Computational Genomics MSCBIO2070/02-710
Spring 2015
PS1 Solutions

1.
a.

UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys
UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys
UUA	Leu	UCA	Ser	UAA	Stop	UGA	Stop
UUG	Leu	UCG	Ser	UAG	Stop	UGG	Trp
CUU	Leu	CCU	Pro	CAU	His	CGU	Arg
CUC	Leu	CCC	Pro	CAC	His	CGC	Arg
CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
GUU	Ile	GCU	Thr	GAU	Asn	GGU	Ser
GUC	Ile	GCC	Thr	GAC	Asn	GGC	Ser
GUA	Ile	GCA	Thr	GAA	Lys	GGA	Arg
GUG	Met	GCG	Thr	GAG	Lys	GGG	Arg
AUU	Val	ACU	Ala	AAU	Asp	AGU	Gly
AUC	Val	ACC	Ala	AAC	Asp	AGC	Gly
AUA	Val	ACA	Ala	AAA	Glu	AGA	Gly
AUG	Val	ACG	Ala	AAG	Glu	AGG	Gly

b. At each position there are 4! possible permutations so we have $(4!)^3$ total permutations. If you restricted yourself to pairwise permutations, or took out some repeats I gave full credit as long as you had a reasonable explanation.

c. The list should produce exactly the same set of amino acids.

GUG		AUG	
GUC	Val	AUC	Val
GUA	Val	AUA	Val
GUU	Val	AUU	Val
GAG	Glu	AAG	Glu
GCG	Ala	ACG	Ala
GGG	Gly	AGG	Gly
AUG	Met	GUG	Met
UUG	Leu	UUG	Leu
CUG	Leu	CUG	Leu

1. continued

a. |AAA|AAX|,|XXA|AAA|AXX|,|XAA|AAA|
b. All three reading frames contain the stop codon; none can admit this n-mer.

c.

S = CCGGU
CCG|GUN
NCC|GGU
NNC|CGG|UNN
Stop codons appear in two of the three possible reading frames; therefore only one possible reading frame can admit this n-mer.

d. Yes, Aspartic acid and Glutamic acid are both encoded by GAN.

2.
a.

$$\mathrm{P(x\ =\ 0)} = \frac{1^0 * e^{-1}}{0!} = 0.368 = 36.8\%\ not\ sequenced$$

$$\mathrm{P(x\ =\ 0)} = \frac{2^0 * e^{-2}}{0!} = 0.135 = 13.5\%\ not\ sequenced\ for\ 2X\ coverage$$

$$\mathrm{P(x\ =\ 0)} = \frac{4^0 * e^{-4}}{0!} = 0.0183 = 1.83\%\ not\ sequenced\ for\ 4X\ coverage$$

$$\mathrm{P(x\ =\ 0)} = \frac{8^0 * e^{-8}}{0!} = 0.00033 = 0.033\%\ not\ sequenced\ for\ 8X\ coverage$$

As coverage increases, the percentage of the genome not covered exponentially decreases.

b.

$$P(x = 3) = \frac{10^3}{3!} e^{-10} = 0.76$$

$$P(x \leq 3) = P(x = 0) + P(x = 1) + P(x = 2) + P(x = 3) = \left(\frac{10^0}{0!} + \frac{10^1}{1!} + \frac{10^2}{2!} + \frac{10^3}{3!}\right) * e^{-10}$$
$$= (1 + 10 + 50 + 166.66) * 4.54 * 10^{-3} = 0.0103$$

c.

$$P(x = 0) = \frac{\lambda^x * e^{-x}}{x!} = \frac{C^0 * e^{-C}}{0!} = e^{-C}$$

d. Total gap length = $P(x = 0) * G = G * e^{-C}$
Number of bases not sequenced = number of bases total * P(single base not sequenced)

e. Expected number of gaps = $P(x = 0) * N = N * e^{-C}$
Think of this as looking at the next base after the end of each read and checking if it is sequenced. If the next base is not sequenced, then there's a gap, so the expected number of gaps = number of reads sequenced * P(single base not sequenced)

3.

a.

$$Bx_i = y_i = \begin{bmatrix} a_1 & 1-a_1 \\ \vdots & \vdots \\ a_k & 1-a_k \end{bmatrix} \begin{bmatrix} x^c \\ x^l \end{bmatrix} = \begin{bmatrix} y^1 \\ \vdots \\ y^k \end{bmatrix}$$

b.

$$\begin{bmatrix} a_1 & 1-a_1 \\ \vdots & \vdots \\ a_k & 1-a_k \end{bmatrix} \begin{bmatrix} x_1^c \\ x_1^l \end{bmatrix} = \begin{bmatrix} y_1^1 & \cdots & y_n^1 \\ \vdots & \ddots & \vdots \\ y_1^k & \cdots & y_n^k \end{bmatrix}$$

c. Example MATLAB code

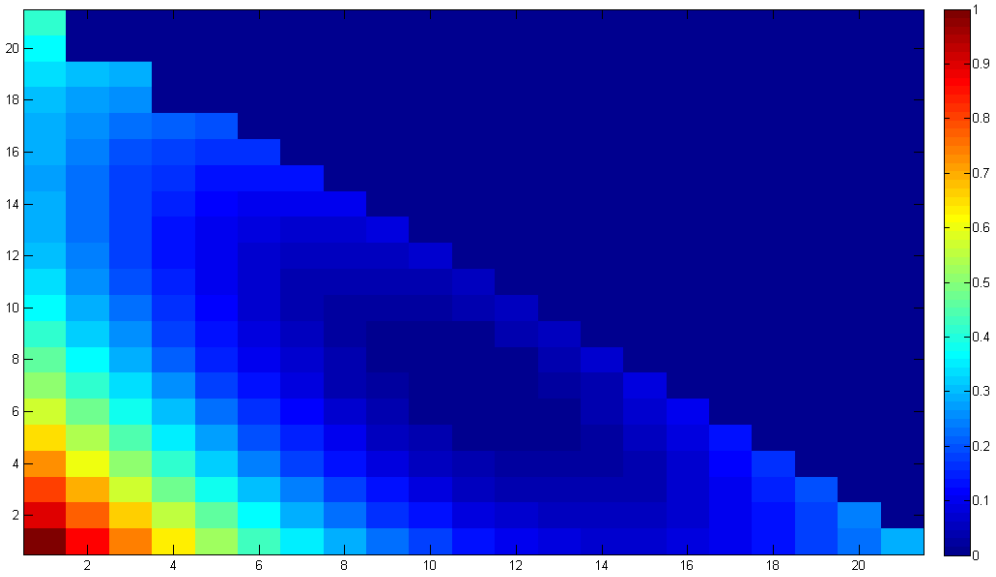
```
data = textread('expression_data.txt');
mixed = data(:,1:4);
A = [0.2 0.8; 0.4 0.6; 0.6 0.4; 0.8 0.2];
recover= [];
for i = 1:size(mixed,1)
y = mixed(i,:)';
x = A\y;
recover(i,:) = x';
end
```

b. Example MATLAB code

```
pure = textread('pure_samples.txt');
pure = pure(:,1:end-1);

mix = textread('mixture_data.txt');
mix = mix(:,1:end-1);

step = 0.05;
ctri = 0;
err = [];
for i = 0:step:1
ctri = ctri + 1;
labelx(ctri) = i;
ctrj = 0;
for j = 0:step:1-i
ctrj = ctrj + 1;
labely(ctrj) = j;
k = 1 - i - j;
if k < 0
k = 0;
end
out = i*pure(:,1) + j*(pure(:,2)) + k*pure(:,3);
err(ctri,ctrj) = sum((mix - out).^2/length(mix));
%err(ctrj,ctri) = err(ctri, ctrj);
end
end
figure; imagesc(err/max(err(:))); axis xy; colorbar;
```



Note that half the heatmap should not be filled in, as those are impossible values of a1 and a2, i.e. a1 = a2 = 1 in the top right corner.

b. I accepted answers that gave an explanation about an iterative algorithm, most people used an EM algorithm.