True/False and Multiple Choice

1. **True** / **False** The Needleman-Wunsch algorithm finds every maximally scoring local alignment between two sequences.

2. **True** / **False** In the BLOSUM62 substitution matrix, the match score \( s(X, X) \) of a rarely occurring amino acid is higher than that of a frequently occurring amino acid.

3. **True** / **False** When performing protein BLAST with window size \( w \), a matching target and query sequence must have at least one contiguous substring of length \( w \) in common.

4. **True** / **False** A feature function in a linear chain conditional random field can depend on any part of the label (hidden state) sequence.

5. **True** / **False** The \( k \)-means algorithm finds the global minimum sum of squared errors.

6. For the matrix below, indicate whether it is ultrametric, additive, both or neither.

   (a) Ultra-metric

   (b) Additive

   (c) Both

   (d) Neither

   \[
   \begin{array}{cccc}
   & a & b & c & d \\
   a & 0 & 2 & 8 & 8 \\
   b & 2 & 0 & 8 & 8 \\
   c & 8 & 8 & 0 & 4 \\
   d & 8 & 8 & 4 & 0 \\
   \end{array}
   \]

7. **True** / **False** The minimum number of substitutions for this tree is 5.

8. **True** / **False** The coalescence time inferred from a finite sample is typically smaller (more recent) than the true coalescence time.
Short Answer

9. Within our framework for sequence alignment, what algorithm (global or local) and costs (gap, match and mismatch) would you use to find:

(a) Minimum edit distance (i.e. the minimum number of insertions, deletions and substitutions needed to transform one string into the other).

(b) Length of the longest common substring.

(c) Hamming distance (i.e. number of positions at which two contiguous and equal length strings are different).

10. We are hashing two sequences with the comb 1101, where 1=match, 0=don’t care. Give an example of two DNA sequences of length 10 that have 7 matches in an ungapped alignment, but do not have any matches by our comb.

11. Explain one advantage and one disadvantage of posterior decoding compared to Viterbi decoding for HMMs.
12. In lecture, we developed an HMM that uses biases in the frequencies of dinucleotides to detect CpG islands. We would now like to design an HMM that uses biases in the frequencies of codons (trinucleotides) to distinguish between protein-coding and non-coding regions. For example, in protein-coding sequences, the codon AGG is very common, while the stop codon TAG is very rare. How many states would you include in this HMM, and why?

13. What objective function does the fuzzy $k$-means algorithm try to optimize?

14. What is the naive Bayes assumption for a given set of features?

15. Why does the maximum parsimony algorithm tend to underestimate the true number of substitutions on a phylogeny?

16. For a given allele, why is the observation of a long haplotype (high EHH score) not sufficient to infer selection?
17. Consider the graph below. Find the path where the sum of all the edges along the path is maximized. Indicate the path on the graph.

18. Consider one iteration of the EM algorithm for a motif of length 3. Below we have provided you with a set of sequences and a $Z$ matrix ($Z_{ij}$ gives the probability that position $j$ in sequence $i$ is the start of the motif). Compute the next $M$ matrix (a position weight matrix representing the motif at the next iteration). Assume that all pseudocounts are 0 and that the background nucleotide distribution is uniform.

<table>
<thead>
<tr>
<th>Position</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>A</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>A</td>
<td>G</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>G</td>
<td>C</td>
<td>A</td>
<td>T</td>
<td>G</td>
<td>G</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$Z$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

19. We have seen a number of algorithms that use the hamming distance $D(x, y)$ between two strings $x$ and $y$, which treats every mismatch as equally likely, resulting in a fixed penalty/cost. However, in DNA sequences, transitions ($A \leftrightarrow G$, $C \leftrightarrow T$) are more likely than transversions (any other mismatch). Design a mapping $f$ from sequences over $S = \{A, C, G, T\}$ to sequences over an augmented alphabet $S' = \{A, C, G, T, +, -\}$, so that the hamming distance $D(f(x), f(y))$ effectively penalizes transitions with a cost of 1 and transversions with a cost of 2. Show how your mapping would compare sequences GTC and AGC and the resulting score.
20. When genomes are initially sequenced, it is common for the sequencing process to leave gaps of "missing sequence" due to errors or incomplete information. Since these gaps represent limitations of our technology and not the underlying biology, we would like to treat them differently when constructing sequence alignments. You are given draft-quality DNA sequences that have the character "#" at certain positions to indicate an unknown amount of missing sequence. Design a pairwise global sequence alignment algorithm that does not penalize deletions following a "#", and uses a linear gap penalty otherwise. Assume you are given two sequences $X$ and $Y$, a score matrix $s(a,b)$ where $a$ and $b$ are nucleotides, and a gap cost $d$. You do not have to specify the traceback procedure.