18.417 - Introduction to Computational Molecular Biology **PS 3**
Bonnie Berger and Manolis Kamvysselis November 15, 2001

Problem Set 3
Due Date: Thursday, November 29

1. **Phylogeny:**

a) We will show that no matter what data we have, if we are dealing with only three OTU's (the species which supply our actual data points), we can construct a perfect fit to the data. To this end, consider the following distance data on species A,B,C.

	A	B	C
A	0		
B	x	0	
C	y	z	0

Construct a phylogeny tree which fits the data perfectly. Clearly, the distances on the tree that you turn in as a solution should be variables, i.e., $(x + y - z)^2$.

b) Now we will prove that with as few as four data points, it is possible to have data which does not fit any tree perfectly. Recall the following lemma.

Lemma 1 (4 point condition) *A metric space O is additive iff given any 4 points in O, we can label them i, j, k, l, such that*

$$d_{ij} + d_{kl} = d_{ik} + d_{jl} \geq d_{il} + d_{jk}$$

Show that the following data defines a metric space (all the measured data obeys the triangle inequality), but that metric space is not additive (this will allow us to conclude that there is no phylogeny tree which fits the data perfectly). Hint: use the above lemma.

	A	B	C	D
A	0			
B	1	0		
C	1	.7	0	
D	1.1	1	.5	0

c) Considering the data from part b), what phylogeny tree does UPGMA (the Unweighted Pair Group Method with Arithmetic mean) lead to? Show your work.

2. **Gene Chips:** The following problem is due to Pablo Tamayo.

   Normal vs Renal Carcinoma Gene Expression Dataset

   The following table shows numerical values for mRNA concentrations obtained using DNA-microarrays from 6 normal and 6 renal carcinoma samples in human patients. The data was obtained using the chips manufactured by this company
   `http://www.affymetrix.com/technology/index.html`

   Based on the given data, answer the following three questions:

   a) Which genes are the best "markers" to separate normal from carcinoma samples?

   b) How would you classify new samples into normal and carcinomas using those markers?

   c) If you were to make a clinical assay to make this classification and could only test for three genes which three would you choose? (explain)

   The dataset shows 20 genes selected from a total of 6,817 in the chip. The first six columns are readings from normal cells, and the second six columns are readings from cancerous cells. This data is available in the table below, and also in a file located at `/mit/18.417/ProblemSets/genechip-data.txt`

12	L07648	42	175	−50	29	59	154	1087	252	93	309	66	60
13	U12465	1410	2120	1009	1070	1481	1965	5734	1714	1038	1487	3819	4935
14	X59798	−130	380	−358	−37	230	154	1050	592	1367	1872	1414	900
15	U39318	253	229	365	470	373	258	437	714	559	568	317	134
16	X52541	979	375	434	341	909	426	280	783	220	19	19	68
17	X56494	315	1091	62	−8	967	483	3303	2525	2104	1719	2002	2215
18	Z30644	870	1452	1707	1745	1500	1243	315	596	622	532	350	425
19	U96915	124	332	21	−95	151	145	652	257	51	−74	60	0
20	D10995	182	142	718	737	199	155	231	436	469	645	129	157
21	X51435	237	149	262	629	160	133	144	343	96	196	232	173
22	X01677	4788	4480	2535	2470	7088	6925	4040	4899	1538	8773	4289	6837
23	D42039	78	134	284	641	143	163	98	193	421	412	249	116
24	J03827	643	606	250	870	764	604	1659	606	459	482	316	858
25	U43189	110	195	298	329	65	137	188	291	668	423	148	61
26	S45630	1043	2263	1069	1845	2255	1613	5213	1191	2702	1962	473	142
27	U37251	337	243	398	662	107	184	271	452	388	584	243	200
28	U85992	281	206	546	670	26	219	217	195	357	292	448	169
29	L22342	373	163	573	824	422	368	311	359	926	605	298	243
30	Z30425	276	169	596	612	344	400	132	575	415	292	393	93
31	U70671	579	146	821	1787	806	632	247	369	995	790	356	838

Protein Motif Recognition: A Hidden Markov Model consists of some states and some transition probabilities. Sometimes we also include the initial probabilities in the definition of an HMM.

A common approach to building a Hidden Markov Model is to take some states, take some data that you want the HMM to represent, and then to apply the *EM framework* to generate transition probabilities that represent the data somewhat well. *EM* stands for *Expectation Maximization*.

What do we mean by somewhat well? EM is a *local maximum* heuristic. Although it does not necessarily lead to the transition probabiities which were most likely to have produced the given data, we can prove that for most initial guesses, EM will produce a better set of guesses. EM improves upon the guessed transition probabilities until we have reached a local maximum, where no small changes to the transition probabilities will improve our accuracy.

The EM algorithm is a general method for learning a parametric model of a partially-observable stochastic random process. Such a process generates two kinds of data: observable values O and hidden values H. Both O and H are random variables whose joint distribution depends only on some set of parameters P. Our goal is to learn the most likely value of P for a process by observing its output; in other words, we want the model that best fits the data. Because we cannot observe the hidden data, we will derive the expected value for the hidden data H at each step using P and O, and then re-evaluate our choice of parameters P in light of this estimate for the hidden data H. Formally, what we want to solve is the following, Given a fixed O, find the value of P maximizing [1]

$$\Pr[O \mid P] = \sum_{H} \Pr[O, H \mid P]$$

Formally, we define indicator random variables for the values of the hidden data. That is, let $H(i,j)$ be the random variable which is 1 if (i,j) happened and 0 if it didn't. $H(i,j)$ is the j^{th} possibility for $O(i)$. Starting from estimate P_z for P, compute H. After zeroing out the random variables that we know didn't happen because of O, normalize the rows to be probability measures. Now we pick P_{z+1} in such a way as to maximize the chance of getting this hidden data. This is achieved by setting each element $P_{z+1}(k)$ to be the post-O-normalization probability of $P(k)$ in H. Repeat.

Conside the following situation. The ABO blood type has three alleles, A, B, and O. The blood type of an individual depends as follows on what pairs of alleles he or she has: type A if the pair is A/A or A/O; type B if the pair is B/B or B/O; type AB if the pair is A/B; type O if the pair is O/O.

Let $p(\mathrm{A})$ be the fraction of A alleles in the population, and define $p(\mathrm{B})$ and $p(\mathrm{O})$ similarly. These fractions are non-negative and sum to 1. Making

[1] This paragraph paraphrased from Jeremey Buhler's 1/22/98 lecture notes on Dick Karp's class.

the reasonable assumption that every trait is inherited independently and at random, the probability that an individual has a given pair of alleles is the same as the probability of obtaining the pair in two random draws from the population. For example, the probability of having the pair of alleles A/B is $2p(\mathrm{A})p(\mathrm{B})$, while the probability of having alleles A/A is $p(\mathrm{A})^2$.

Say we go out and perform an experiment, sampling 30 individuals. We measure that 16 of them have blood type A, 2 have blood type B, 1 has blood type AB, and 11 have blood type O. Using EM, we will calculate the values of $p(\mathrm{A})$, $p(\mathrm{B})$, and $p(\mathrm{O})$ most likely to have given rise to the observed data.

We begin by setting up the problem as follows. Denote our observable data by O, and the hidden data by H. O consists of the 30 blood types we have measured (the phenotypes). H consists of the pairs of alleles that the individuals have (the genotypes). The parameters we are trying to optimize are $p(\mathrm{A})$, $p(\mathrm{B})$, and $p(\mathrm{O})$. Denoting our parameter matrix by P, we will find that our data fits the EM framework more straightforwardly if we use the following representation for P.

$$P = \begin{pmatrix} p(\mathrm{A}) & p(\mathrm{A}) \\ p(\mathrm{B}) & p(\mathrm{B}) \\ p(\mathrm{O}) & p(\mathrm{O}) \end{pmatrix}$$

Because EM is an iterative algorithm, we need an intial guess for P. Denoting this by P_0, and noting that there seem to be more A's than B's, but a lot of O's, we will guess

$$P_0 = \begin{pmatrix} .4 & .4 \\ .2 & .2 \\ .4 & .4 \end{pmatrix}$$

Note that this is a probability measure - the values sum to 1 in each column. The first step in the EM algorithm is to calculate the probability of getting all the possible hidden data given the parameters we have guessed. That is, what is the expected H given P_0? We represent H as four rows corresponding to the four different phenotypes. H should actually have one row for each observed individual, but since most of them will be the same, this more compact representation makes a lot of sense.

$$H = \begin{pmatrix} \text{probabilities of genotypes corresponding to phenotype A} \\ \text{probabilities ... B} \\ \text{probabilities ... AB} \\ \text{probabilities ... O} \end{pmatrix}$$

Now we calculate H_0 (H given P_0). Let each row be the probabilities for the genotypes in the order
AA, AO, OA, BB, BO, OB, AB, BA, OO

$$H_0 = \begin{pmatrix} .16 & .16 & .16 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .04 & .08 & .08 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .08 & .08 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .16 \end{pmatrix}$$

We want to set P_1 to maximize the probability of getting back O. We will first explain the general methodology, and then walk through the calculation of the top left entry of P_1. The general methodology for calculating entry (i, j) of P_1 is to normalize each row of H_0 so that it sums to a probability measure, and then to add up all the entries of H_0 where a type i allele occurs in position j, multiplying each row by the number of individuals it represents, and finally dividing the sum by the total number of individuals.

As an example, we calculate the left-most $p(\mathrm{A})$ estimate for P_1 using H_0. First we normalize H_0, obtaining $\hat{H}_0$.

$$\hat{H}_0 = \begin{pmatrix} 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/5 & 2/5 & 2/5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Now we sum over all the genotype entries in which the allele A appears on the left. Each row corresponding to the phenotype A (the first row in $\hat{H}_0$) contributes $1/3 + 1/3$, and there are 16 such individuals. The only other contribution is from the AB row, which contributes $1/2$. There is only one such individual. Thus we get that the new value for $p(\mathrm{A})$ in the left side of P_1 is $(16 * 2/3 + 1 * 1/2)/30 = .372$

Finishing out the calculations yields

$$P_1 = \begin{pmatrix} .372 & .372 \\ .056 & .056 \\ .572 & .572 \end{pmatrix}$$

For the next step, we plug in P_1, calculate H_1, and then get out P_2.

As a sanity check that we are making forward progress, let us calculate the chance of getting our observed data given the hypothesized distribution of alleles in the population reflected in P_0. A bit of calculation shows that this was about $8.2\text{x}10^{-5}$. The same calculation on P_1 shows the probability has risen to $1.3\text{x}10^{-2}$

Your assignment is to calculate what the P_i converge to, to three significant decimal places. I.e., run the algorithm until P_i doesn't change. I suggest you use matlab, but you are welcome to do this by hand - less than six more iterations are necessary. If you are interested, you are welcome to calculate what the probability of getting your observed data is given this final P_i - how much further does it rise from $1.3 \text{x} 10^{-2}$?