# Math 127 (Spring 2007)

Homework #2

15 Feb 2007

## Question 1

The *genetic code* is a description of how information on DNA or RNA is translated into proteins in living organisms. The DNA molecule is made up of a double helix of bases, known as nucleotides. There are four nucleotides: Adenine, Cytosine, Guanine and Thymine, which are referred to by the four letters, A, C, G and T. In the 1960s, it was discovered that proteins are formed by translating the 64 different triplets of nucleotides into the 20 different types of amino acids found in proteins. These triplets of nucleotides are called *codons*. The genetic code is the translation table between codons and amino acids. It has been found that almost all organisms use the same translation table, referred to as the standard genetic code (shown in the table below, from ASCB Pg 128). As can be seen from the table, three of the codons do not correspond to any amino acid. They are *stop codons*, because they signal to protein-forming mechanisms in the cell where the nucleotide-to-amino-acid translation should stop.

| | T | C | A | G |
|---|---|---|---|---|
| | TTT $\mapsto$ Phe | TCT $\mapsto$ Ser | TAT $\mapsto$ Tyr | TGT $\mapsto$ Cys |
| T | TTC $\mapsto$ Phe | TCC $\mapsto$ Ser | TAC $\mapsto$ Tyr | TGC $\mapsto$ Cys |
| | TTA $\mapsto$ Leu | TCA $\mapsto$ Ser | TAA $\mapsto$ *stop* | TGA $\mapsto$ *stop* |
| | TTG $\mapsto$ Leu | TCG $\mapsto$ Ser | TAG $\mapsto$ *stop* | TGG $\mapsto$ Trp |
| | CTT $\mapsto$ Leu | CCT $\mapsto$ Pro | CAT $\mapsto$ His | CGT $\mapsto$ Arg |
| C | CTC $\mapsto$ Leu | CCC $\mapsto$ Pro | CAC $\mapsto$ His | CGC $\mapsto$ Arg |
| | CTA $\mapsto$ Leu | CCA $\mapsto$ Pro | CAA $\mapsto$ Gln | CGA $\mapsto$ Arg |
| | CTG $\mapsto$ Leu | CCG $\mapsto$ Pro | CAG $\mapsto$ Gln | CGG $\mapsto$ Arg |
| | ATT $\mapsto$ Ile | ACT $\mapsto$ Thr | AAT $\mapsto$ Asn | AGT $\mapsto$ Ser |
| A | ATC $\mapsto$ Ile | ACC $\mapsto$ Thr | AAC $\mapsto$ Asn | AGC $\mapsto$ Ser |
| | ATA $\mapsto$ Ile | ACA $\mapsto$ Thr | AAA $\mapsto$ Lys | AGA $\mapsto$ Arg |
| | ATG $\mapsto$ Met | ACG $\mapsto$ Thr | AAG $\mapsto$ Lys | AGG $\mapsto$ Arg |
| | GTT $\mapsto$ Val | GCT $\mapsto$ Ala | GAT $\mapsto$ Asp | GGT $\mapsto$ Gly |
| G | GTC $\mapsto$ Val | GCC $\mapsto$ Ala | GAC $\mapsto$ Asp | GGC $\mapsto$ Gly |
| | GTA $\mapsto$ Val | GCA $\mapsto$ Ala | GAA $\mapsto$ Glu | GGA $\mapsto$ Gly |
| | GTG $\mapsto$ Val | GCG $\mapsto$ Ala | GAG $\mapsto$ Glu | GGG $\mapsto$ Gly |

Because there are many more possible codons than types of amino acids, usually, several different codons map to the same amino acid. However, for a given amino acid, there is often a preference for certain codons to be used in the translation process than other codons. This is known as *codon bias*. Different organisms exhibit a bias for different codons, and though reason for this phenomena is not well understood.

## Question 2

Every occurrence of 6 in $\tau_1\tau_2\tau_3\tau_4$ results in the substitution of

$$\begin{aligned} f_6 &= 1 - f_1 - f_2 - \ldots - f_5, \\ l_6 &= 1 - l_1 - l_2 - \ldots - l_5 \end{aligned}$$

in $p_{\tau_1\tau_2\tau_3\tau_4}$, producing many monomial terms when the polynomial is expanded. Hence, we postulate that the polynomial $p_{6666}$ has the most number of terms. Also, when two letters are equal in $\tau_1\tau_2\tau_3\tau_4$, there is a possibility for many of the monomial terms to be equal, thus reducing the number of terms. Thus, we postulate that the polynomials $p_{iiii}$ for any $i \in [5]$ has the least number of terms. Indeed, checking against the expansions of the 1296 polynomials in Question 3, we see that $p_{6666}$ has the most with 4701 terms, and $p_{iiii}, i \in [5]$ has the least with 16 terms.

**Addendum by Bernd Sturmfels**: *if we work with the homogeneous version of this model, where e.g. $f_6$ is an unknown and not set equal to $1 - f_1 - f_2 - \ldots - f_5$ then each coordinate function of the model is a homogeneous polynomial of degree seven in the $12+4 = 16$ parameters. These polynomials have either 14 or 16 terms. The maximum, 16, is attained, for instance, for $\tau = (1, 2, 3, 4)$, and the minimum, 14 is attained, for instance, for $\tau = (1, 1, 1, 1)$. See the* `maple` *output file which I e-mailed to everyone on February 17.*

## Question 3

Since each monomial in $p_{\tau_1\tau_2\tau_3\tau_4}$ has degree at most 4 in the ten unknowns $f_1, f_2, \ldots, l_5$ and degree at most 3 in the two unknowns $x, y$, the total number of monomials occurring in the 1296 coordinate polynomials is at most

$$\binom{10+4}{4}\binom{2+3}{2} = 10010.$$

We generate the 1296 polynomials and collect the coefficients of the monomials in a $1296 \times 10010$ matrix, and compute the rank of the matrix. The answer is 966.

## Question 4

By the multinomial theorem for tropical arithmetic,

$$\begin{aligned} & [(w_1 \odot q_1^{a_{11}} \odot q_2^{a_{21}} \odot \ldots \odot q_d^{a_{d1}}) \oplus \ldots \oplus (w_n \odot q_1^{a_{1n}} \odot q_2^{a_{2n}} \odot \ldots \odot q_d^{a_{dn}})]^m \\ = & \bigoplus\nolimits_{u_1+\ldots+u_n=m} (w_1 \odot q_1^{a_{11}} \odot q_2^{a_{21}} \odot \ldots \odot q_d^{a_{d1}})^{u_1} \odot \ldots \odot (w_n \odot q_1^{a_{1n}} \odot q_2^{a_{2n}} \odot \ldots \odot q_d^{a_{dn}})^{u_n} \\ = & \bigoplus\nolimits_{u_1+\ldots+u_n=m} (w_1^{u_1} \odot \ldots \odot w_n^{u_n}) \odot q_1^{a_{11}u_1+\ldots+a_{1n}u_n} \odot \ldots \odot q_d^{a_{d1}u_1+\ldots+a_{dn}u_n} \\ = & \bigoplus\nolimits_{u_1+\ldots+u_n=m} (w \cdot u) \odot q^{A\cdot u} \\ = & \bigoplus\nolimits_{b\in\mathbb{N}^d} (\min\nolimits_{A\cdot u=b}\{w \cdot u\}) \odot q^{A\cdot u} \end{aligned}$$

where $q^v = q_1^{v_1} \odot q_2^{v_2} \odot \ldots \odot q_n^{v_n}$ and the last equality involves collecting the terms containing $q^{A\cdot u}$. Thus, we see that the optimal value of (2.4) is the coefficient of the monomial $q^b = q_1^{b_1} q_2^{b_2} \ldots q_d^{b_d}$ in the above expression, which is the $m$th tropical power of the expression (2.5).

## Question 5

Computing by hand, we see that

$$D_G^2 = D_G^3 = \begin{bmatrix} 0 & 3 & 7 & 7 & 5 \\ 3 & 0 & 5 & 4 & 2 \\ 7 & 5 & 0 & 1 & 5 \\ 7 & 4 & 1 & 0 & 6 \\ 5 & 2 & 5 & 6 & 0 \end{bmatrix}$$

Thus, $D_G^4 = D_G^3 \odot D_G = D_G^2 \odot D_G = D_G^3$. Inductively, we have $D_G^2 = D_G^3 = D_G^4 = D_G^5 = D_G^6$. Since the entry $d_{ij}^{(n)}$ of $D_G^n$ gives the length of the shortest path from vertex $i$ to vertex $j$ using at most $n$ edges, the equality of the above matrices imply that the shortest path between any two vertices uses at most 2 edges.

The tropical determinant of $D_G$ is 0. It is the minimum of the sum of 5 entries in the matrix chosen such that there is exactly one entry in each row and each column. Since all the entries are positive, such a sum is at least 0. In this case, this minimum is achieved, by choosing the diagonal elements of the matrix. Another way of looking at the tropical determinant is this: it corresponds to choosing 5 directed edges in the graph G such that each vertex occurs exactly once as the source of some edge and also exactly once as the destination of some edge, with the aim of minimizing the total length of the edges. Here, the optimal choice lies in picking the edges that start and end with the same vertex, giving a total length of 0.

## Question 6

As mentioned in the proof, the entry $d_{ij}^{(2)}$ of the matrix $D \odot D$ equals $\min S_{ij}$ where $S_{ij} = \{d_{ik} + d_{kj} | 1 \leq k \leq n\}$. Thus, if $D \odot D = D$, then $d_{ij} = d_{ij}^{(2)} = \min Sij \leq d_{ik} + d_{kj}$ for all $i, j, k \in [n]$, so the matrix $D$ represents a metric. Conversely, if $D$ represents a metric, then given any $i, j \in [n]$, $d_{ij} \leq d_{ik} + d_{kj}$ for all $1 \leq k \leq n$, so $d_{ij} \leq \min S_{ij} = d_{ij}^{(2)}$. But, one of the elements in $S_{ij}$ is $d_{ij} = 0 + d_{ij} = d_{ii} + d_{ij}$, where $d_{ii} = 0$ because $D$ is the matrix of a dissimilarity map. Thus, $d_{ij}^{(2)} = \min S_{ij} \leq d_{ij}$. Combining the two inequalities gives $d_{ij}^{(2)} = d_{ij}$ for all $i, j \in [n]$, so $D \odot D = D$.

Consider the four vertices of a unit square, labeling them $1, 2, 3, 4$ in a counter-clockwise direction. Let $d(i, j)$ be the Euclidean distance between vertex $i$ and $j$. Then, $d$ is a metric since the Euclidean distance satisfies the triangle inequality. The matrix $D$ that represents this metric and its tropical square is

$$D = \begin{bmatrix} 0 & 1 & \sqrt{2} & 1 \\ 1 & 0 & 1 & \sqrt{2} \\ \sqrt{2} & 1 & 0 & 1 \\ 1 & \sqrt{2} & 1 & 0 \end{bmatrix}, \quad D \odot D = \begin{bmatrix} 0 & 1 & \sqrt{2} & 1 \\ 1 & 0 & 1 & \sqrt{2} \\ \sqrt{2} & 1 & 0 & 1 \\ 1 & \sqrt{2} & 1 & 0 \end{bmatrix}$$

So, indeed, $D \odot D = D$. On the other hand, consider the dissimilarity map $d_G$ represented by the matrix $D_G$ in Question 5. This map is not a metric because $d_G(1, 5) = 11 > 5 = d_G(1, 2) + d_G(2, 5)$. Indeed, we also see that $D_G \odot D_G \neq D_G$.

## Question 7

Arnold Levine, "The Evolution of Influenza Viruses in the 20th and 21st Centuries", 12 Feb (Mon).

The lecture discusses two reasons for the effectiveness of the influenza virus in infecting the human population: genetic drift and genetic shift. It points out how mathematical analysis of decade-old data on the virus revealed interesting patterns in the evolution of the virus and accounted for the pandemics that occurred over the past century.

The lecture begins by introducing the biology of viruses. Viruses are usually ineffective in infecting human populations because once it hits a person, he develops immunity against the virus which protects him from future infections. One exception to this rule is the HIV virus, which succeeds by attacking the immune system of the body. The other exception is the influenza virus which can hit a person winter after winter or cause a pandemic across the globe. The influenza virus has 8 chromosomes and 2 proteins on its surface, called HA and NA, which help it in attacking the respiratory system.

There are many different strains of the influenza virus, some affecting only humans, others birds, and so on. Flu strains can be classified according to the type of HA and NA protein that it produces. Usually, strains which spread easily between humans do not spread between birds. However, when a human and bird strain are both present in a human body, there is a possibility that the two strains switch chromosomes with each other, producing a new viral strain which may be extremely dangerous and infectious. This is known as genetic shift. Studying the major flu pandemics of the past century reveals that they were caused by new strains due to genetic shift. The current fear is that the H5N1 bird flu virus may switch chromosomes with a human flu virus, producing a new H5 strain that sparks a new pandemic.

Viruses also escape the immunity that humans develop against it by slightly mutating themselves so that the body no longer recognizes it. This is known as genetic drift. The key reason for high mutation rate in viruses compared to that in animals is because their chromosomes are made of RNA which replicates with much lower fidelity than DNA. A recent mathematical study of existing data shows that the human body has developed a way of countering the flu virus that is not found in birds. It modifies the RNA of the virus by changing the G nucleotide to A's, thus rendering the virus less effective. This is the reason why there are so many strains of bird flu but only a few human strains.

The main insight that the speaker hopes to share with the audience is that there is a wealth of information in existing biological data waiting to be mined with mathematical techniques. New insights into the nature of genetic shift and genetic drift would not be possible without analyzing large amounts of population data. These new insights can be important in the development of new strategies against future pandemics, saving thousands, or even millions, of lives.

## MATLAB code for Question 4

```
function [pvect] = M127hw2q3b()
% This function returns a 1296 x 10010 matrix with each
% row corresponding to a coordinate polynomial and each
% column corresponding to some monomial in the polynomials.

pvect = [];

% run through all \tau_1, \tau_2, \tau_3, \tau_4
for a = 1:6
for b = 1:6
for c = 1:6
for d = 1:6
    pvect = [pvect; p(a, b, c, d)];
end
end
end
end

function [coef] = p(a, b, c, d)
% This function computes the 1x10010 coefficients
% of the monomials in the expansion of p_{abcd}.
% We manually expand the polynomials, rather than
% use MATLAB's symbolic library.

nnlist = [1 2 4 2 4 6 4 2 2 4 6 4 2 4 2 1]';
sslist = [ 1  0  0  0  0  0;
           1 -1  0  0  0  0;
           1 -1 -1  1  0  0;
          -1  1  0  0  0  0;
           1 -1 -1  1  0  0;
           1 -1  1  2 -2 -1;
           1 -1 -1  1  0  0;
          -1  1  0  0  0  0;
          -1  1  0  0  0  0;
           1 -1 -1  1  0  0;
          -1  2  1 -1 -2  1;
           1 -1 -1  1  0  0;
          -1  1  0  0  0  0;
           1 -1 -1  1  0  0;
          -1  1  0  0  0  0;
           1  0  0  0  0  0];
ttlist = [ 0  0  0  0  0  0;
           2  0  0  0  0  0;
           5  2  4  1  0  0;
           1  4  0  0  0  0;
           5  2  4  1  0  0;
           9  1  2  4  5  8;
           3  4  7  8  0  0;
           3  7  0  0  0  0;
           1  2  0  0  0  0;
           5  2  4  1  0  0;
           3  4  7  5  8  9;
           3  4  7  8  0  0;
           3  4  0  0  0  0;
           3  4  7  8  0  0;
           6  7  0  0  0  0;
           6  0  0  0  0  0];

coef = zeros(1, 10010);

% running through \sigma_1, \sigma_2, \sigma_3, \sigma_4
for i = 1:2
for j = 1:2
for k = 1:2
for l = 1:2

    abcd = [a b c d];
    ijkl = [i j k l];

    ii = sum((ijkl-1).*[8 4 2 1])+1;
    nn = nnlist(ii);
    ss = sslist(ii, :);
    tt = ttlist(ii, :);

    expsize = zeros(1, 4);
    expcoef = zeros(6, 4);
    expsign = zeros(6, 4);

    for col = 1:4
        if (abcd(col) == 6)
            expsize(col) = 6;
            if (ijkl(col) == 1)
                expcoef(:,col) = [11 1 2 3 4 5]';
            else
                expcoef(:,col) = [11 6 7 8 9 10]';
            end
            expsign(:,col) = [1 -1 -1 -1 -1 -1]';
        else
            expsize(col) = 1;
            if (ijkl(col) == 1)
                expcoef(1,col) = abcd(col);
            else
                expcoef(1,col) = abcd(col)+5;
            end
            expsign(1,col)=1;
        end
    end

    for w = 1:expsize(1)
    for x = 1:expsize(2)
    for y = 1:expsize(3)
    for z = 1:expsize(4)
```

```
        curpower = zeros(1, 11);
        cursign = expsign(w,1)*expsign(x,2)*expsign(y,3)*expsign(z,4);
        curpower(expcoef(w,1)) = curpower(expcoef(w,1))+1;
        curpower(expcoef(x,2)) = curpower(expcoef(x,2))+1;
        curpower(expcoef(y,3)) = curpower(expcoef(y,3))+1;
        curpower(expcoef(z,4)) = curpower(expcoef(z,4))+1;

        front_index = 1;
        sumleft = 3;
        for ii = 1:10
            sumleft = sumleft - curpower(ii);
            for jj = 0:sumleft
                front_index = front_index+numpart(jj,11-ii);
            end
        end

        for ii = 1:nn
            coef(front_index+tt(ii)*1001) = coef(front_index+tt(ii)*1001)+cursign*ss(ii);
        end

    end
    end
    end
    end

end
end
end
end

function [m] = numpart(k, n)
% number of ordered partitions of k into n parts.

if (k == -1)
    m = 0;
else
    m = nchoosek(n+k-1, k);
end
```