

# Assignment 1

|  |  |  |
| --- | --- | --- |
| Date:  Course:  Name:  Student number: | 05/04/2025  MMG1344H  Maria Eleni Fafouti  1010799094 | |
|  | |  |
|  | |  |
|  | |  |

# Part 1

Section 1:

To calculate the probability of a 150 a.a protein matching the following pattern:

, we must consider the individual probabilities of matching each position. Based on the table given, they are calculated as follows:

|  |  |
| --- | --- |
| Position | Frequency |
| P(R) | 4% |
| P(x) | 100% |
| P(S) | 9.3% |
| P(T) | 5.0% |
| P([ST]) | P(S) + P(T) = 14.3% |

The probability for a single hexamer matching the query sequence would be:

However, our protein of interest is 150 amino acids long, which means that there are multiple hexamers within it. If we take a smaller, 10-residue protein as an example and look for 4-mers we will find that there are 7 possibilities (see image below).

A close-up of a math test

AI-generated content may be incorrect.

Figure 1. Example showing how to determine the number of k-mers in an amino acid sequence of a given length. In this case an example sequence of 10 amino acids is queried for 4-mers.

This means that in general, we can use this formula to identify the number of k-mers in a sequence of a specified length:

In our case, this number is 145.

We need to calculate the probability of at least one of the 145 6-mers to match the query sequence. This would be the probability of one 6-mer matching the query, multiplied by the total number of 6-mers in the protein sequence. The result is 0.177906.

Section 2:

To calculate the posterior probability that Yfp1 is phosphorylated by Cmk, we will utilize Bayes’ theorem. We know that in Ptacek et al, 4000 proteins were screened and 87 of them had patterns that matched kinases. Out of the 87, only 9 matched Camk2.

To calculate the posterior probability that Yfp1 is phosphorylated by Cmk2, given that it matches the R-x(2)-[ST]-x-[ST] pattern, we apply the following:

The following table explains the terms of the equation and assigns them values based on the Ptacek et al study:

|  |  |  |
| --- | --- | --- |
| Term | Definition | Value |
|  | the probability we wish to find |  |
|  | the probability that Yfp1 matches the R-x(2)-[ST]-x-[ST] pattern (previous section) |  |
|  | the probability that Yfp1 matches the R-x(2)-[ST]-x-[ST] pattern, given that it is phosphorylated by Cmk2 |  |
|  | the probability that a protein is phosphorylated by Cmk2 |  |

Note: the probability that Yfp1 matches the R-x(2)-[ST]-x-[ST] pattern, given that it is phosphorylated by Cmk2 is 1, because matching the pattern is a prerequisite for Camk2 phosphorylation.

After performing the calculation as explained above, the probability that Yfp1 is phosphorylated by Cmk2, given that it matches the R-x(2)-[ST]-x-[ST] pattern is 0.0126.

# Part 2

Section 1

To count the fraction of proteins which match the pattern R-x(2)-[ST]-x-[ST] in the FASTA file provided, we will first import the fasta file and save it as a pandas dataframe. The dataframe contains 2 columns: ID and sequence (more details in the output.txt file). This means that we have to iterate over all the sequences with our pattern search.

After defining the pattern in the regex format, a function is defined to search for it in each sequence in the .fasta file. Each sequence in the .fasta file, likely corresponds to a protein, so we will have to count how many of these sequences match this pattern at least once.

Then, we will count all the proteins that match the pattern at least once and divide that number with the total number of proteins to get the frequency as our output.

Section 2

To compare the frequency of the amino acids in the yeast genome versus the *S. cerevisiae* as provided in the assignment instructions, we first must define the 20 amino acids.

After that, I initialized a counter which will be used to count amino acids in different instances. First, I used the counter to find how many appearances of each amino acid are there in each protein sequence. Then, I summed the appearances of each amino acid across sequences to get the total times it appears in the proteome. After that, I iterate over the string of amino acids to compute the frequency for each one and saving it all in a dictionary, which is later converted into a vector, as required.

Section 3

To compare the two amino acid frequency outcomes (answer form Step 2 vs. table given in part 1), we have to integrate both into 1 pandas dataframe. First, I will import the table from the pdf file of the assignment instructions. To do this, I used the package “pdfplumber”.

When using this package, I noticed that it was easier to extract the text first, then split it in terms of lines and spaces and then save it as a pandas dataframe. I had to remove the “%” sign next to each value for a clearer readout.

I proceeded to save the vector that was computed in section 2 (with the S288C yeast strain frequencies). This one needed two modifications to be comparable with the table provided in the assignment instructions: I renamed the amino acids: switched from the 1-letter naming convention to the 3-letter one and multiplied the frequency values by 100 such that the two tables are comparable.

After merging the two tables and computing the difference between the 2 columns of amino acid frequencies, I computed the difference between them in a new column: “Difference”, which helped me think of the possible reasons for why they might be different. The paragraph explaining is found in the code output.txt file.

Section 4

To compute the marginal probabilities, the following calculation will take place:

A frequency matrix will be created from the vector of 20 frequencies calculated in Step 2. After converting the frequency vector from step 2 into a dictionary, the amino acid codes could be distinguished from their frequencies. The, the probability calculation above was computed for each row and column combination, and hence for all cells in the matrix.

I chose to color this matrix based on the probability value and save this as a figure in the current working directory.

Section 5

This section is similar to the previous one, but we are asked to calculate the di-amino acid frequencies empirically. This means that we will not use the amino acid frequencies, but count how often the amino acid pairs actually occur in the proteome.

To do this, we read into the .fasta file and create a sliding window of 2 to count the amino acid pairs. The unique amino acid pairs are saved in a matrix and every time a specific pair is seen, the count is increased by one. Each time a pair of any kind is detected, the count of total pairs is increased. The value of total pairs will be used to divide the matrix of counts and compute the empirical frequencies.

This is how the two matrices from section 4 and 5 look like, side-by-side:

A screenshot of a graph

AI-generated content may be incorrect.

Section 6

To calculate the conditional amino acid frequencies given that a certain amino acid is preceded by Isoleucine (I) or Glutamine (Q), I will use the following calculation:

This means that the conditional probability of a certain amino acid (x) exisiting in the sequence given that it is preceded by y (in our case it is either isoleucine or glutamine), is computed by the probability of the amino acid of interest (x) being preceded by y dividing by the total occurrences of y.

I will define this within an equation and apply if for both cases of Isoleucine or Glutamine being the first amino acid and an amino acid pair. This means that 2 probabilities will be computed for each of the 20 amino acids. I organized the output in a pandas dataframe.

Section 7

To compare the empirical amino acid frequencies from section 6 to the empirical amino acid frequencies determined in section 2, I placed both in the same pandas dataframe and computed their difference. This means that I ended up with 40 values, 20 with the difference of Isoleucine being the first amino acid, compared to the marginal amino acid frequencies, and 20 with the difference of Glutamine being the first amino acid compared to the marginal amino acid frequencies.

After that, I decided to visualize the differences I computed earlier. I used a colored scale where white would correspond to zero, blue corresponds to negative values and red to positive values. This is how the differences look like:

A graph of different colored lines

AI-generated content may be incorrect.

In the visualization above, we can clearly see the differences between the conditional frequencies from section 6 and the marginal amino acid frequencies in the yeast genome. In general, the two frequency categories show very little divergence from each other; any differences – if any – lie in the range of 0-0.03 units of absolute difference in frequency. Di-amino acid pairs where the first amino acid is Isoleucine, present very small divergence from the marginal amino acid frequencies. Only the Isoleucine-Proline (I-P) However, certain di-amino acid pairs where the first amino acid is Glutamine, show some divergence from the marginal frequencies. For example, the Glutamine-Serine (Q-S), displays a lower frequency compared to Serine alone. The Glutamine-Glutamine (Q-Q) pair on the other hand, displays a higher frequency than the marginal frequency of glutamine alone. Several other pairs (such as Q-E, Q-L, Q-Y) show a similar behavior.

These differences have several biological explanations. To begin with, Glutamine contains a side chain that enables hydrogen bonding (Rhys et al., 2012), highlighting its role in protein structure. Hydrogen bonds are important to maintain secondary protein structure (formation of alpha helices and beta pleated sheets) which is vital for proteins to function properly. A potential explanation for the increased frequency of Q-Q pairs is that this leads to more stable Hydrogen bonds (as two adjacent sites are available on the protein), leading to more stable protein structure overall. Further, Q-rich regions are usually present in transcription factors and have roles in protein-protein interactions (Martinez-Yamout et al. 2023).