BF527: Applications in Bioinformatics

Please prepare a typed report to submit through Blackboard. Please include printouts of all Python code that you write. Your code should follow the guidelines laid out in class, including commenting. Partial credit will be given for nonfunctional code that is logical and well commented. This assignment must be completed on your own.

HOMEWORK #3

See Blackboard for assignment and due dates

PROBLEM 3.1 (30%):

Consider the following biological sequence analysis tasks. For each task, **(A)** write the type of sequence analysis algorithm that would best accomplish the task, and **(B)** explain why. The algorithms have been discussed in class in Lectures 3-5. There may be more than one correct choice per task (you need **only provide one**).

i. Visualize a sequence inversion that occurred between two closely related viral strains.
ii. Given highly conserved ribosomal RNA sequences from five insect species, determine quantitatively which two species are the most closely related.
iii. Evaluate a new sequencing technology by comparing the results of a 300 BP read to a previous result based on Sanger sequencing. The beginning of the Sanger sequence is known to contain some "junk"—do not penalize for this.
iv. Identify a shared DNA binding domain in two otherwise unrelated proteins.

PROBLEM 3.2 (30%):
(A) Manually fill out a dynamic programming (DP) matrix for the pair of short DNA sequences below using the Needleman-Wunsch global alignment algorithm, and the scoring scheme below
(B) Include traceback arrows
(C) Report the score of the optimal alignment(s)
(D) Write out one possible optimal alignment
(E) How many equally optimal alignments are there (i.e., are there any ties)?

Scoring scheme: match = +2; mismatch = 0; gap = -1
Sequence 1 = **`GATGGCT`**
Sequence 2 = **`CAGGT`**

Note: You may opt to use/fill in the matrix template provided on Blackboard (also shown below) for this problem.

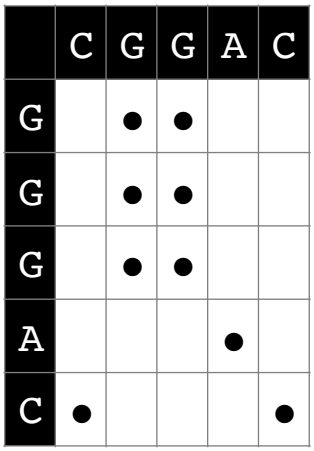
	-	G	A	T	G	G	C	T
-								
C								
A								
G								
G								
T								

PROBLEM 3.3 (40%):

(A) Write a python script called `dotcounter.py` that, given a pair of strings A and B, determines the total number of dots that would appear in the dot plot comparing the two strings and prints their coordinates. Your program does *not* need to draw the actual dot plot! For example, if the two strings are A = **GGGAC** and B = **CGGAC**, then your program would output:

```
Matches:
A1 and B2
A1 and B3
A2 and B2
...
A5 and B5
Total matches:
9
```

Refer to the dot plot below to check that this is correct.



(B) Run your program with the two input strings A = **ACTTGGCCAT** and B = **AGTAGCGCCT** and include the results (list of matches and number of total matches) in your homework.

(C) Thought question (*full points for explaining your logic, right or wrong*): How many dots would you expect to observe given two *random* DNA sequences of length ten? Do you think that the two sequences from **(B)** are random or related?

PROBLEM 3.4 Extra Credit (20%):

Write a python script called `global_alignment.py` that would extend `dotcounter.py` to incorporate Needleman-Wunsch algorithm to do a global alignment between two sequences.
Suggested scoring matrix: match=1, mismatch =0, gap = -1.