# Homework 4: Trees and Models

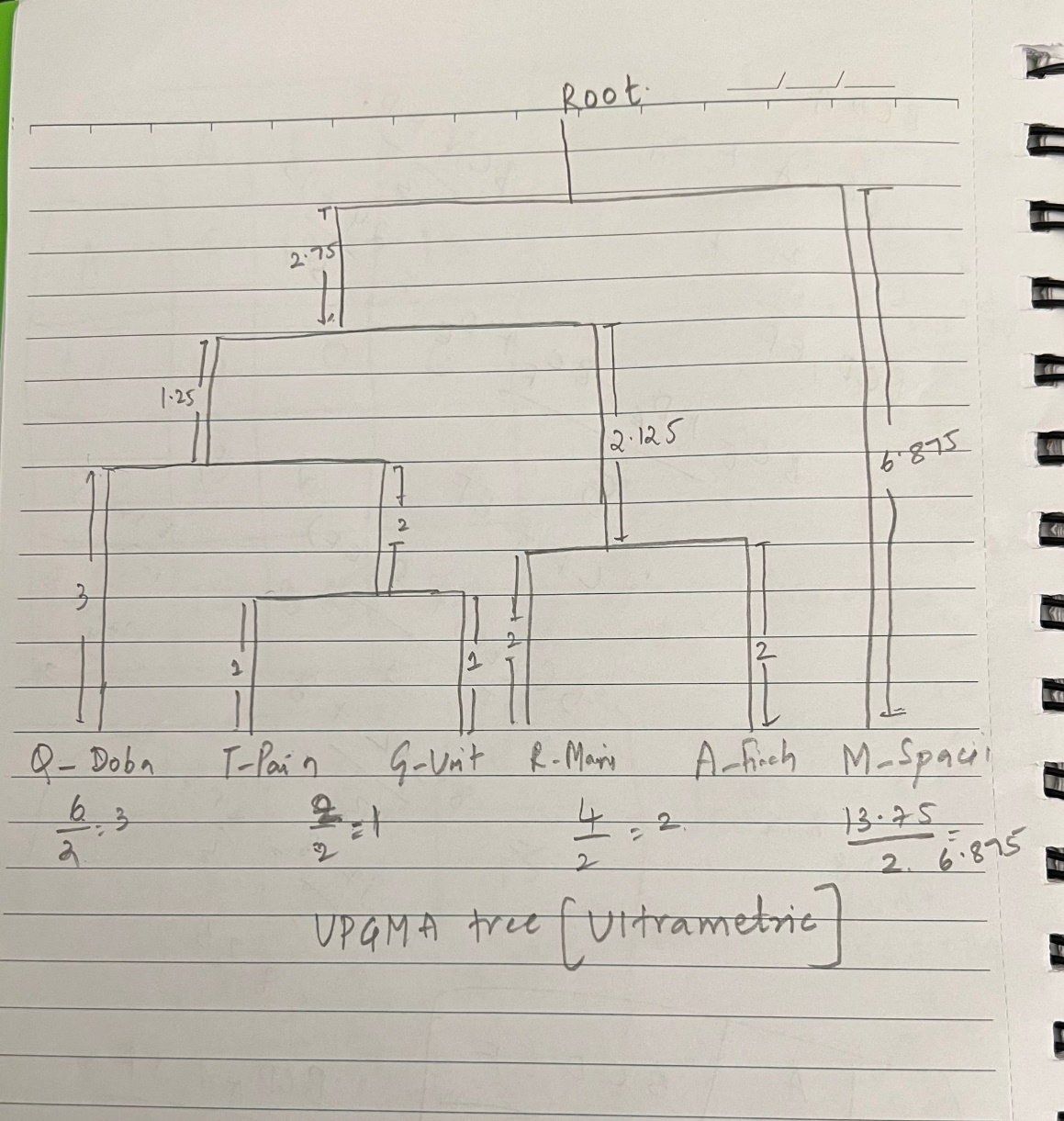
# (100 points total)

**Assignment guidelines**

1. Submit your assignment files on canvas under module 9: Trees and Models
2. Please submit your code in a file called [name].py. Your code should be easy to open in a text editor so that someone can download and use the function you write.
3. Please submit a pdf with the answers to the questions at the bottom of the assignment (and your visualization)
4. Please submit a fasta file (text file) with the output of your code called [name].fa

**Complete the class assignment: UPGMA (50 points)**

1. Complete the functions find smallest and update matrix (30 points)
   1. Code meets specifications – the function exists and makes correct input and output
      1. Code includes findSmallest() function which takes in a distance matrix and identifies the row and column of the smallest entry – chooses randomly if multiples ones are equal. (5 points)
      2. Code includes updateMatrix() function which takes in a distance matrix row and column and removes those species from the matrix replacing them with the average distances to all the remaining points (15 points)
   2. Correct tree is returned (10 points)
2. Edit the program so that distances are returned (20 points)
   1. Output is correct with distances being added to the species tree in bracket notation like: ((species\_1,species\_2:distance), species\_3:distance) (10 points)
   2. Draw the correct tree. Remember that it should be ultrametric! (10 points)



**Complete the class assignment: Models (50 points)**

1. We have two models (Coding and Noncoding) that represent the likelihood of a codon changing from a given codon to any different codon including itself. They represent the probabilities at time **t** where t is the distance of divergence between the ancestor and M.Spacii. Complete the functions to calculate the log likelihood of the coding model describing the sequences (20 points).
   1. Code meets specifications:
      1. scoreModels function is completed (5 points)
      2. Program outputs a list of sequences by ID with an associated label “likely coding” or “likely noncoding” (5 points)
      3. Resulting classification is correct (10 points)
2. Evaluate your results (20 points)

For this section use the data file Spacii\_2100.fa – a more divergent set of sequences.

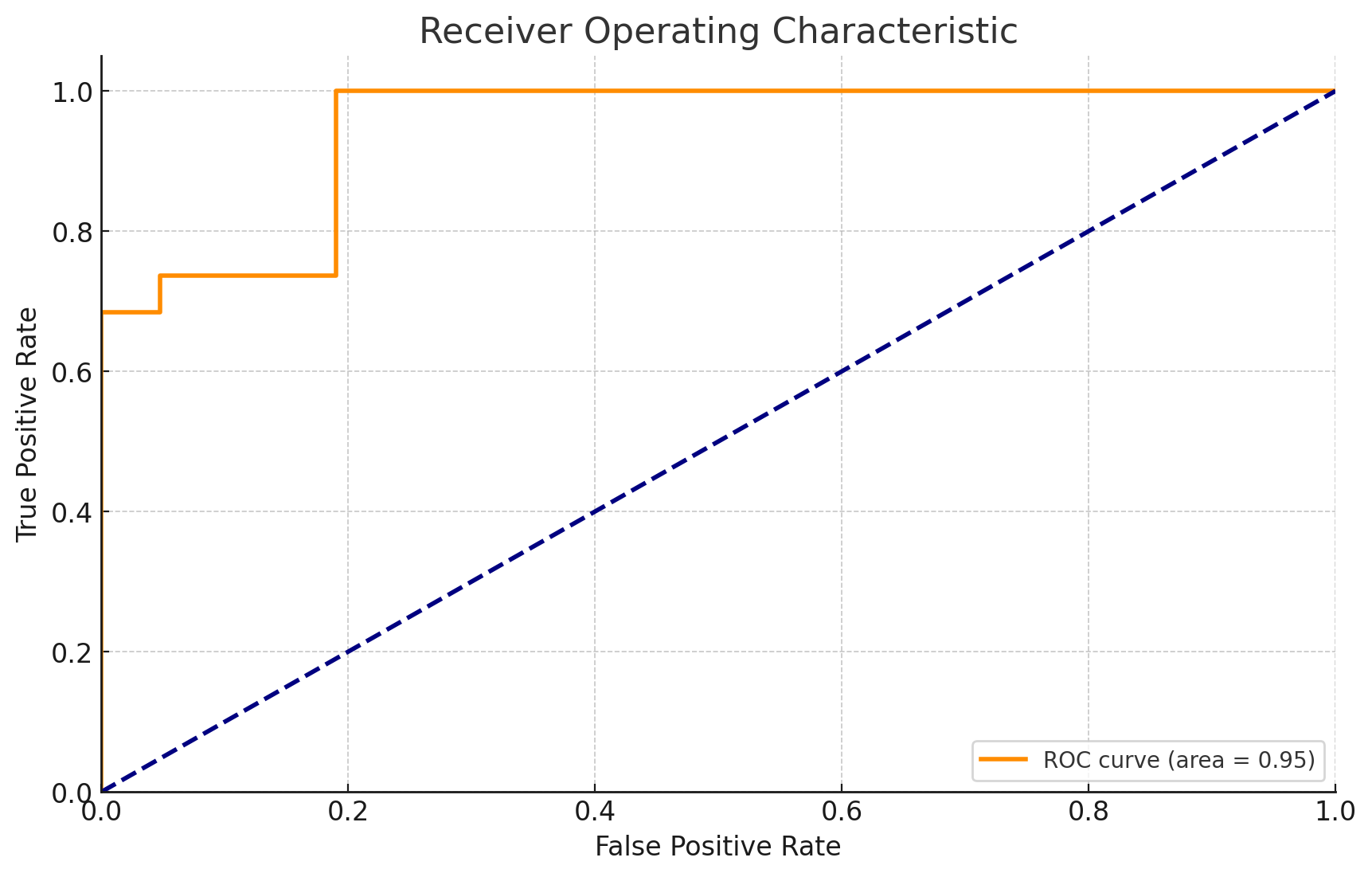
* 1. Consider that sequences marked with \_n\_ are truly noncoding and \_c\_ are truly coding. Conceptually, explain why some are misclassified (10 points)

**Answer:**

Misclassifications in our sequence analysis can be attributed to several factors based on the analysis. First, the models used might not fully encapsulate the intricate patterns that distinguish coding from noncoding sequences, leading to inaccuracies. This limitation is often compounded by evolutionary variability; over time, noncoding sequences can evolve to resemble coding sequences, thereby confusing the model. Additionally, there's an inherent overlap in the motifs or patterns present in both coding and noncoding sequences, which further challenges accurate classification. The training data for these models, which might not represent the full diversity of realworld sequences, also plays a crucial role in these misclassifications. Lastly, the inherent randomness in biological sequences adds another layer of complexity, resulting in unexpected similarities between coding and noncoding sequences, and thus leading to potential misclassification.

* 1. Create an ROC curve with the results using the likelihood ratio as a threshold (what is the true and false positive rate for various threshold values). You are not required to code this. Based on the ROC curve, what cutoff would you use? Justify your answer – be sure to explain whether you might care more about specificity or sensitivity! (10 points)

**Answer:**



**True and false positive threshold values**

**Threshold = 71.79:**

TPR: 0.0 (No true coding sequences are identified)

FPR: 0.0 (No noncoding sequences are misclassified as coding)

**Threshold = 95.81:**

TPR: 0.684 (68.4% of true coding sequences are correctly identified)

FPR: 0.0 (No noncoding sequences are misclassified as coding)

**Threshold = 101.38:**

TPR: 0.737 (73.7% of true coding sequences are correctly identified)

FPR: 0.048 (4.8% of noncoding sequences are misclassified as coding)

**Threshold = 108.89:**

TPR: 1.0 (100% of true coding sequences are correctly identified)

FPR: 0.190 (19.0% of noncoding sequences are misclassified as coding)

**Threshold = 146.93:**

TPR: 1.0 (100% of true coding sequences are correctly identified)

FPR: 1.0 (100% of noncoding sequences are misclassified as coding)

These values demonstrate the tradeoff between sensitivity and specificity at different thresholds. Lower thresholds increase sensitivity (TPR) but can also increase the rate at which noncoding sequences are wrongly identified as coding (FPR). Conversely, higher thresholds improve specificity (lower FPR) but reduce sensitivity (lower TPR). The optimal threshold would balance these two rates depending on the specific requirements of the analysis.

**Cutoff :**

Based on the ROC curve analysis, a balanced threshold cutoff would be between 101.38 and 108.89. This range offers a good tradeoff, with a True Positive Rate of about 73.7% to 100%, indicating effective sensitivity. The False Positive Rate ranges from approximately 4.8% to 19%, maintaining reasonable specificity. This threshold selection captures most true coding sequences while limiting the misclassification of noncoding sequences, suitable for phylogenetic analysis where accuracy is critical.

**Justification:**

I would prioritize specificity slightly more than sensitivity. High specificity minimizes false positives, crucial for avoiding incorrect assumptions about gene functions and evolutionary history. While sensitivity is important, a slightly lower True Positive Rate is acceptable to significantly reduce false positives. The chosen threshold (101.38 to 108.89) balances this, still identifying the most true coding sequences.

1. Consider the models at time **t‘** > **t** : (10 points)
   1. Describe what the coding matrix will look like at time 2**t**. Show how you would calculate probability of TTT staying TTT after time 2**t**. No need to write out the full equation – you may use (…) – just show me enough to make it clear you follow the calculation by listing a few terms. (5 points)

Answer:

The coding matrix at time ( 2t ) will show compounded probabilities of codon transitions from time ( t ). Each entry is the sum of the probabilities of all possible paths from one codon to another over the time ( 2t ).

For 'TTT' to stay 'TTT', we calculate:

We assume the initial probability of TTT staying TTT at time t is 0.7

There are 63 possible codons that TTT could mutate to

The probability of TTT changing to any one of the 63 codons is (1 0.7)/63

To calculate the probability at time 2t, we consider two cases:

TTT stays TTT at both times t and t, which is 0.7 0.7

1. Probability of 'T' Changing to Another Nucleotide at Time (t) ((s(t))):

s(t) = (1 -0.7)/63 i.e approx 0.00476

2. Probability of 'T' Staying 'T' After Time (2t) ((r(2t))):

( r(2t) ), the first element of the matrix ( S(2t) ), is approximately (0.4901).

3. Probability of "TTT" Staying "TTT" After Time (2t):

The probability is ( r(2t)^3 approx 0.1177.

b) Make a prediction about your classification success if the sequences had longer to diverge. Explain how you would expect the ROC curve to change as **t** increases. (5 points)

Answer:

It is anticipated that the current model's classification success will decrease as the sequences' divergence times, ( t ), grow. This is due to the fact that longer divergence promotes more mutations, which increase sequence differences. This would lead to an increase in misclassifications as the model's capacity to discriminate between coding and noncoding sequences would probably decline.

As a result, the ROC curve would move in the direction of the diagonal line, which stands for random chance, suggesting that the model is less successful. A drop in the area under the curve (AUC), which indicates a reduced overall classification accuracy, would be indicative of this shift. Furthermore, the ideal ROC curve threshold for striking a balance between specificity and sensitivity may shift.