

### Assignment #3

Before you start work on this assignment, be sure to go through the material on dot-plots and global alignments in Lecture 6 on “Sequence Alignment II” and the 3 earlier Labs.

Fill out this document with your answers.

1. Draw a dot-plot to compare the following two sequences (use the table below in this LibreOffice document to make the dot-plot):

seq x: **ACTTAGC**

seq y: **GCTAGT**

reverse complement of x:

reverse complement of y:

Dot plot for sequence x and y:

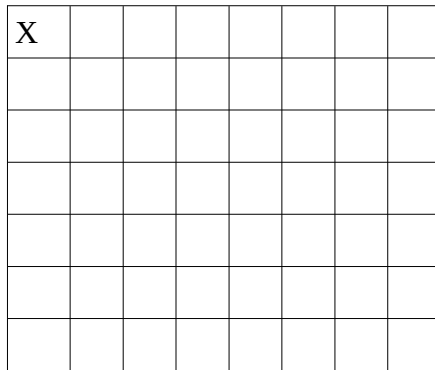
X								

Draw the 3 dot-plots to compare reverse complements:

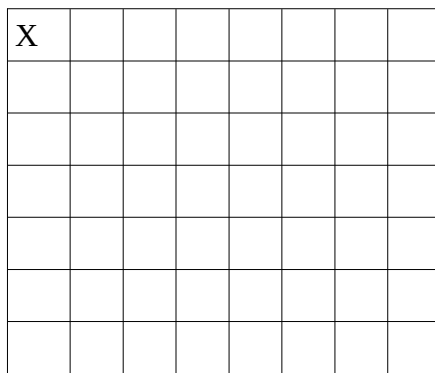
Dot-plot for rev-comp of x vs. y

X								

Dot-plot for rev-comp of x vs. rev-comp of y

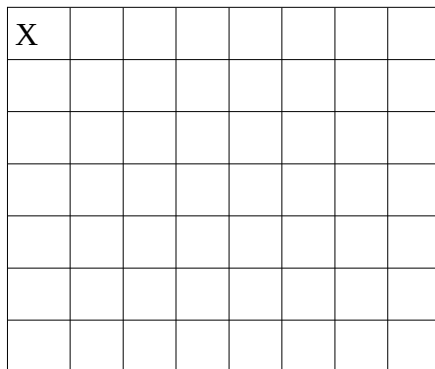


Dot-plot for x vs. rev-comp of y.



Make second versions of each of the above 4 plots in which you apply a window of size 5 and a stringency of 2.

x vs. y:



rev-comp of x vs. y

X							

rev-comp of x vs. rev-comp of y

X							

x vs. rev-comp of y.

X							

## 2. Sequence Alignment

Start with this “sequence #1”: **TGGTAGAGCG**

Make 4 mutations to this sequence #1: two insertions, one deletion, and one substitution anywhere in the sequence but spread out uniformly (do not cluster mutations in one part of the sequence) to come up with a related but different sequence (“sequence #2”).

Write down what your mutations were and how you got this second sequence.

“sequence #2”:

We know what the “optimal” alignment should be because we introduced the mutations. Write down what the alignment of sequence #1 and sequence #2 should be:

sequence #1: **TGGTAGAGCG**  
                  **??????????**  
sequence #2: **XXXXXXXXXX**

To these two sequences, apply:

- i. the “basic” dynamic programming global alignment algorithm (pages 9-12 of Lecture 6 Notes).

Format the array using a Table (like the dot-plots above) or the LibreOffice spreadsheet program, Calc (like Excel).

What is the “path” you end up with from the lower right corner to the upper left corner?

Path:

Remember that the output of this algorithm is an **alignment** we get from the path and that you should end up with an alignment.

- ii. the “basic” algorithm with **nonzero gap penalties** (pages 12-15).

Format the array using a Table.

What is the “path” you end up with from the lower right corner to the upper left corner?

Path: Remember that the output of this algorithm is an alignment and that you should end up with an alignment.

Which alignment is closest to what you expected?

Where did the dynamic programming methods go wrong?

### 3. Python program #1

Write a Python program that will read in a Fasta file (containing a single DNA sequence in either upper case or lower case or both) and print out the transcribed RNA sequences (uppercase is fine) of the original sequence and of the reverse complement of the original sequence.

Also print out the number of possible occurrences of codons for Methionines in the sequence and in its reverse complement. Explain in detail whether we have to consider frames and the reasons.

### 4. Python program #2

Write a Python program that will read a Fasta file containing a single protein sequence and print out the number of possible DNA sequences that could code for that protein sequence. You only have to print a number, not the sequences.

For each Python program in parts 3 and 4 above submit two FASTA files that you used to test the program

Submit the 7 files (1 document, 2 programs, 4 FASTA files) for this assignment.