}{L_L!}, \tag{7}$$

and

$$W_R(x) = \frac{(\frac{V-xA}{v})^{L_R}}{L_R!}. \tag{8}$$

Use these results to show that

$$S_{tot}(x) = k_B L_L \log \frac{V+xA}{v} - k_B \log L_L! + k_B L_R \log \frac{V-xA}{v} - k_B \log L_R!, \tag{9}$$

and make a plot of the resulting entropy of the two sides and the total entropy as a function of the position of the partition x.

2 Mutation correlation and physical proximity on the gene

Do problem 4.4 from PBoC2 shown in Figure 1. You might find it useful to read section "Flies and the Rise of Modern Genetics" starting on page 170 of PBoC2.

• 4.4 Mutation correlation and physical proximity on the gene

In Section 4.6.1, we briefly described Sturtevant's analysis of mutant flies that culminated in the generation of the first chromosome map. In Table 4.2, we show the crossover data associated with the different mutations that he used to draw the map. A crossover refers to a chromosomal rearrangement in which parts of two chromosomes exchange DNA. An illustration of the process is shown in Figure 4.26. The six factors looked at by Sturtevant are B, C, O, P, R, and M. Flies recessive in B, the black factor, have a yellow body color. Factors C and O are completely linked, they always go together and flies recessive in both of these factors have white eyes. A fly recessive in factor P has vermilion eyes instead of the ordinary red eyes. Finally, flies recessive in R have rudimentary wings and those recessive in M have miniature wings. For example, the fraction of flies that presented a crossover of the B and P factors is denoted as BP. Assume that the frequency of recombination is proportional to the distance between loci on the chromosome.

Reproduce Sturtevant's conclusions by drawing your own map using the first seven data points from Table 4.2.

Keep in mind that shorter "distances" are more reliable than longer ones because the latter are more prone to double crossings. Are distances additive? For example, can you predict the distance between B and P from looking at the distances B(C,O) and (C,O)P? What is the interpretation of the two last data points from Table 4.2?

Table 4.2: Fraction of crossovers of six sex-linked factors in *Drosophila*. (Adapted from A. H. Sturtevant, *J. Exp. Zool.* 14:43, 1913.)

Factors	Fraction of crossovers
BR	115/324
B(C,O)	214/21736
(C,O)P	471/1584
(C,O)R	2062/6116
(C,O)M	406/898
PR	17/573
PM	109/458
BP	1464/4551
BM	260/693

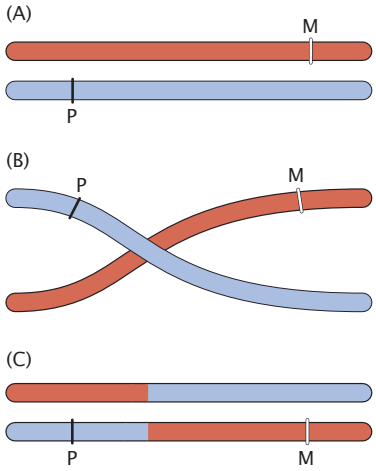

Figure 4.26: Crossing over of chromosomes. (A) Chromosomes before crossing over showing two loci labeled P and M. (B) Illustration of the crossing-over event. (C) Chromosomes after crossover.

Figure 1: Problem 4.4 from PBoC.

3 Counting Proteins with Partitioning Statistics

One of the great challenges in quantitative cell biology is to be able to turn the fluorescence values obtained from fusions to proteins to an actual absolute number of proteins. While there are many ways to "calibrate" such measurements using standards of a known concentration, in this problem, we will explore how we can use bacterial cell division, pure thought and the binomial distribution in order to calibrate a fluorescent protein.

(a) Begin by reading the paper by Rosenfeld *et al.* entitled "Gene Regulation at the Single-Cell Level" (posted on the website with the homework) and write a one paragraph commentary on the paper with special reference to how they used the binomial partitioning as a way to count repressor proteins. What is the experiment they did and what were they trying to

learn?

In the rest of the problem we work out for ourselves the ideas about binomial partitioning introduced in the Rosenfeld *et al.* paper in order to consider the concentration of proteins as a function of time in dividing cells. In particular, the point of this problem is to work out the concentration of protein given that we start with a single parental cell that has N copies of this protein. In the Rosenfeld experiment, at some point while the culture is growing, the production of the protein is stopped by providing a chemical in the medium and then the number of copies per cell is reduced as a result of dilution as the cells divide.

Interestingly, this problem opens the door to one of the most important themes in physics, namely, that of fluctuations. In particular, as the cells divide from one generation to the next, each daughter does not really get $N/2$ copies of the protein since the dilution effect is a stochastic process. Rather the partitioning of the N proteins into daughter cells during division follows the binomial distribution. Analyzing these fluctuations can actually lead to a quantification of the number of copies of a protein in a cell.

(b) We think of the N copies of the protein as being divided between the two daughters with N_1 going to daughter 1 and $N - N_1$ going to daughter 2. Explain how the probability of N_1 proteins going to daughter cell one is given by the binomial distribution

$$P(N_1, N) = \binom{N}{N_1} p^{N_1} q^{N-N_1}, \tag{10}$$

where the probability of a protein going to daughter cell 1 is p, and the probability of a one protein going to daughter 2 is $q = 1 - p$. For your explanation you can choose to show a formal mathematical derivation, or qualitatively walk us through the meaning of each term in the equation. Remember that, while for most of the course we could use the "stadium seating" approximation to think about how to place N_1 spectators in N seats, here N and N_1 are of comparable magnitude. This situation, which already encountered in the context of the DNA entropic spring, calls for the binomial coefficient $\binom{N}{N_1}$.

We can also calculate the mean of the probability distribution (also called the first moment of the distribution) by invoking a cool trick using the derivative with respect to p

$$\langle N_1 \rangle = \sum_{N_1=0}^{N} N_1 \binom{N}{N_1} p^{N_1} q^{N-N_1} = p \frac{\partial}{\partial p} \sum_{N_1=0}^{N} \binom{N}{N_1} p^{N_1} q^{N-N_1}. \tag{11}$$

This equation can be rewritten as

$$\langle N_1 \rangle = p \frac{\partial}{\partial p} \left((p+q)^N \right) = p N_{mother} (p+q)^{N-1}, \tag{12}$$

where we made use of the fact that

$$\sum_{N_1=0}^{N} P(N_1, N) = (p+q)^N. \tag{13}$$

Using $p + q = 1$, Equation 12 leads to

$$\langle N_1 \rangle = pN. \tag{14}$$

(c) Work out the expected averaged fluctuations squared in the partitioning process after each division by noting that the averaged fluctuations can be written as $\langle (N_1 - N_2)^2 \rangle$, where N_1 and N_2 are the number of proteins that end up in daughter cells 1 and 2, respectively. Show that, if $p = q = 0.5$, the partitioning error is given by $\langle (N_1 - N_2)^2 \rangle = N$. To make this possible, use the derivative trick twice such that

$$\langle N_1^2 \rangle = \sum_{N_1=0}^{N} N_1^2 \binom{N}{N_1} p^{N_1} q^{N-N_1} = p\frac{\partial}{\partial p}\left[p\frac{\partial}{\partial p}\left(\sum_{N_1=0}^{N} \binom{N}{N_1} p^{N_1} q^{N-N_1} \right) \right] \tag{15}$$

as well as the result $\langle N_1 \rangle = pN$ described above. In addition, use the fact that $N = N_1 + N_2$, in order to calculate the average partioning error as

$$\langle (N_1 - N_2)^2 \rangle = \langle [N_1 - (N - N_1)]^2 \rangle = \langle (2N_1 - N)^2 \rangle. \tag{16}$$

Remember that $\langle N \rangle = N$, as N is a constant in our problem.

(d) Next, look at the Rosenfeld paper and explain how measuring fluorescence variations can be used to calibrate the exact number of copies of the fluorescent protein in a cell. Specifically, assume that the fluorescence intensity in each cell can be written as $I = \alpha N$, where α is an as-yet unknown calibration factor and N the number of proteins in the cell. Explain what this equation means and why you think it is justified. Derive an expression relating I_1, I_2 and I_{tot} using the result of part (c). Make a qualitative schematic showing a plot of $\langle (I_1 - I_2)^2 \rangle$ versus I_{tot} and explain how to get the calibration factor α from this plot. Note that we're asking to draw up an explanation, not to actually make a plot with Python..

(e) Now we are going to repeat the Rosenfeld experiment numerically in order to *fit* the calibration factor. Consider a fluorescent protein such that the calibration factor between the intensity and the number of fluorophores is 50, that is $I = 50N$. Generate intensity data by choosing $N_1 + N_2 = 10, 50, 100, 1000$ and 5000 and for each case, "partition" the proteins from the mother cell to the two daughters 100 times (i.e. as if you are looking at 100 mother cells divide for each choice of the protein copy number). To make this possible, flip a coin for each molecule in order to decide whether the molecule is going to daughter cell 1 or 2 (and remember how we've done something similar to this earlier in the course when modeling diffision as coin flips). Then, make a plot of the resulting $\langle (I_1 - I_2)^2 \rangle$ vs I_{tot} just as we did analytically in the previous problem. What I mean is that you need to make a plot of all of your simulation results. Then, do a fit to your "data" using a numpy function (see the note below) and see how well you recover the calibration factor that you actually put in by hand. Plot the fit on the same graph as all of the "data".

Note: You can use `numpy.polyfit` to perform a linear fit to your "data" using the syntax `numpy.polyfit(x, y, deg)` where `x` is the data x-coordinate, `y` is the data y-coordinate, and `deg` is the degree of the polynomial you'd like to fit to your data (for instance, you would

use `deg = 1` for a linear fit). You can also use `numpy.linalg.lstsq` if you'd rather phrase the problem as a matrix equation (this is reasonably simple to do as well, and an example of a linear fit performed using this function is provided in the Numpy documentation linked to above).