Lab 2: Sequence Alignment Dynamic Programming Algorithm
Due: Monday, February 17th – at the start of class

The purpose of this lab is to learn and implement a dynamic programming algorithm for global sequence alignment using pseudocode that has already been provided.

For today, we will assume that all substitutions are equally likely, and assign a single value for a mismatch score. Your algorithm will take as inputs the mismatch score, match score and the gap score, and should output the optimal (maximum) global alignment score between two sequences as well as the alignment itself.

Download the starter code from Moodle (Lab 2 Files). Following the pseudocode provided, write the NeedlemanWunsch function, including filling in the hints table. The rest of the code is already provided.

What I need from you:

1. Turn in your completed sequence alignment algorithm code file. (upload to Moodle)
2. Answer the following questions.
    a. Run your code using TACGGGTAT as sequence 1 and GGACGTACG as sequence 2. Assume that the match premium is +1 and the mismatch and indel penalties are -1. What is the optimal global alignment of these 2 sequences and what score does it achieve?

    b. Consider finding the optimum global alignment of the sequences GATTACA and TAGACAT. Assume that a match has a score +2; a mismatch has a score -1; and an indel has a penalty of -2.

    Fill in the following scoring matrix by hand – check your work with your program.

	ε	G	A	T	T	A	C	A
ε								
T								
A								
G								
A								
C								
A								
T								

What is the optimal global alignment of these sequences?

c. Write down the recursive rule for the global sequence alignment algorithm. Remember that all recursive rules include a base case.

d. At this point, you should have a good understanding of how the Needleman-Wunsch algorithm constructs optimal, global alignments. A related problem is that of finding and aligning conserved regions in otherwise dissimilar sequences by looking for optimal partial or subsequence matches between the sequences. These are referred to as local alignments.
Consider the sequences AAAGCTCCGATCTCG and TAAAGCAATTTTTGGTTTTTTTCCGA. Two similar regions in these sequences, AAAGC and TCCGA, are separated by regions that are very different. A global alignment program should find the AAAGC alignment, but will fail to correctly align the sequences so that the TCCGA sequences also match up. (Try it!)

To find subregions of similarity, large gaps must be expected and should not adversely affect the alignment score. While in global alignments, negative values are useful since they indicate a move away from similarity, in local alignments, negative scores are no longer useful.

Think about a small change that you could make to your recursive rule in (c) to modify the rule to work for local alignment instead of global alignment. Write down your new recursive rule, or simply describe the change(s) you would make.