**Question 1: List the standard alphabets defined in sym.py (e.g. Bool\_Alphabet). Please ensure your spelling is correct and your answers are separated by a comma (,).**

Bool\_Alphabet, DNA, RNA, DNAwN, RNAwN, Protein, ProteinwX, ProteinwSTOP, ProteinwGAP, DSSP\_Alphabet, DSSP3\_Alphabet

**Question 2A: List the 'special' functions from the Sequence class. Just enter the function name, for example, for \_\_len\_\_(self) just enter \_\_len\_\_**

\_\_init\_\_, \_\_len\_\_, \_\_str\_\_, \_\_iter\_\_, \_\_contains\_\_, \_\_getitem\_\_

**Question 2B: Provide an example (in text, not code) of the use of each function from Question 2A**

\_\_init\_\_: Create a sequence with the sequence data. Specifying the alphabet, name and other information about the sequence are all optional. The sequence data is immutable (stored as a string).

**Question 3A: How many different types of identifiers are associated with the sequences in mystery2.fa? To answer this, submit the first two letters common to the identifiers.**

NP, XP

**Question 3B: Which databases do the identifiers from Question 3A map to?**

Ensembl, uniport

Many of you did not provide the name of the dataset as asked. Instead, some of you provided the name of species from which the sequences are derived. For example, the most widely used protein sequence databases are **NCBI, Uniprot and PDB**. Normally each database has its unique prefix in its sequence headers (e.g. NP, XP), thus you can infer the name of the database from the header.

**Question 4A: How many entries are in sigpep\_at.fa?**

1852

**Question 4B: How many entries are in lipmet\_at.fa?**

153

**Question 5: How many TAG lipases did you find?**

7

**Question 6A: Describe the physico-chemical properties represented by each default colour used in the alignment (including the white/uncoloured amino acids).**

green: Nonpolar(G), Nonpolar(C), Nonpolar(P), Nonpolar(F), Nonpolar(W), Nonpolar(I), Nonpolar(L), Nonpolar(M), Nonpolar(V), Nonpolar(A)

#66bbff: Polar(S), Polar(T), Polar(Y), Polar(N), Polar(Q)

red: Basic polar(H), Basic polar(K), Basic polar(R)

orange: Acidic polar(D), Acidic polar(E)

**Question 6B: Show your own 'hydrophobic' colour scheme (as a list of affected amino acids)**

blue(hydrophobic): V, I, L, F, W, Y, M

otherwise white

**Question 6D: Provide the rough boundaries of the fifth transmembrane domain. Enter your boundaries in the following format: 10 – 30**

290-310

The “gpcr.aln” file contains an alignment of 23 GPCRs (protein sequences), which means these protein sequences should be like some extent. First, amino acids with the same physicochemical properties can play the same structural role for a protein, thus they can substitute one another more frequently without affecting the structural integrity of the protein. Therefore, if you have amino acids colour-coded by their physico-chemical properties in alignments, a position within the same colour can be interpreted as aligned position. Second, the conserved sequence regions are critical for their 3D structures and functions; these regions are therefore important. If you look at the alignment, you will find blocks of aligned positions; these commonly represent domains.

The “transmembrane domain” represents an aligned block that consist of almost all hydrophobic amino acids and in a range of 10-30 bp. You would expect 7 such blocks in the alignment.

To address this question, you would (a) first identify hydrophobic versus hydrophilic amino acids, (b) make the colour scheme to only consider hydrophobicity, (c) browse through the alignment and find the fifth hydrophobic block, and (d) note down the range of this block. For example, you may find the first block at ~100-130 bp, and the second at ~140-160 bp.

Some of you also did not use the readClustalFile function properly. This function requires two mandatory parameters, the “. aln” file and an alphabet. Don’t forget the alphabet. Here is an example of correct usage:

aln = readClustalFile('p450.aln', Protein\_Alphabet)

**PRAC2**

Calculate the probability that two amino acids a and b will appear aligned purely by random chance, eab.

* The chance of seeing two of aa aligned, assuming all differences in the sequences were randomly generated, is eaa=pa2.
* The chance of seeing a and b aligned is eab=2papb

The 2 is because there are two ways of selecting two different items by chance (ab or ba). The matrix e is our random model.

If we now compare these two probabilities, from the evolutionary model and the random model, we can see whether it is more likely to see two specific amino acids aligned by chance or by an evolutionary process.

If we set the score to be qab/eab, this will give us what we want. Greater than 1 and the alignment is likely from evolution, between 0 and 1 and it is likely by random chance. If we take the log of this score, then we end up with a negative score for chance and a positive score for the evolutionary model. This is called the log-odds score, sab = log(qab/eab), and this is what we will be calculating

**Question 1A: Provide the answer to your calculation of eaa**

0.0196

**Question 1B: Provide the answer to your calculation of eab**

0.0616

**Question 1C: Provide the answer to your calculation of sab**

-0.60200451209

**Question 1D: Provide an explanation for how the calculation of substitution scores works**

**Question 2A: Enter your Python code for calculating seqPairs. This should be submitted to Coder Quiz in the format seqPairs = MY\_ANSWER**

seqPairs = math.factorial(numSeqs) / (2 \* math.factorial(numSeqs - 2))

**Question 2B: Enter your Python code for calculating aaPairs. This should be submitted to Coder Quiz in the format aaPairs = MY\_ANSWER**

aaPairs = columns \* seqPairs

**Question 2C: Enter your Python code for calculating eab where a == b . This should be submitted to Coder Quiz in the format eab = MY\_ANSWER**

eab = p[a] \* p[a]

**Question 2D: Enter your Python code for calculating eab where a!= b. This should be submitted to Coder Quiz in the format eab = MY\_ANSWER**

eab = 2 \* p[a] \* p[b]

**Question 3B: Submit the alignment if the gap penalty is set to -8 . To differentiate the two rows of the alignment separate them by a comma e.g. MADMAN,MAD-AM.**

THISLINE-, ISALIGNED

**Question 3C: Submit the alignment if the gap penalty is set to -4 . To differentiate the two rows of the alignment separate them by a comma e.g. MADMAN,MAD-AM.**

THIS-LI-NE-, --ISALIGNED

**Question 4A: How many cells would the S matrix (in the alignGlobal function) contain when aligning HQ659871.1 and JX416721.1?**

2345868

**Question 4B: If you leave the DNA substitution matrix untouched, what is a biologically sensible gap penalty?**

-4

**Question 4C: What steps did you take to determine 4B?**

if the gap penalty is very negative, more so than any score in the substitution matrix, then it will never be good score-wise to place gaps. As a result, the sequences will be lined up with no gaps, regardless of how bad the result is. Conversely, a too-small gap penalty may lead to an alignment where there is little, or no identity overlap between the sequences.

**Question 4D: Given the original DNA substitution matrix and a gap penalty of -5, at what position is the first ATG codon in reading frame +2 of seqB (JX416721.1)?**

92