Agreement form for Bioinformatics take-home midterm

I understand that the take-home midterm in Bioinformatics is intended to test my own personal knowledge of and ability to apply concepts, methods, and factual information from the course. I therefore agree to do **ALL** of the following:

1. Complete all problems **<u>on my own</u>**, without discussing them with any other person (other than Drs. Weisstein and Beck)

2. Use **<u>only the following resources</u>:**
   - course PowerPoint slides and your own lecture notes
   - *Learning Python, 5th edition* (linked on the course Blackboard site)
   - any code and documentation that you or a member of your own team produced for an earlier assignment in this course
   - any other resource that has been approved *in writing* by Dr. Weisstein

3. **<u>Report any evidence</u>** that other students are violating or seeking to violate these terms.

I understand that failure to uphold these terms constitutes a serious violation of Truman's Academic Integrity policy and will be treated accordingly.

Signature: _______________________________________ Date: ____________

1a. Write and thoroughly document an efficient Python program that locates possible eukaryotic promoters. This program should carry out the following steps:

- Import a FASTA file containing up to 50 DNA sequences;
- Check to make sure each sequence contains only valid characters;
- Allow the user to manually input up to five promoter elements, each consisting of a *name*, *consensus sequence* (possibly using ambiguity codes: see next page), and *score*;
- For each sequence, find the **closest** match for each individual promoter element, assign a subscore representing the quality of that match, and add subscores to compute a total score for that sequence; and
- Report valid sequences in decreasing order of total score.

b. Use your program from part (a) to search the sequences in the file ***MidCS1.txt*** for the promoter elements listed at right. Print the resulting output. (Note: The input file contains several deliberate formatting errors that your program must handle appropriately.

Name	**Consensus sequence**	**Score**
Initiator	TCAKTY	64
TATA box	TATAWAAR	32
BREu	SSRCGCC	8

2. We have defined D as the proportion of sites that differ between two DNA sequences. Let us now define $v = V / D$ as the percentage of those differences that are ***transversions***. For example, consider two sequences in which 97% of sequences are identical, 2% differ by a transition, and 1% differ by a transversion. For this pair of sequences, $D = 0.03$ and $V = 0.01$, so $v = 0.01/0.03 = 0.33$.

a. Calculate and graph the Jukes-Cantor and Kimura 2-parameter distances as functions of v. Assume that the two sequences being compared differ at 20% of their positions ($D = 0.20$). Your graph should have an interval size **no greater than 0.01**; that is, it should plot d_{JC} and d_{K2P} for $v = 0.00$, $v = 0.01$, $v = 0.02$, and so on, up to the maximum possible value of v.

b. Explain why the Jukes-Cantor distance does not depend on v, but the Kimura 2-parameter distance does.

c. For what **exact** value of v does the Jukes-Cantor distance most closely approximate the more precise Kimura 2-parameter distance? **Clearly explain why.**

3a. Most eukaryotic genes contain some regions that are undergoing neutral evolution and other regions that are undergoing purifying selection. Explain why. Your answer should include predictions regarding which specific regions of the gene will undergo each evolutionary pattern.

b. Which of the regions above would be most appropriate to use for determining the age of a fairly recent speciation event (e.g., between two closely related lizard species)? **Clearly explain your reasoning.**

4a. Write and thoroughly document an efficient Python program that (i) imports up to four DNA sequences in FASTA format, (ii) globally aligns all four sequences using dynamic programming, and (iii) clearly outputs **each** optimum (i.e., highest-scoring) global alignment and its corresponding alignment score.
Tip: For simplicity, consider using sequential pairwise alignment. In other words, consider all possible pairs of the four initial sequences and separately align each pair. Then choose the pair of sequences that yields the highest-scoring global alignment, remove those two sequences, and add the alignment. You now have three sequences to align (two of the original sequences plus the partial alignment from the previous step). Repeat until all sequences are combined in a single alignment.

b. Use your program from part (a) to align the sequences contained in the file *MidCS4.txt*, and print the resulting output. Appropriate values for these closely related sequences might be as follows: match bonus +5, mismatch bonus –4, gap penalty –11.

Nucleotide ambiguity codes

These codes are used to indicate uncertainty or degeneracy in a nucleotide sequence. For example, a consensus sequence of **RCC** means that the first position could be either an **A** or a **G**. The table at right lists commonly used ambiguity codes.

Code	**Nucleotides**
A	A
C	C
G	G
T	T
R	A or G
Y	C or T
S	C or G
W	A or T
K	G or T
M	A or C
B	C, G, or T
D	A, G, or T
H	A, C, or T
V	A, C, or G
N	A, C, G, or T
–	gap