Computational Biology – HW1

JK00
王正宇
2010011344

Problem 1

Problem Specification: given n, and n numbers a[1], …, a[n]. We are required to return a sequence i_1, i_2, …, i_k, such that 1<=i_1<i_2<…<i_k<=n and a[i_1]<a[i_2]<…<a[i_k], and k is as large as possible.
Main Idea: We decompose the question of finding the longest increasing subsequence into n interrelated questions, i.e., for 1<=i<=n, we want to find the longest increasing subsequence of a[i], a[i+1],…, a[n], where the first element in the subsequence must be a[i]. We denote the answer sequence is ans[i]. Apparently, the answer to the question when i=n is just a[n] with length 1, i.e., ans[n]=”n”. For 1<=i<n, the answer is equal to “i ” + ans[j] with the maximum ans[j].length, where j takes over {j|j>i & a[j]>a[i]}. The answer to this question is just the longest ans[i].
Answer: The proposed algorithm is described below in pseudocode:

```
ans[n] := “n”;
for i = n-1 downto 1 do
ans[i] := “”;
for j = i+1 to n do
        if (a[i]<a[j] && ans[i].length<ans[j].length) ans[i] := ans[j];
ans[i] := “i ” + ans[i];
endfor
j := argmax_{1<=i<=n}(ans[i].length);
return ans[j];
```

Notice that the algorithm’s time & space complexity are both O(n^2). The space complexity can be easily reduced to O(n) if for each ans[i], we only need to record ans[i].length and ans[i]’s second element. The time complexity can be reduced to O(n log n) by some data structure technique. For example, firstly, discretization (O(n log n) time) the values in a such that it becomes a permutation of n. Then we maintain (O(n log n) time in all) ans[i].length as a interval tree supporting dynamically increasing one element and querying the maximum number (return the index) of an interval.
Discussion: This problem, the longest increasing subsequence problem can be reduced to the problem discussed in the class, i.e., the problem of finding the longest common subsequence of two sequences. The reduction is quite simple, that is, the answer to the question is the longest common subsequence between a and a’, where a’ is a ascending reordering of a.

Problem 2

Main Idea: We do this problem according to the formula given in the second lecture.
Answer:

The A matrix:

20	14	7	10
14	7	12	9
7	12	10	7
10	9	7	5

The P matrix:

0.125	0.0875	0.04375	0.0625
0.0875	0.04375	0.075	0.05625
0.04375	0.075	0.0625	0.04375
0.0625	0.05625	0.04375	0.03125

The q's

0.31875
0.2625
0.225
0.19375

The S matrix:

0.207255	0.0447359	-0.494261	0.0119461
0.0447359	-0.454255	0.238892	0.100742
-0.494261	0.238892	0.210721	0.00357782
0.0119461	0.100742	0.00357782	-0.183363

If we multiply it by 10 and take the floor round, we will bet one possible substitution matrix

2	0	-5	0
0	-5	2	1
-5	2	2	0
0	1	0	-2

Problem 3

Answer:

(3a) My student ID number is 2010011344, and the last four digit is "1344". After converting it into base-four-number, it is "111000", whose corresponding DNF sequence is "TTTAAA". My birthday is 19911116, and the last four digit is "1116". After converting it into base-four-number, it is "101130", whose corresponding DNF sequence is "TATTCA".

Dynamic programming table:

		T	T	T	A	A	A
	0	-3, ←	-6, ←	-9, ←	-12, ←	-15, ←	-18, ←
T	-3, ↑	1, ↖	-2, ↖←	-5, ↖←	-8, ←	-11, ←	-14, ←
A	-6, ↑	-2, ↑	0, ↖	-3, ↖←	-4, ↖	-7, ↖←	-10, ↖←
T	-9, ↑	-5, ↖↑	-1, ↖	1, ↖	-2, ←	-5, ↖←	-8, ↖←
T	-12, ↑	-8, ↖↑	-4, ↖↑	0, ↖	0, ↖	-3, ↖←	-6, ↖←
C	-15, ↑	-11, ↑	-7, ↑	-3, ↑	-1, ↖	-1, ↖	-4, ↖←
A	-18, ↑	-14, ↑	-10, ↑	-6, ↑	-2, ↖	0, ↖	0, ↖

Therefore the best alignment score is 0, and the alignment is

TTTAAA

|||| ||

TATTCA

(3b) The last three digits of my birthday is “116”, which is “1310”, i.e., “TCTA”. The following is the table for local alignment:

		T	T	T	A	A	A
	0	0	0	0	0	0	0
T	0	1, ↖	1, ↖	1, ↖	0	0	0
C	0	0	0, (↖)	0, (↖)	0, (↖)	0	0
T	0	1, ↖	1, ↖	1, ↖	0	0	0
A	0	0	0, (↖)	0, (↖)	2, ↖	1, ↖	1, ↖

The best local alignment score is 2, and there are two possible alignments:

```
(1) TTTAAA
    ||||
    TCTA
(2) TTTAAA
      ||
    TCTA
```

Problem 4

Answer:

(4a) Mine is TTTAAA, the other possible four are, TTCGTC, TTGGAC, TTCCAC, TTCAGG. We number them from 1 to 5 accordingly.

The matrix of the alignment scores is:

	1	2	3	4	5
1	6	0	-2	0	0
2	0	6	2	2	0
3	-2	2	6	2	-2
4	0	2	2	6	-2
5	0	0	-2	-2	6

We can find that gene 2 is the one which maximizes the summation of similarities, i.e., has the maximal average alignment score. Therefore, we pick gene 2 as the center sequence. The overall alignments according to the star algorithm is just simply align all the other genes with gene 2.

```
         TTTAAA
            |
TTCCAC---TTCGTC--- TTCAGG
            |
         TTGGAC
```

(4b) Firstly, we need to convert the alignment scores into distances.

	1	2	3	4	5
1		1.2	1.4	0.8	0.5
2	1.2		1.3	1.1	1
3	1.4	1.3		0.9	1
4	0.8	1.1	0.9		0.7
5	0.5	1	1	0.7	

The algorithm is a greedy algorithm, which I think is quite like Huffman tree's construction. The pseudocode:

```
List = {1}, {2}, ..., {n};
for k = 1 to n-1 do
    find two different elements i,j in List such that Dis(i,j) is the maximum;
    union i,j, namely delete i, j from the List, and add {i,j} to List;
    for all other elements in List, say h, Dis(k,{i,j})=min{Dis(k,i}, Dis(k,j)}
endfor
```

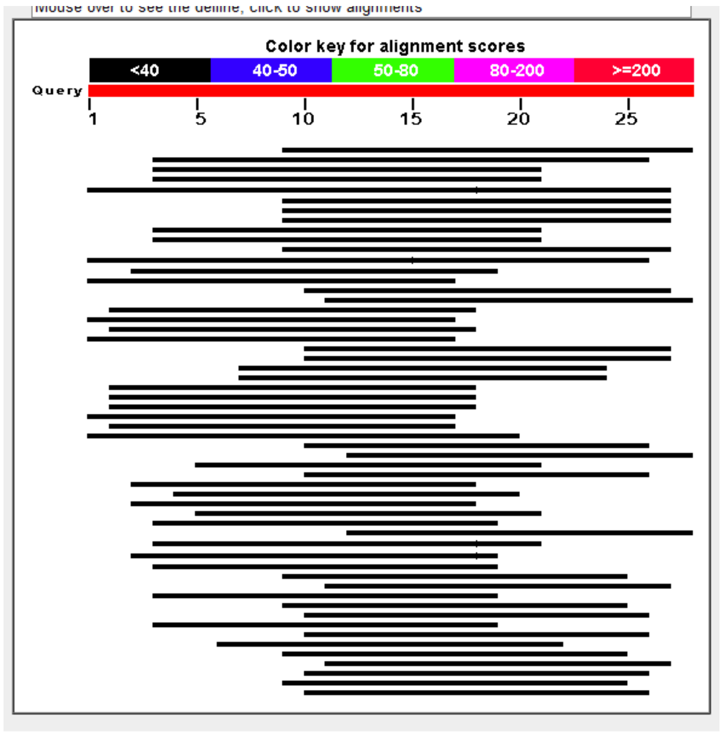
The tree:

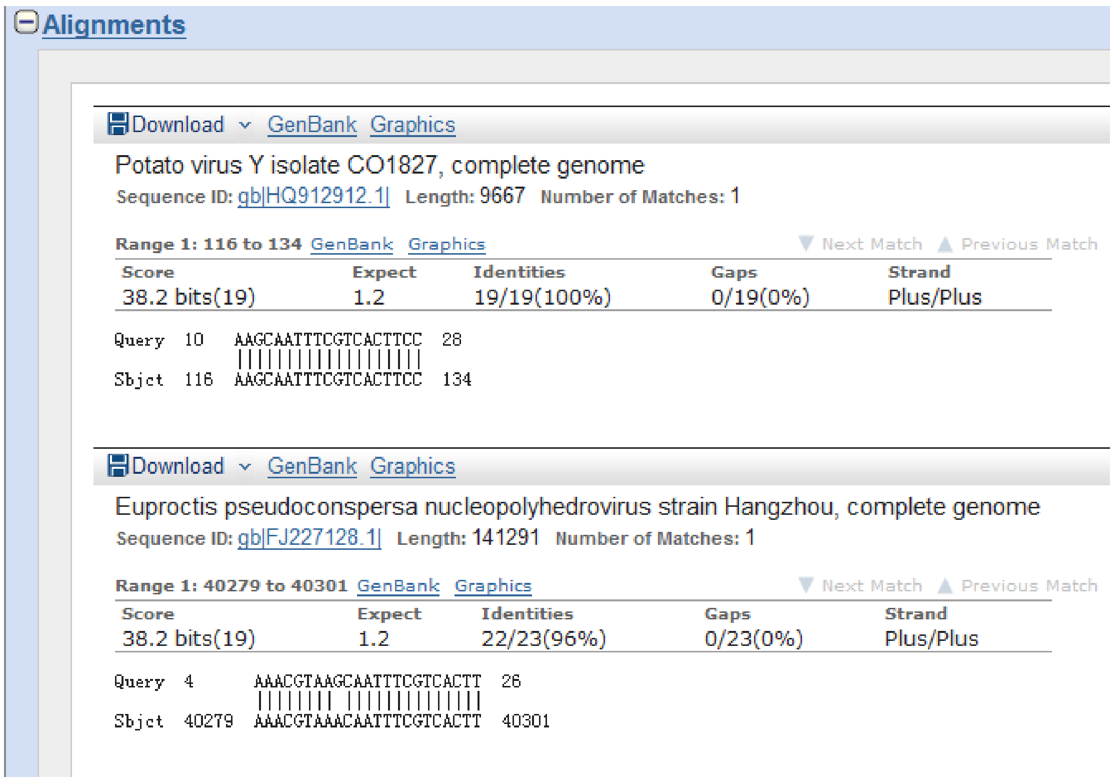
```
            {1,2,3,4,5}
               /     \
          {1,2,3,4}   5
           /     \
       {1,2,3}    4
        /       \
    {1,3}        2
   /       \
  1          3
```

Problem 5

Answer: Maybe "TTTAAA" is too short, it does not find anything similar. So I tried another sequence called "TTTAAACGTAAGCAATTTCGTCACTTCC".

Below are some parts of the outcome:


Mouse over to see the defline, click to show alignments
Color key for alignment scores
<40
40-50
50-80
80-200
>=200
Query
1
5
10
15
20
25



Alignments

Download GenBank Graphics

Potato virus Y isolate CO1827, complete genome

Sequence ID: gb|HQ912912.1| Length: 9667 Number of Matches: 1

Range 1: 116 to 134 GenBank Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Identities	Gaps	Strand
38.2 bits(19)	1.2	19/19(100%)	0/19(0%)	Plus/Plus

```
Query  10     AAGCAATTTCGTCACTTCC  28
              |||||||||||||||||||
Sbjct  116    AAGCAATTTCGTCACTTCC  134
```

Download GenBank Graphics

Euproctis pseudoconspersa nucleopolyhedrovirus strain Hangzhou, complete genome

Sequence ID: gb|FJ227128.1| Length: 141291 Number of Matches: 1

Range 1: 40279 to 40301 GenBank Graphics ▼ Next Match ▲ Previous Match

Score	Expect	Identities	Gaps	Strand
38.2 bits(19)	1.2	22/23(96%)	0/23(0%)	Plus/Plus

```
Query  4      AAACGTAAGCAATTTCGTCACTT  26
              |||||||| ||||||||||||||
Sbjct  40279  AAACGTAAACAATTTCGTCACTT  40301
```