Phys847/492 **Problem Set #1** **Due: Thurs Oct 2 in class**

Random Polymers: Viruses (From "Mechanics of the Cell" by Dave Boal)

The λ-phage virus that infects bacteria has a genome that is 97000 bp long.

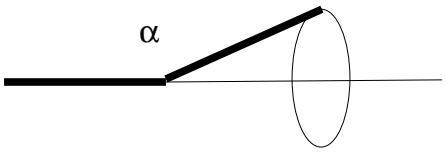
a) Find the length of its genome in nanometers. How does this compare with DNA's persistence length?

b) What is the volume of the viral DNA, and what is the radius of the smallest spherical shell that it could be packed into? How does this compare with DNA's persistence length?

c) Once released into the bacterial cell, what is the root-mean-squared distance that it expands too? Is this smaller than the size of a typical bacterium?

Random Polymers: Bond angles (From "Mechanics of the Cell" by Dave Boal)

Here we're going to look at the end-to-end distance for a polymer that has fixed angles between its monomers. Consider a polymer such as linear alkane that has bond angles between successive carbon atoms that is fixed at a value, α. The bonds are free to rotate around each other (see figure).



The length and orientation of the bond between atom *i* and atom *i+1* defines a bond vector $\boldsymbol{b}_i$. Assume all bond lengths are the same and that remote bonds can intersect.

a) Show that the average projection of $\boldsymbol{b}_{i+k}$ on $\boldsymbol{b}_i$ is $\langle \boldsymbol{b}_i \cdot \boldsymbol{b}_{i+k} \rangle = b^2(-\cos\alpha)^k$.

b) Write the average end-to-end distance $\langle r_{ee}^2 \rangle$ in terms of $\langle \boldsymbol{b}_i \cdot \boldsymbol{b}_j \rangle$ and show that

$$\langle r_{ee}^2 \rangle = N[1+(2-2/N)(-\cos\alpha)+(2-4/N)(-\cos\alpha)+\ldots]$$

c) Show that in the large N limit, this becomes

$$\langle r_{ee}^2 \rangle = Nb^2(1-\cos\alpha)/(1+\cos\alpha)$$

d) In this limit, and with α = 109.5°, what would be the effective bond length for an equivalent ideal random chain?

Protein Folding:

Let us assume that 16mer HP sequences can only fold into compact structures that completely fill a 4x4 lattice. In this problem you are going to try and compute the designability of 4x4 compact structures. This problem will involve doing several computations.

Model:

The sequence of a protein is a particular ordering of H's and P's (e.g. HHPPHPHPP...) with $h_H = -1$ and $h_P = 0$ kT. A compact structure on a lattice consists of core sites and surface sites given by $s_C = 1$ and $s_S = 0$ where C represents a buried core site and S represents a surface site. The structure can be described by a surface exposure sequence that gives the sequence of surface exposures (see Figure 1).

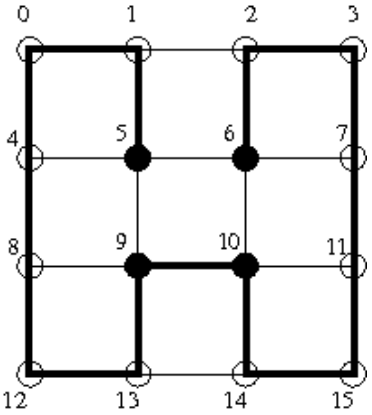


Figure 1: A compact 4x4 structure. Black circles represent the 4 core sites (s=1) with white circles representing the surface sites (s=0). If the structure starts at site 5, the path is given by the sequence: 5,1,0,4,8,12,13,9,10,14,15,11,7,3,2,6. Translated into a surface exposure sequence, the corresponding structure is s = 1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,1.

The energy of a sequence on a given structure in the Solvation model is given by $E = \sum_i h_i s_i$. The structure that has the lowest energy is the ground state structure. Note: if a sequence has more than one structure with the same lowest energy, then there is no unique ground state structure.

Structure Generation:

The challenge is to try and enumerate systematically all self-avoiding walks (SAWs) on a fixed size lattice. If the SAW visits every site on the lattice this is known as a Hamiltonian path. Generating all Hamiltonian paths on a lattice is a challenging problem. Fortunately there is some code that will randomly sample paths – i.e. every time the

algorithm is run it generates a random compact structure. Your task is to run the script for a 4x4 grid, generating a list of randomly generated structures and then determine if you have enumerated all possible structures. Here's what you need to do:

a) Go to the course webpage and run the "Hamiltonian Paths Generator Script" using a grid size = 4. You may need to generate lots of structures to guarantee that you have found them all. A given structure consists of an ordered path of the visited lattice sites (number is as in Fig. 1). Output this list of paths to a file.

Using your list of paths, write some code to convert each into the corresponding surface exposure sequence, **s**. Write out the list of unique sequences. How many unique structure sequences of 0's and 1's are there?

Designability Calculation:

b) Now calculate the designability of each of these structure sequences. To do this make random HP sequences, and for each one, evaluate the energy on each of your unique structures. If there is just a single unique ground state structure, give that structure a 'hit'. By doing many random HP sequences you will gradually build up the hits for each structure. The number of hits a structure gets is its designability. (Note: for non-pallindromic structure sequences, you should average together the hits of the two complementary structure sequences since they should have roughly equal numbers of sequences designing them. Then just report a single designability for one of the structure sequences so as to avoid double counting).

c) Make a ranked list of your designabilities and make a log-log plot of designability vs rank. (Look up the paper by Li, Tang and Wingreen, Science 1996 to see a designability plot for larger structures). Does it show Zipf law behaviour?

d) One of the properties highly designable structures is that they are energetically stable. When you are calculating the energies for each structure in (b) for a particular sequence, also calculate the average energy difference ΔE between the unique ground state structure (only if it exists) and the higher energy structures. Then tabulate the average of this energy difference $\langle \Delta E \rangle$ over all the sequences that design a particular structure. Make a plot of $\langle \Delta E \rangle$ vs designability. Does it show that the energy difference grows with designability? (A 4x4 lattice may be too small to see this property).

RNA Folding:

Consider the RNA sequence AUAUAU. Use the simple RNA folding model presented in class to answer the following questions. In the model, each stack contributes an energy $-E_s$.

a) What is the minimum energy structure? What is the minimum energy?

b) Draw the six possible structures that this sequence can adopt?

c) Using the above six structures, evaluate the partition function via $Z=\sum_i \exp(E_i/kT)$ where E_i is the energy of each structure.

d) If $E_s=4kT$, what is the probability of the sequence being in the minimum energy structure?

e) Use the recursive method detailed in class to calculate $Z=Z_{1,6}$. Please give all the $Z_{i,j}$ and $\hat{Z}_{i,j}$ that contribute to the partition function. Your answer should be the same as in (c).

f) The above calculations do not involve the entropy of the unstacked free parts of the polymer (it's as if T=0). If every unpaired base contributes α degrees of freedom (or states), what are the entropies of each of the above 4 structures. (Hint: entropy = k ln(# of states)). How would you change your calculation of Z?

g) The free energy of a structure is $F=E-TS$, evaluate the free energies for each of the 4 structures.

h) At what temperature does the open structure become the lowest free energy structure?

PCA Analysis:

Download the 2 data sets from the website. The first is a collection of 2-dimensional data points and the second is a set of 5-dimensional data.

a) Plot the 2-D data set.

b) For each data set compute the average $\vec{x}$ and the covariance matrix.

c) Compute the eigenvalues and eigenvectors of the covariance matrix for both data sets.

d) For the 2-D data set, draw on the 2-D data plot the directions of the two eigenvectors and the location of the average $\vec{x}$.

e) For the 5-D data set, compute the projection of each data point onto each of the 5 eigenvectors (i.e. For data vector $\vec{x}_i$, subtract off the average $\vec{x}$, $\vec{x}_i'=\vec{x}_i-\vec{x}$ and compute $\vec{x}_i'\cdot\vec{v}_j$ for each eigenvector $\vec{v}_j$). Make a 2D plot of the projections using the top two eigenvectors as the x and y axis respectively. You can now see your high dimensional data set on the most prominent data directions.