Εισαγωγή στην Βιοπληροφορική

Χειμερινό εξάμηνο 2017-2018

Χατζηγεωργίου Άρτεμις

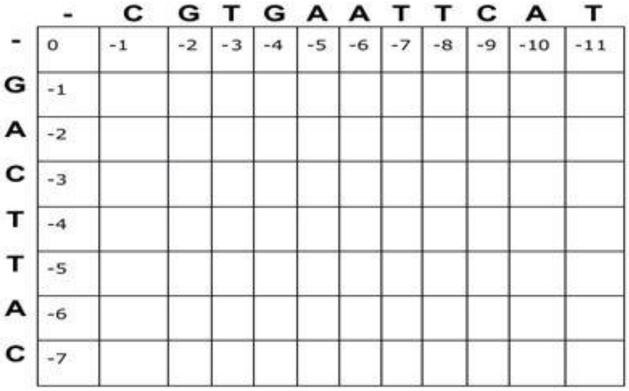
Παπαδομανωλάκης Ελευθέριος 1634

Άσκηση 3: Global Alignment Sequences

1. Align the following sequences with global alignment ACGGTAG CCTAAG
2. Return the DPA matrix
3. the BackTrack matrix
4. the path
5. the alignment score

Method used to find out the best possible alignment in this exercise: Needleman-Wunsch Algorithm

1. **Initialization Step:**
   We initialize the 1st row and 1st column of the array(hashmap with i,j as keys) with the gap penalty * pos(i or j)
   Etc. (with gap=-1)



	-	C	G	T	G	A	A	T	T	C	A	T
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
G	-1											
A	-2											
C	-3											
T	-4											
T	-5											
A	-6											
C	-7											

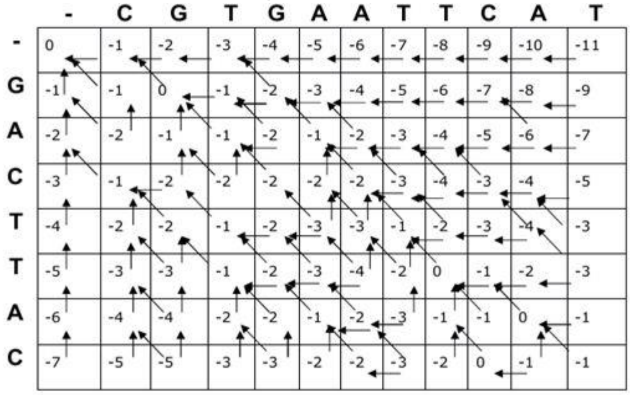
2. **Matrix Fill Step:**
   Starting from 1,1(G-C) we find the best possible score for every cell.
   The function for every fill is:
   $M_{i,j} = \text{Maximum}(M_{i-1,j-1} + S_{i,j}\ ,\ M_{i,j-1} + W\ ,\ M_{i-1,j} + W)$

Where:

$S_{i,j}$ = Match or Mismatch
W = Gap penalty

We also keep track, in another hash map, the of each source for the score in every cell, in order to use in in Trace Back Step.

We then have something like this:



	-	C	G	T	G	A	A	T	T	C	A	T
-	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
G	-1	-1	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-2	-2	-1	-1	-2	-1	-2	-3	-4	-5	-6	-7
C	-3	-1	-2	-2	-2	-2	-2	-3	-4	-3	-4	-5
T	-4	-2	-2	-1	-2	-3	-3	-1	-2	-3	-4	-3
T	-5	-3	-3	-1	-2	-3	-4	-2	0	-1	-2	-3
A	-6	-4	-4	-2	-2	-1	-2	-3	-1	-1	0	-1
C	-7	-5	-5	-3	-3	-2	-2	-3	-2	0	-1	-1

Arrows indicate the 'information' we can extract from the hash map mentioned above.

3. **Trace back Step:**

Recursively we find the best possible path from the root of the tree created by arrows, from (i_{max},j_{max}) till (i=0 or j=0).

When we reach an intersection of 2 paths we check both possibilities and in the end keep the one with the maximum score. In the end we have a single line path where we know every possible best path (1>=possible paths), of course all returned paths have the score.

Now that we now the best possible alignment(s) we are ready to build the output

Starting from the end of the path(i=0 or j=0) we iterate all cells where in each transition from one cell to another we find the element(in each respective position) of each sequences that we wanted to align according to the following rules:

- **Transition was right**

  The char for that position for the 1st sequence is '-' and the char of the 2nd is the jth char where j is the position before the transition.

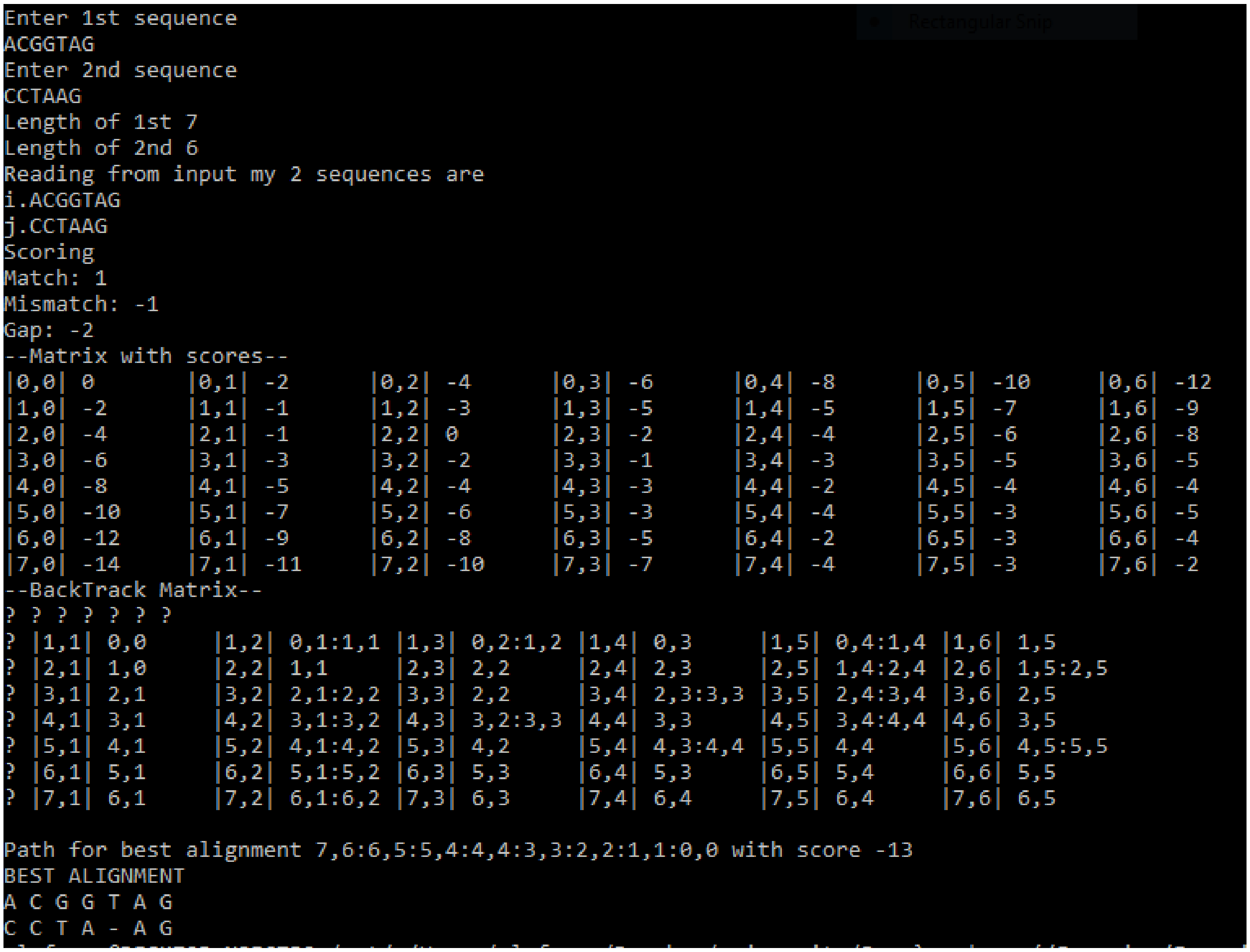
- **Transition was down**

  The char for that position for the 2nd sequence is '-' and the char of the 1st is the ith char where i is the position before the transition.

- **Transition was right-down**

  The char for that position for each sequence is the respective char for every sequence in the position before the transition.

Output of the script given:



```
Enter 1st sequence
ACGGTAG
Enter 2nd sequence
CCTAAG
Length of 1st 7
Length of 2nd 6
Reading from input my 2 sequences are
i.ACGGTAG
j.CCTAAG
Scoring
Match: 1
Mismatch: -1
Gap: -2
--Matrix with scores--
|0,0| 0       |0,1| -2      |0,2| -4      |0,3| -6      |0,4| -8      |0,5| -10     |0,6| -12
|1,0| -2      |1,1| -1      |1,2| -3      |1,3| -5      |1,4| -5      |1,5| -7      |1,6| -9
|2,0| -4      |2,1| -1      |2,2| 0       |2,3| -2      |2,4| -4      |2,5| -6      |2,6| -8
|3,0| -6      |3,1| -3      |3,2| -2      |3,3| -1      |3,4| -3      |3,5| -5      |3,6| -5
|4,0| -8      |4,1| -5      |4,2| -4      |4,3| -3      |4,4| -2      |4,5| -4      |4,6| -4
|5,0| -10     |5,1| -7      |5,2| -6      |5,3| -3      |5,4| -4      |5,5| -3      |5,6| -5
|6,0| -12     |6,1| -9      |6,2| -8      |6,3| -5      |6,4| -2      |6,5| -3      |6,6| -4
|7,0| -14     |7,1| -11     |7,2| -10     |7,3| -7      |7,4| -4      |7,5| -3      |7,6| -2
--BackTrack Matrix--
? ? ? ? ? ? ?
? |1,1| 0,0     |1,2| 0,1:1,1 |1,3| 0,2:1,2 |1,4| 0,3     |1,5| 0,4:1,4 |1,6| 1,5
? |2,1| 1,0     |2,2| 1,1     |2,3| 2,2     |2,4| 2,3     |2,5| 1,4:2,4 |2,6| 1,5:2,5
? |3,1| 2,1     |3,2| 2,1:2,2 |3,3| 2,2     |3,4| 2,3:3,3 |3,5| 2,4:3,4 |3,6| 2,5
? |4,1| 3,1     |4,2| 3,1:3,2 |4,3| 3,2:3,3 |4,4| 3,3     |4,5| 3,4:4,4 |4,6| 3,5
? |5,1| 4,1     |5,2| 4,1:4,2 |5,3| 4,2     |5,4| 4,3:4,4 |5,5| 4,4     |5,6| 4,5:5,5
? |6,1| 5,1     |6,2| 5,1:5,2 |6,3| 5,3     |6,4| 5,3     |6,5| 5,4     |6,6| 5,5
? |7,1| 6,1     |7,2| 6,1:6,2 |7,3| 6,3     |7,4| 6,4     |7,5| 6,4     |7,6| 6,5

Path for best alignment 7,6:6,5:5,4:4,4:3,3:2,2:1,1:0,0 with score -13
BEST ALIGNMENT
A C G G T A G
C C T A - A G
```

In each matrix we consider the elements of the columns to be each char of the 2nd sequence while the elements of the row of the 1st.
In Back Track matrix we save 1 or more sources (for score) with ':' in between them. This means that in every cell we have the origins of each score
The multiple paths are saved like this &i,j:i-1,j-1&, in this way we now each path from any node till the end of the path.

- while it's implemented to return more than one possible paths showing every alignment is not yet possible, you can just print the raw data returned.
- All vars are hardcoded to the script