21/09/2020

Name: Panwar Abhash Anil
Enroll no.: 2019MSBDA024
Course: M.Sc. Cs. (BDA)

Bio Informatics [MBD517]

Internal - II nd

Q1
Ans

→ Smith-Waterman algorithm: The algorithm is used to determine similar regions between two sequences of nucleic acid or protein sequences. The algorithm compares segments of all possible lengths and optimizes the similarity measure.

Smith-Waterman algorithm align two sequences by mismatches/matches, insertions and deletions. Both insertions and deletions are the operations that introduce gaps, which are represented by dashes.

Performance:

- Worst-Case : $O(mn)$
- Worst-case : $O(mn)$

Space Complexity

Algorithm:

① Initialize the matrix : (N×M)
The top row are left column are filled with zero.

② If the sub-alignment score become negative, restart the search

③ Fill the scoring matrix using

$$H_{ij} = \max \begin{cases} 0 \\ \max_{l \geq 1} \{ H_{i,j-l} - W_l \} \\ \max_{k \geq 1} \{ H_{i-k,j} - W_k \} \\ H_{i-1,j-1} + s(a_i, b_j) \end{cases}$$

$$(1 \leq i \leq n \ , \ 1 \leq j \leq m)$$

④ Traceback: Starting at the highest score in the scoring matrix H and ending at a matrix cell that has a score of 0, traceback based on the source of each score recursively to generate the best local alignment.

→ Seq1 : A C G T A T C G C G T A T A

Seq2: G A T G C T C T C G G A A A

Match = +1
Mismatch = 0
Gap penalty = -1

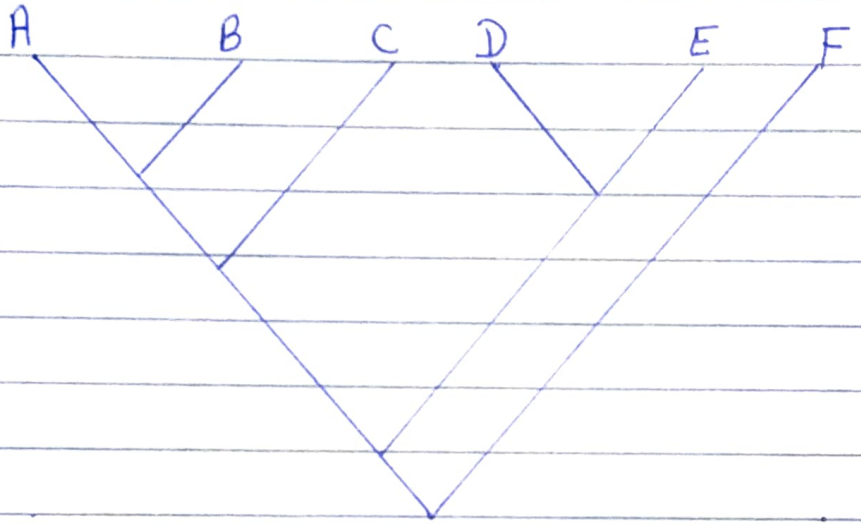
Alignment Score table is

		A	C	G	T	A	T	C	G	C	G	T	A	T	A
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0
A	0	1	0	0	1	1	0	0	0	1	0	1	1	0	1
T	0	0	1	0	1	1	2	1	0	0	1	1	1	2	1
G	0	0	0	2	1	1	1	2	2	1	1	1	1	1	2
C	0	0	1	1	2	1	1	2	2	3	2	1	1	1	1
T	0	0	0	1	2	2	2	1	2	2	3	3	2	2	1
C	0	0	1	0	1	2	2	3	2	3	2	3	3	2	2
T	0	0	0	1	1	1	3	2	3	2	3	3	3	4	3
C	0	0	1	0	1	1	2	4	3	4	3	3	3	3	4
G	0	0	0	2	1	1	1	3	5	4	5	4	3	3	3
G	0	0	0	1	2	1	1	2	4	5	5	5	4	3	3
A	0	1	0	0	1	3	2	1	3	4	5	5	6	5	4
A	0	1	1	0	0	2	3	2	2	3	4	5	6	6	6
A	0	1	1	1	0	1	2	3	2	2	3	4	6	6	7

Score = 7

Q2
Ans Given standard Newick format as

((A,B)C)(D,E))F



(Phylogenetic Tree)

Q3
Ans

Given Distance matrix:

→T-1:

Species	A	B	C	D
B	3	-	-	-
C	6	5	-	-
D	9	9	10	-
E	12	11	13	9

Using UPGMA and TDM for above table we get
Min distance = 3 of Species A and B

→ T-2 : By grouping A and B

Species	AB	C	D
C	5.5	-	-
D	9	10	-
E	11.5	13	9

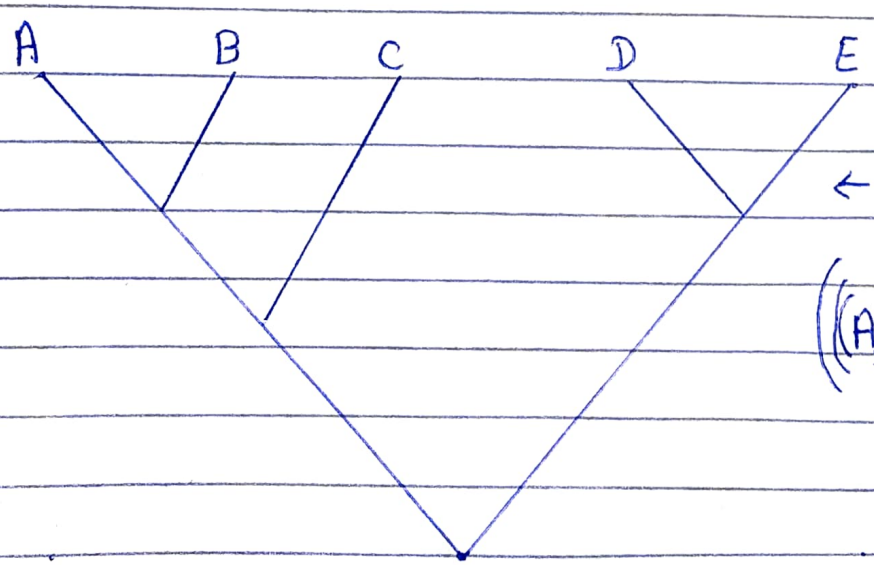
Now min. distance is 5.5 of AB and C

→ T-3 : By grouping AB and C

	ABC	D
D	9.5	-
E	12.25	9

→ T-4 :

	ABC
DE	10.625



← Phylogenetic Tree

((A,B)C)(D,E))

Q4) You have a file named "Protein_seq.fasta" which contain 100 protein sequences. The header of the protein sequence is like that "gi|5524211|gb|AAD44166.1| cytochrome b [Elephas maximus maximus]". Then

September 21, 2020

1 Writing a small Python code to identify the unique sequence?

[1]:
```
from Bio import SeqIO
import re

fasta_filename='Protein_seq.fasta'
fasta_seq=list()
fasta_des=list()

#loop to get sequence id,seq and description
for record in SeqIO.parse(fasta_filename, "fasta"):
    fasta_seq.append(str(record.seq))
    fasta_des.append(repr(record.description))

#create dataframe
df = pd.DataFrame(data={'fasta_des':fasta_des,'fasta_seq':fasta_seq}, index =␣
 ↪fasta_seq)

#remove duplicates and give unique sequences
df = df[~df.index.duplicated()]

#list of sequence
unique_seq=list(df.fasta_seq)

print("Total sequences are",len(fasta_seq))
print("Total unique sequences are",len(unique_seq))
```

```
<IPython.core.display.Javascript object>


Total sequences are 100
Total unique sequences are 100
```

2 Using Python code how you separate the Species from that unique sequences?

[2]:
```
#create list of unique sequences description
seq_desc=list(df['fasta_des'])

def ExtractSpecies(seq_desc):
    species=list()
    for i in range(len(seq_desc)):
        species.append(re.findall(r"\[(.*?)\]", seq_desc[i])[0])
    s=set(species)
    return(s)

species=ExtractSpecies(seq_desc)
print(species)
```

```
{'Staphylococcus', 'Bacillus cereus Q1', 'Ectocarpus sp. CCAP 1310/34', 'Hondaea
fermentalgiana', 'Staphylococcus aureus', 'Lingulodinium polyedra'}
```