COMP 561 Assignment 2

Jingyuan Wang 260860682

October 15, 2019

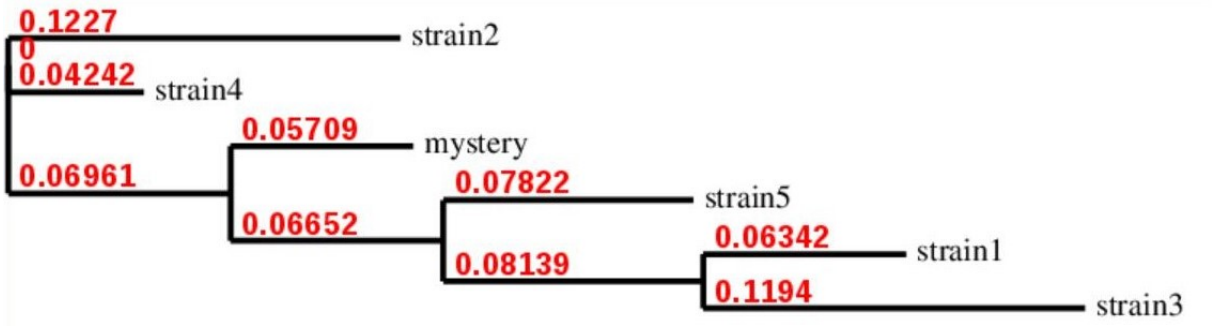
Question 1

a. When performing blast with the given query sequence, with parameters of word size 16 and Expect Value 10, the attained result is as followed.

Description	Max Score	Total Score	Query Cover	**E value**	Per. Ident	Accession
Cyprinus carpio genome assembly common carp genome, scaffold 000001635	43.6	43.6	25%	1.7	93.33%	LN592857.1
Glarea lozoyensis ATCC 20868 P-loop containing nucleoside triphosphate hydrolase mRNA	43.6	43.6	19%	1.7	100.00%	XM_008079891.1
Thalassophryne amazonica genome assembly, chromosome: 23	41.7	41.7	18%	5.9	100.00%	LR722988.1
Myripristis murdjan genome assembly, chromosome: 21	41.7	41.7	18%	5.9	100.00%	LR597570.1
Sphaeramia orbicularis genome assembly, chromosome: 15	41.7	41.7	20%	5.9	96.00%	LR597472.1
PREDICTED: Felis catus small proline-rich protein 2G-like (LOC105260168), mRNA	41.7	41.7	20%	5.9	96.00%	XM_023249723.1
Macaca mulatta BAC CH250-534M21 (Children's Hospital Oakland Research Institute Rhesus macaque Adult Male BAC Library) comple	41.7	41.7	25%	5.9	90.32%	AC210639.3
Macaca mulatta BAC CH250-460J1 (Children's Hospital Oakland Research Institute Rhesus macaque Adult Male BAC Library) complete	41.7	41.7	25%	5.9	90.32%	AC210638.2

After checking with taxomony of each organism, it is found that "Glarea lozoyensis ATCC 20868 P-loop containing nucleoside triphosphate hydrolase mRNA" which is a kind of ascomycetes, is very likely to have caused the infection. This fungus contributes to produce pneumocandin B0, thus the name of disease should be pneumonia.

b. The picture below shows the phylogenetic tree with five various strains as well as our pathogen and branch lengths are proportional to divergence of different sequences.



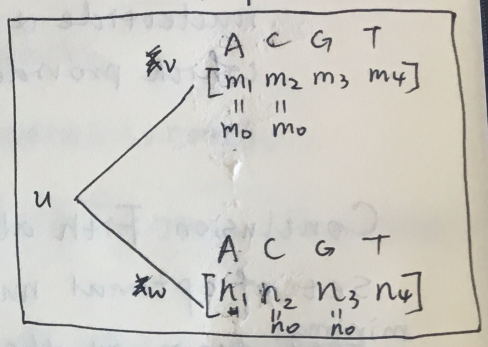
Let m denotes the pathogen named mystery and $s1$ to $s5$ represents strain1 to strain5. We calculate divergence between m and each other strains as equation 1. It is proved that strain4 has more similarity with our pathogen strain and, as a consequence, treatment for strain4 is more likely to be appropriate for the patient.

$$\begin{aligned} D(m, s1) &= 0.05709 + 0.06652 + 0.08139 + 0.06342 = 0.26842 \\ D(m, s2) &= 0.1227 + 0.06961 + 0.05709 = 0.2494 \\ D(m, s3) &= 0.05709 + 0.06652 + 0.08139 + 0.1194 = 0.3244 \\ D(m, s4) &= 0.04242 + 0.06961 + 0.05709 = 0.16912 \\ D(m, s5) &= 0.05709 + 0.06652 + 0.07822 = 0.20183 \end{aligned} \tag{1}$$

Question 2

In Fitch algorithm, Score(u) is the minimum parsimony score of subtree rooted at node u, and X_u is the set of nucleotides ~~which~~ with which node u is able to achieve Score(u). Proofs are as follows: (nucleotide at u is α)

I. For leaf u, it is obvious that $X_u = \{x\}$ where x is at u. Therefore Score(u) = 0



II. For internal u.

① When $X_w \cap X_v \neq \emptyset$:

△ Suppose a condition as → where for tree v, nucleotides A and C can help achieve score(v) = m_0, then $m_3 \geq m_0+1$, $m_4 \geq m_0+1$. ($X_v = \{A, C\}$)

similarly, score(w) = n_0, $n_1 \geq n_0+1$, $n_4 \geq n_0+1$ ($X_w = \{C, G\}$)

To perform Sankoff alg, there are 3 situations.

a) if $\alpha \in X_v \cap X_w$, $F_u[C] = 0 + m_0 + 0 + n_0 = m_0 + n_0$.

b) if $\alpha \in X_v \cup X_w - X_v \cap X_w$, $F_u[A] = 0 + m_0 + \min(n_1, n_0+1)$
$= m_0 + n_0 + 1$
(or similarly $F_u[G] = m_0 + n_0 + 1$)

c) if $\alpha \in 1-(X_v \cup X_w)$, $F_u[T] = \min(m_0+1, m_4) + \min(n_0+1, n_4)$
$= m_0 + n_0 + 2$

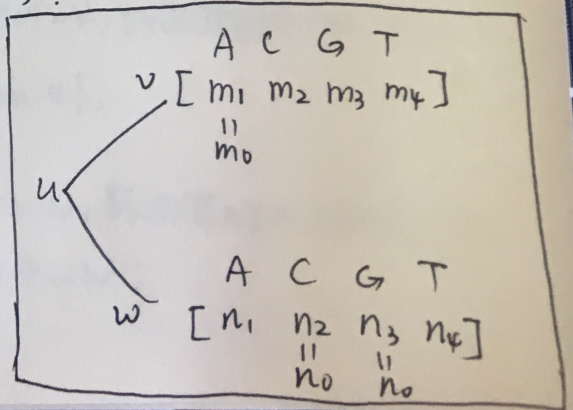
∴ Under this circumstance, a nucleotide $\alpha \in X_v \cap X_w$ will always achieve score(u), with value of $m_0 + n_0$ = score(v) + score(w).

② When $X_w \cap X_v = \emptyset$

Same as above, try to use Sankoff alg.

a) if $\alpha \in X_v$

$F_u[A] = m_0 + \min(n_1, n_0+1)$
$= m_0 + n_0 + 1$



b) if $\alpha \in X_W$

$F_u[C] = n_0 + 0 + \min(m_2, m_0+1) = m_0 + n_0 + 1$

$F_u[G] = m_0 + n_0 + 1$

c) if $\alpha \in 1 - X_V \cup X_W$

$F_u[T] = \min(n_4, n_0+1) + \min(m_4, m_0+1) = m_0 + n_0 + 2$

∴ Minimal parsimony score is achieved when nucleotide $\alpha \in X_V$ or $\alpha \in X_W$ (i.e. $\alpha \in X_V \cup X_W$) which provides score(u) = $m_0 + n_0 + 1$ = score(w) + score(v) + 1

Conclusion: Fith algorithm keeps track of a set of optimal nucleotides at node u and the minimal ~~best~~ score at the same time. The two algorithms are identical and Sankoff algorithm being a general representation of Fitch alg.

Question 3

a) Pseudo code:

```
// For each node u, maintain a matrix Fu[K][M].
// where Fu[i][m] denotes for the ith traits, the
// minimal possible Σ_{(u,v)∈E(Tu)} |Du-Dv(i)| with i-th traits has
// a value of m at node u.
for i = 1→K:  // for each traits
    Froot[i] = calculateParsScore(i, root);
    Du[i] = min(Fu[i][0], Fu[i][1], Fu[i][2], ..., Fu[i][M]);
return Du;
int[M+1]
function calculateParsScore(int i, node u){
    if(u→leftChild != null)
        Fv[i] = calculateParsScore(i, u→leftChild);
    if(u→rightChild != null)
        Fw[i] = calculateParsScore(i, u→rightChild);
    if(u→leftChild == null && u→rightChild == null)
        Fu[i] = [+∞] * M;
        Fu[i][u.value(i)] = 0; // Fu[i] = [∞, +∞, +∞, 0, ..., +∞]
        return Fu[i];
    for m = 0→M:  min_v = Fv[i][0] + |m-0|;
        for n = 0→N:
            min_v = min(min_v, Fv[i][n] + |m-n|);
        min_w = Fw[i][0] + |m-0|;
        for n = 0→N:
            min_w = min(min_w, Fw[i][n] + |m-n|);
        Fu[i][m] = min_v + min_w;
    return Fu[i];
```

b) $O(k \cdot (n-1) \cdot M \cdot M) = O(k \cdot n \cdot M^2)$

c) If values of different traits vary significantly, we will need to waste time on calculating integer values that would never be chosen.
eg. for a trait ranges between 0 to 50, the value of internal node u can not exceed 50 in order to achieve minimized $\sum_{(u,v) \in E(T_w)} |D_u(i) - D_v(i)|$. However since M can be 1500 (according to example in the question), a total of $n \cdot (1500-50) \cdot (1500-50)$ times operations are wasted.

Therefore, we can first perform a BFS or DFS to check the max value of i-th trait as M_i and then use the algorithm above but maintain a different size of F_u for each trait. (i.e. the ith trait has an array of M_i and only need to calculate values within this range)