

CS 530 Geometric and Prob. Methods in CS

Homework 3

Erik Barry Erhardt

October 2, 2006

Note: All code can be found in the Appendix.

Exercise 1 *Buchnera amino acid proportions.*

The bar chart in Figure 1 on page 1 presents the proportions of each amino acid in the Buchnera chromosome amino acid sequence. Table 1 on page 2 gives the proportions plotted in Figure 1.

The calculated entropy is $H = - \sum_{i=1}^{21} p_i \log_2(p_i) = 4.0998$.

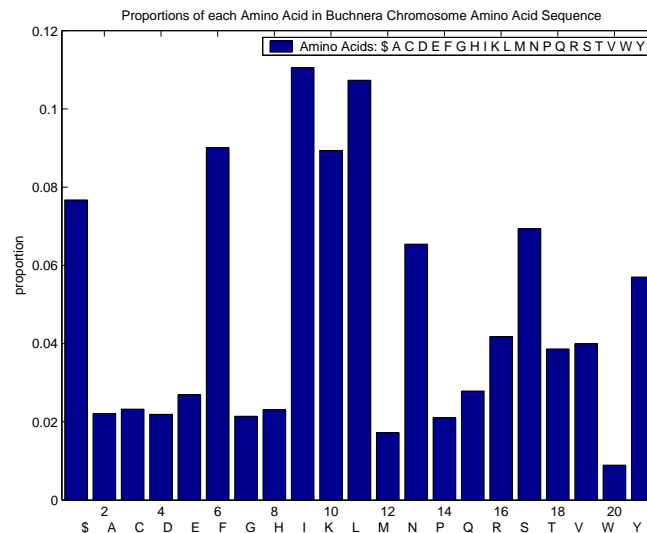


Figure 1: Exercise 1, Proportions of each amino acid in the Buchnera chromosome amino acid sequence.

Symbol	Proportion	Frequencies
\$	0.076718	16384
A	0.022092	4718
C	0.023235	4962
D	0.021919	4681
E	0.026948	5755
F	0.090101	19242
G	0.021441	4579
H	0.023104	4934
I	0.11053	23604
K	0.089366	19085
L	0.10732	22920
M	0.017218	3677
N	0.065415	13970
P	0.021029	4491
Q	0.027875	5953
R	0.041801	8927
S	0.069362	14813
T	0.038631	8250
V	0.039937	8529
W	0.0089436	1910
Y	0.057014	12176
	1.0	213560

Table 1: Exercise 1, Proportions of each amino acid in the Buchnera chromosome amino acid sequence.

Exercise 2 *Huffman coding.*

The tree in Figure 2 on page 3 (alternatively, Figure 3 on page 4) shows the Huffman coding of the amino acids based on the observed probabilities for Buchnera.

Sym	Probs	Freqs	Huff	1	2	3	4	5	6	7
\$	0.0767	16384	GG					G-		
A	0.0221	4718	GTG		G-					
C	0.0232	4962	AA			A-0.1167				A-1.000
D	0.0219	4681	GTC		C-					
E	0.0269	5755	AC			C-				
F	0.0901	19242	TC						C-	
G	0.0214	4579	GTA		A-0.0886	=====>	T-			
H	0.0231	4934	GTT		T-					
I	0.1105	23604	TT						T-	
K	0.0894	19085	TA						A-0.3001	>T-
L	0.1073	22920	TG						G-	
M	0.0172	3677	CGG	G-						
N	0.0654	13970	GA					A-0.3001	=====>	G-
P	0.0210	4491	CGT	T-						
Q	0.0279	5953	AG			G-				
R	0.0418	8927	CC				C-			
S	0.0694	14813	GC					C-		
T	0.0386	8250	AT			T-				
V	0.0399	8529	CA				A-0.1859	=====>	C-	
W	0.0089	1910	CGC	C-						
Y	0.0570	12176	CT				T-			
-	0	0	CGA	A-0.0472	=====>	G-				

Figure 2: Exercise 2, Huffman coding of the amino acids based on the observed probabilities for Buchnera.

The expected length of the Huffman code is $E(L) = \sum_{i=0}^{20} L_i p_i = 2.1357$,

while the length of dna sequences for amino acids are all 3 (and there are duplicate dna triplets for the same amino acid).

DNA efficiency is $\eta_{dna} = \frac{H}{E(L) \log_2(4)} = \frac{4.0998}{3 \times 2} = 0.6833$. (Actually less than this since there are duplicate dna triplets for the same amino acid).

Huffman efficiency is $\eta_{Huffman} = \frac{H}{E(L) \log_2(4)} = \frac{4.0998}{2.1357 \times 2} = 0.9598$.

The Huffman coding is more than 40% more efficient than the dna coding.

Sym	Probs	Freqs	Huff	1	2	3	4	5	6	7
C	0.0232	4962	AA			A	0.1167=====			A 1.000
E	0.0269	5755	AC			C	///			
Q	0.0279	5953	AG			G	//			
T	0.0386	8250	AT			T	/			
V	0.0399	8529	CA					A	0.1859=====	C
R	0.0418	8927	CC					C	///	
-	0	0	CGA	A	0.0472====	G	/			
W	0.0089	1910	CGC	C	///					
M	0.0172	3677	CGG	G	//					
P	0.0210	4491	CGT	T	/			/		
Y	0.0570	12176	CT					T	/	
N	0.0654	13970	GA					A	0.3001=====	G /
S	0.0694	14813	GC					C	///	
\$	0.0767	16384	GG					G	//	
G	0.0214	4579	GTA	A	0.0886====	T	/			
D	0.0219	4681	GTC	C	///					
A	0.0221	4718	GTG	G	//					
H	0.0231	4934	GTT	T	/					/
K	0.0894	19085	TA					A	0.3001=>	T /
F	0.0901	19242	TC					C	///	
L	0.1073	22920	TG					G	//	
I	0.1105	23604	TT					T	/	

Figure 3: Exercise 2 (alternative shows groupings together), Huffman coding of the amino acids based on the observed probabilities for Buchnera.

Exercise 3 *Conditional Entropy.*

$$H_{t|t-1} = - \sum_{i=1}^{21} \sum_{j=1}^{21} p_{t,t-1}(i,j) \log_2 p_{t|t-1}(i|j) = 4.0647$$

Exercise 4 *Limiting Distribution.*

Because there is a single largest eigenvalue equal to 1, the transition matrix $P_{t|t-1}$ is both irreducible and aperiodic. The limiting distribution is the normalized eigenvector associated with the eigenvalue equal to 1. Because the difference between the limiting distribution and the observed distribution of amino acids is 0 to 5 decimal places, I won't replot the distribution; refer to Figure 1 on page 1.

Exercise 5 *Coin flip transition probability matrix.*

$$P_{t|t-1}(t|t-1) = \begin{array}{c|cccc} & S & H & HH & HHT \\ \hline S & 0.5 & 0.5 & 0 & 0.5 \\ H & 0.5 & 0 & 0 & 0.5 \\ HH & 0 & 0.5 & 0.5 & 0 \\ HHT & 0 & 0 & 0.5 & 0 \end{array}$$

This matrix is irreducible and aperiodic, with the limiting distribution for $\{S, H, HH, HHT\} = 0.375, 0.25, 0.25, 0.125$.

Exercise 6 *Production system.*

The probability that the 4th is defective given that the first is defective is $P^3(0, t = 4 | 0, t = 1) = 0.6848$.

Exercise 7 *Markov process.*

(a,c) x. The table below gives the distributions $x^{(i)} = P^i x^{(0)}$, $i = 1, 2, 3, 4$, and the limiting distribution of $i = \infty$.

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$	$x^{(\infty)}$
0.1000	0.0700	0.0610	0.0520	0.0467	0.0400
0.1000	0.2800	0.3580	0.3973	0.4174	0.4400
0.5000	0.3200	0.3050	0.2918	0.2862	0.2800
0.3000	0.3300	0.2760	0.2589	0.2497	0.2400

(b) P. Transition matrices:

$$\begin{aligned}
 P^1 &= \begin{bmatrix} 0.4000 & 0 & 0 & 0.1000 \\ 0 & 0.7000 & 0.3000 & 0.2000 \\ 0.3000 & 0.2000 & 0.3000 & 0.4000 \\ 0.3000 & 0.1000 & 0.4000 & 0.3000 \end{bmatrix}, \\
 P^2 &= \begin{bmatrix} 0.1900 & 0.0100 & 0.0400 & 0.0700 \\ 0.1500 & 0.5700 & 0.3800 & 0.3200 \\ 0.3300 & 0.2400 & 0.3100 & 0.3100 \\ 0.3300 & 0.1800 & 0.2700 & 0.3000 \end{bmatrix}, \\
 P^3 &= \begin{bmatrix} 0.1090 & 0.0220 & 0.0430 & 0.0580 \\ 0.2700 & 0.5070 & 0.4130 & 0.3770 \\ 0.3180 & 0.2610 & 0.2890 & 0.2980 \\ 0.3030 & 0.2100 & 0.2550 & 0.2670 \end{bmatrix}, \\
 P^4 &= \begin{bmatrix} 0.0739 & 0.0298 & 0.0427 & 0.0499 \\ 0.3450 & 0.4752 & 0.4268 & 0.4067 \\ 0.3033 & 0.2703 & 0.2842 & 0.2890 \\ 0.2778 & 0.2247 & 0.2463 & 0.2544 \end{bmatrix}.
 \end{aligned}$$

Exercise 8 *Hazel/Naomi kumquats/persimmons.*

(a,b,c) Expressions for Hazel number of kumquats. See transition matrix in (d). (a) is lower diagonal, (b) is diagonal, (c) is upper diagonal.

(d) Transition matrix. This is the transition matrix for a given girl having $\{0, 1, 2, 3\}$ of a given fruit.

$$P = \begin{array}{c|ccc|cc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & \left(\frac{1}{3}\right)^2 = 1/9 & 0 & 0 \\ 1 & 1 & 2\left(\frac{1}{3}\right)\left(\frac{2}{3}\right) = 4/9 & \left(\frac{2}{3}\right)^2 = 4/9 & 0 \\ 2 & 0 & \left(\frac{2}{3}\right)^2 = 4/9 & 2\left(\frac{2}{3}\right)\left(\frac{1}{3}\right) = 4/9 & 1 \\ 3 & 0 & 0 & \left(\frac{1}{3}\right)^2 = 1/9 & 0 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

(e) Irreducible? Aperiodic? P is irreducible since every state communicates with every other state. This is obvious since the upper and lower diagonals are all nonzero.

P is aperiodic since at least one state communicates with itself, both states 1 and 2.

By the way, the limiting distribution is $\{0, 1, 2, 3\} = 0.05, 0.45, 0.45, 0.05$.

Exercise 9 *Ising model.*

The Ising model on a torus using a plus-sign stencil using Gibb's sampling will update each point on the lattice in a random order (so each iteration updates every lattice point, but in a random order), subject to the probability of being either a +1 or -1 by

$$\Pr(X_C = x_C | X_t = x_t \in N_C) = \frac{e^{x_C(x_N+x_S+x_W+x_E)}}{e^{x_C(x_N+x_S+x_W+x_E)} + e^{-x_C(x_N+x_S+x_W+x_E)}}$$

resulting in the probabilities $\Pr(\text{stay same—current state, neighbors})$ in this table:

State	Number of neighbors with +				
	0	1	2	3	4
+1	$\frac{e^{-4}}{e^{+4}+e^{-4}} = 0.9997$	$\frac{e^{-2}}{e^{+2}+e^{-2}} = 0.9820$	$\frac{1}{2} = 0.5$	$\frac{e^{+2}}{e^{+2}+e^{-2}} = 0.0180$	$\frac{e^{+4}}{e^{+4}+e^{-4}} = 0.0003$
-1	$\frac{e^{+4}}{e^{+4}+e^{-4}} = 0.0003$	$\frac{e^{+2}}{e^{+2}+e^{-2}} = 0.0180$	$\frac{1}{2} = 0.5$	$\frac{e^{-2}}{e^{+2}+e^{-2}} = 0.9820$	$\frac{e^{-4}}{e^{+4}+e^{-4}} = 0.9997$
	1	1	1	1	1

The plot in Figure 4 on page 9 shows the evolution of the Ising model at $10^3, 10^4, 10^5, 10^6$ iterations. Just as the probabilities suggest, lattice points change with high probability to the spin of the four points in it's neighborhood stencil.

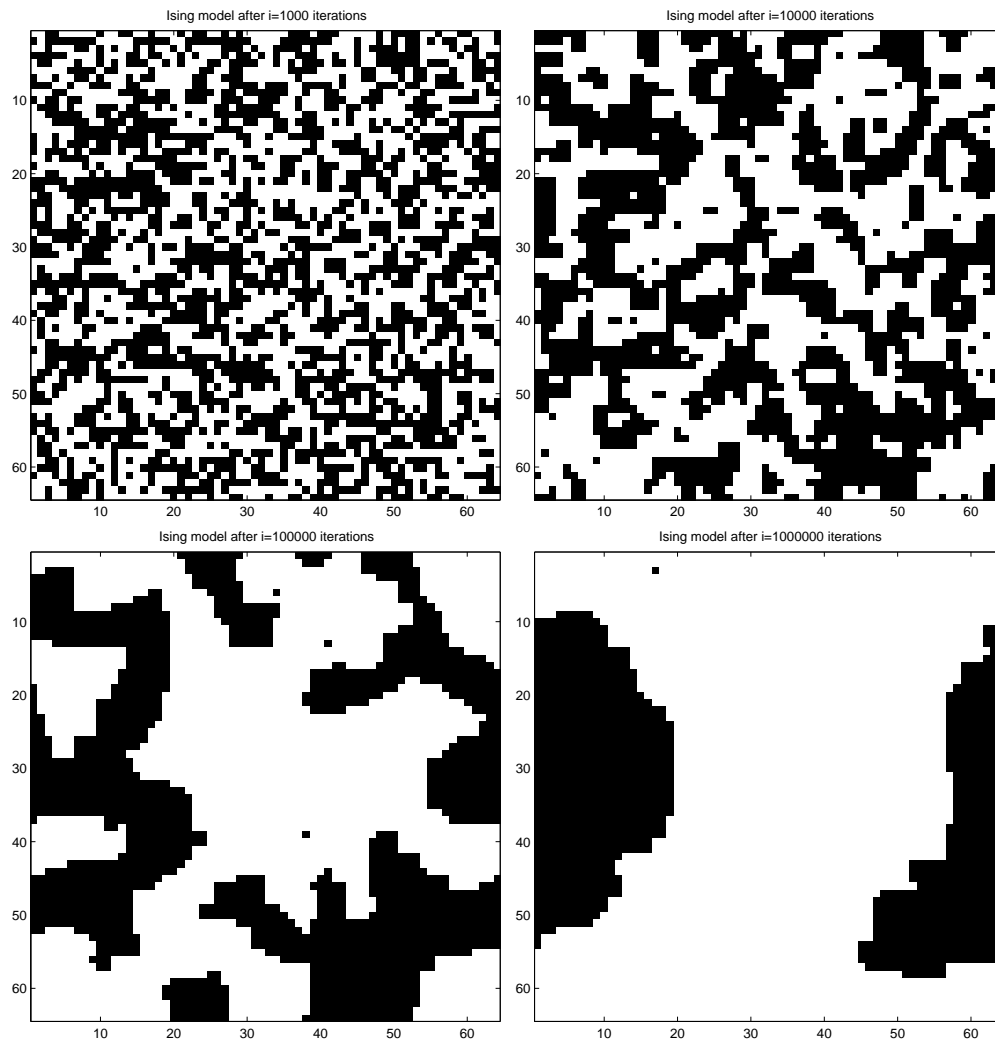


Figure 4: Exercise 9, Ising model evolution to regions of self-similarity.

Appendix

Matlab code used for the exercises

```

%%% Exercise 1 =====
% read data
fn='buchnera_acids.txt';
fid=fopen(fn,'r');
aa=fscanf(fid,'%1s');
aa=aa'; % column vector
fclose(fid);
%size(aa)
%aas=sort(aa); % sorted characters
%ind=find(aas=='S');indl=ind(l); % first index with non blank character
%aat=aas(indl:end); % array to tabulate
n=length(aa) % number of characters
%tab=tabulate(aat);

aac = ['$','A','C','D','E','F','G','H','I','K','L','M','N','P','Q','R','S','T','V','W','Y']; % characters

l=length(aac); % number of characters
naac = zeros(1,l);
for i=1:l;
    naac(i) = length(find(aa==aac(i)));
end;

paa=naac/n; % probabilities

% bar chart of proportion of each amino acid
bar(1:l,paa);
axis([0.5 21.5 0 .12]);
legend('Amino Acids: $ A C D E F G H I K L M N P Q R S T V W Y')
title('Proportions of each Amino Acid in Buchnera Chromosome Amino Acid Sequence')
xlabel('$ A C D E F G H I K L M N P Q R S T V W Y');
ylabel('proportion')
plot_name = strcat('gpcshw03-1.eps'); % plot name
print(gcf, '-depsc2', plot_name); % print plot

H=-sum(paa.*log2(paa)) % Entropy

%%% Exercise 2 =====
t=[naac ; 0];t=sort(t);disp(t);disp(sum(t(1:4)));
t=[sum(t(1:4)); t(5:end)];t=sort(t);disp(t);disp(sum(t(1:4))); % repeat

L=[3 2 3 2 2 3 3 2 2 3 2 2 3 2 3 2 2 2 2 2 3 2 2];
EL=sum(L.*paa);

efficiency_dna=H/(3*log2(4))
efficiency_hof=H/(EL*log2(4))

efficiency_hof/efficiency_dna % relative efficiency

%%% Exercise 3 =====
P = zeros(21,21);
for i=2:n;
    r = find(aac==aa(i)); % to
    c = find(aac==aa(i-1)); % from
    P(r,c) = P(r,c)+1; % increment counter
end;

P=P./sum(sum(P));
Pcond = P./repmat(sum(P,1),21,1); % conditional P, columns sum to 1

HtGtm1 = -sum(sum(P.*log2(Pcond))) % conditional entropy

%%% Exercise 4 =====
[vec,eval]=eig(Pcond);
diag(eval)
vec(:,1)

vec_norm = vec(:,1)./sum(vec(:,1))
[vec_norm paa vec_norm-paa]

%%% Exercise 5 =====
P5=[.5 .5 0 .5;.5 0 0 .5; 0 .5 .5 0; 0 0 .5 0]
[vec,eval]=eig(P5);
diag(eval)
vec(:,1)
vec_norm = vec(:,1)./sum(vec(:,1))

```

```

%%% Exercise 6 =====
P6=[.99 .12;.01 .88]
P6_3=P6^3
P6_3(2,2)

%%% Exercise 7 =====
P7=[.4 0 0 .1;0 .7 .3 .2;.3 .2 .3 .4; .3 .1 .4 .3]
x0=[.1 .1 .5 .3]

% a
x1=P7*x0
x2=P7^2*x0
x3=P7^3*x0
x4=P7^4*x0

% b
P7^2
P7^3
P7^4

% c
[vec,eval]=eig(P7);
diag(eval)
vec(:,1)
vec_norm = vec(:,1)./sum(vec(:,1))

[x0 x1 x2 x3 x4 vec_norm]

%%% Exercise 8 =====
P8=[0 (1/3)^2 0 0; 1 2*(1/3)*(2/3) (2/3)^2 0; 0 (2/3)^2 2*(1/3)*(2/3) 1; 0 0 (1/3)^2 0]
[vec,eval]=eig(P8);
diag(eval)
vec(:,1)
vec_norm = vec(:,1)./sum(vec(:,1))

%%% Exercise 9 =====
s=64;
I=sign(rand(s)-.5);
gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);
warning off MATLAB:divideByZero
for i=1:1e6;
    I=ising(I,'one');
    if mod(i,1e3)==0;
        i
        %gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);
        %tit=strcat('Ising model after i=',num2str(i),' iterations');
        %title(tit);
        %pause(.0001);
    end;

    % output

    if i==1e3; gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);tit=strcat('Ising model after i=',num2str(i),' iterations');
    if i==1e4; gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);tit=strcat('Ising model after i=',num2str(i),' iterations');
    if i==1e5; gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);tit=strcat('Ising model after i=',num2str(i),' iterations');
    if i==1e6; gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);tit=strcat('Ising model after i=',num2str(i),' iterations');

    %I=ising(I);gdisplay(I);axis equal;axis square;axis([.5 s+.5 .5 s+.5]);
    %pause(.0001)
end;

exp(4)/(exp(4)+exp(-4))
exp(2)/(exp(2)+exp(-2))
1/2
exp(-2)/(exp(2)+exp(-2))
exp(-4)/(exp(4)+exp(-4))

```

Listing 1: Matlab routine `ising.m` used to .

```

function I=ising(I,one_or_all);
% function I = ising(I)
%
% Perform one full Gibbs iteration (all sites updated by random sampling)
5 % by the Ising model on an initial two-dimensional images with states in {-1,1}
% Calculations are done on a Torus (edges wrap) with plus-sign stencil.
%
% Erik Barry Erhardt 10/2/2006

```

```

10  % % how to use:
    % I=sign(rand(64)-.5);gdisplay(I);
    % for i=1:500;
    %     I=ising(I,3);if mod(i,10)==10; gdisplay(I); end;
    % end;
15
    % error checking
    if nargin>2; error('One or Two input arguments required'); return; end;
    if nargin==1; one_or_all = 'all'; end;
20  if length(size(I))~=2; error('Two-dimensional image'); return; end;
    if length(find(abs(I)~=1))~=0; error('image must be of -1 or 1 only'); return; end;
    if one_or_all ~= 'one' & one_or_all ~= 'all'; error('second argument must be one or all'); return; end;

25  [nr,nc]=size(I);           % grid size
    Ntot=nr*nc;              % total sites
    [a,b]=meshgrid(1:nr,1:nc); % a has rows {1,2,3,...}, b has cols {1,2,3,...}' to act as indicies
    idxlist=[a(:) b(:)];    % all pairs of indicies
    [tmp,idx] = sort(rand(Ntot,1)); % randomize index order for Gibbs update
30
    % On Torus

    if one_or_all == 'one'; Ntot = 1; end;%if
35  for i=1:Ntot

        r=idxlist(idx(i),1); % row/col pair
        c=idxlist(idx(i),2); % to update

40  % orientation has point (1,1) in upper left of image, just as a matrix index

        if r == 1; % top border, N wraps
            ind_N = nr;
            ind_S = r+1;
45  elseif r == nr % bottom border, S wraps
            ind_N = r-1;
            ind_S = 1;
        else % interior, normal
            ind_N = r-1;
50  ind_S = r+1;
        end;%if

        if c == 1; % top border, W wraps
            ind_W = nc;
            ind_E = c+1;
55  elseif c == nc % bottom border, E wraps
            ind_W = c-1;
            ind_E = 1;
        else % interior, normal
60  ind_W = c-1;
            ind_E = c+1;
        end;%if

        C = I(r, c ); % Center
65  N = I(ind_N, c ); % North
        S = I(ind_S, c ); % South
        W = I(r, ind_W); % West
        E = I(r, ind_E); % East

70  % [r;c;ind_N;ind_S;ind_W;ind_E] % test
        % [C,N,S,W,E] % test
        % I % test

        %calc energy
75  e = C*(N+S+W+E); % exponent
        p_same = exp(e)/(exp(e)+exp(-e)); % probability of staying the same

        % [e,p_same]

80  if (rand(1,1) > p_same) % change with probability (1-p_same);
            I(r,c)=-C;
        end;%if

85  end;%for
    % eof

```