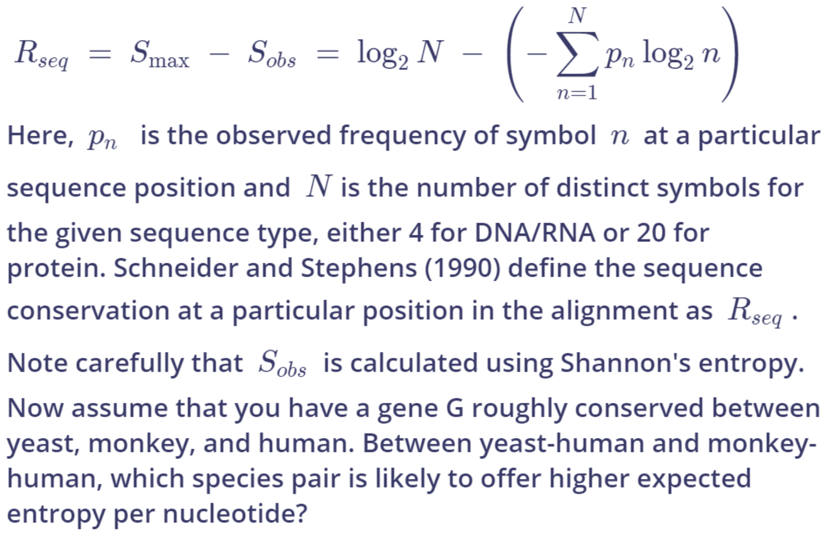
1. You have *N* DNA sequences of length 10000. A motif (identical across the DNA sequences) of length *L* is hidden in all *N* sequences. You need to find the locations of the motif in the sequences that yields the best score. How many combinations of *L*-mers will you need to try to find the motif? Explain.

[8 points]

2.



$$R_{seq} = S_{\max} - S_{obs} = \log_2 N - \left(-\sum_{n=1}^{N} p_n \log_2 n\right)$$

Here, p_n is the observed frequency of symbol n at a particular sequence position and N is the number of distinct symbols for the given sequence type, either 4 for DNA/RNA or 20 for protein. Schneider and Stephens (1990) define the sequence conservation at a particular position in the alignment as R_{seq}.

Note carefully that S_{obs} is calculated using Shannon's entropy.

Now assume that you have a gene G roughly conserved between yeast, monkey, and human. Between yeast-human and monkey-human, which species pair is likely to offer higher expected entropy per nucleotide?

Provide details explanation.

[8 points]

3. Arrange the following steps in order to construct a meaningful probabilistic algorithm for motif discovery.

L. Sample a new site proportional to likelihood and update motif instances

W. Build a weight matrix

U. Update weight matrix

C. Iterate until convergence

P. Select a random position in each sequence

S. Score possible sites in the sequence using weight matrix

R. Select a sequence at random

4. Answer briefly

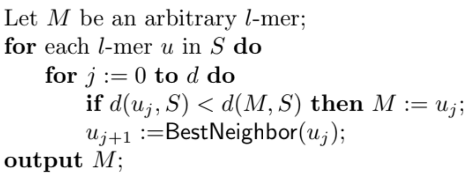
a. What are some of the known functions of non-coding region of the DNA?
b. What is sequencing by synthesis?
c. What are the roles of promoters and enhancers?
d. State the preference for long/short reads for the below usecases

i. I am studying transcriptomic changes in cancer as compared to normal tissue.

ii. I am performing de novo genome assembly.

[8 points]

5. Below is the pseudocode for Patter-Branching algorithm. Comment on its time complexity.



```
Let $M$ be an arbitrary $l$-mer;
for each $l$-mer $u$ in $S$ do
    for $j := 0$ to $d$ do
        if $d(u_j, S) < d(M, S)$ then $M := u_j$;
        $u_{j+1} :=$ BestNeighbor($u_j$);
output $M$;
```

[8 points]