MMG1344H: Foundational Computational Biology (FCB) Course I **Spring 2025**

Assignment 1

Part 1 "A phosphosite? Are you sure?"

Phosphorylation is a post-translational protein modification that can serve to switch the activity of the target protein on and off. There have been systematic efforts to reveal target proteins for the enzymes (kinases) that carry out phosphorylation. One strategy is to express thousands of proteins and array them on the surface of a 'chip' (Ptacek *et al.*, 2005). The protein array can be exposed to a particular kinase of interest in the presence of radiolabeled (gamma-^{33}P) ATP. Target proteins can be identified by virtue of becoming radiolabeled by the kinase. Ptacek *et al.* arrayed 4000 *S. cerevisiae* proteins and identified substrates for 87 protein kinases. This study revealed, for example, 9 target proteins of the S. cerevisiae kinase Cmk2.

Kinases have protein sequence preferences. For example, Cmk2 reportedly phosphorylates proteins matching the pattern R-x(2)-[ST]-x-[ST] (Ptacek *et al*., 2005). (In English, that pattern can be read "Arginine, anything, anything, Serine or Threonine, anything, Serine or Threonine".) Out of the 9 proteins identified as Cmk2 targets by Ptacek *et al.*, 7 contained a single site matching the R-x(2)-[ST]-x-[ST] pattern, and 2 had no matches to that pattern. Amongst all 4000 proteins on the array, 1325 matched the pattern.

1. Imagine you're looking at the sequence of your favourite protein (aptly named Yfp1), which was not measured on the Ptacek *et al.* array. You see that Yfp1 does have a match to the R-x(2)-[ST]-x-[ST]. The Yfp1 sequence is only 150 amino acids long, which is shorter than the median *S. cerevisiae* protein which is ~375 aa long (Zhang, 2000). Using the table of amino acid frequencies provided below, calculate the probability that a random sequence of length 150 would match the pattern Rx(2)-[ST]-x-[ST] at least once just by chance. For the sake of simplicity, you can assume independence between positions and potential sites within the protein.
2. Assuming that the Ptacek *et al.* results are completely accurate and that the proteins measured by Ptacek *et al.* are representative of the other proteins not present on the array, what is the posterior probability that Yfp1 is phosphorylated by Cmk2, given that it matches the R-x(2)-[ST]-x-[ST] pattern

Observed Frequency of Amino Acids in *S. cerevisiae* according to (Xia, 2018)

Amino Acid	Frequency
Ala	5.9%
Arg	4.0%
Asn	6.2%
Asp	5.9%
Cys	1.3%
Gln	4.0%
Glu	6.6%
Gly	5.0%
His	2.2%
Ile	6.5%
Leu	9.5%
Lys	7.3%
Met	2.1%
Phe	4.1%
Pro	4.4%
Ser	9.3%
Thr	5.0%
Trp	1.4%
Tyr	3.4%
Val	5.9%

Part 2 "At the proteome scale"

Now that you have completed Part 1, it is time to dive deeper into the yeast proteome. In this part, you will examine the proteome of the yeast strain S288C, a widely used laboratory strain used to generate the reference yeast proteome in the Saccharomyces Genome Database (SGD).

We'll use the FASTA file ("orf_trans.fasta"), which contains the reference sequences of the S288C proteome, in the assignment folder:

1. Let's empirically count the fraction of proteins which match the pattern R-x(2)-[ST]-x-[ST], the same kinase binding motif investigated in Part I.

Tip: There are multiple methods for pattern matching; and you are welcome to choose the method with which you're most comfortable. For example, you can take a brute-force approach by generating all possible amino acid fragments that satisfy the pattern and scan the proteome with each one of the fragments. Alternatively, you might convert the pattern into a regular expression which can be used to quickly scan the proteome. Should you choose the regular expression approach, you might find this resource (https://regex101.com/) helpful to develop and debug your regular expression.

Output: A frequency.

2. Now let's double-check the observed amino acid frequency table we used in Part 1, by empirically measuring the marginal frequency of each of the 20 amino acids in the yeast proteome.

Output: A vector with 20 frequencies.

3. Compare the empirically measured amino acid frequency from Step 2 with the observed amino acid frequency in Part 1. Do they agree with each other? If they disagree, think of one possible source of difference with the previous table.

Tip: When answering the questions, please make sure to explain your rationale.

Output: A paragraph describing the comparison and answering the questions.

4. Using your marginal amino acid frequencies from Step 2, calculate the expected frequencies of di-amino acid 'words', e.g., Alanine followed by Glycine forms the di-amino acid word AG. Here you should assume independence between amino acids.

Output: A 20-by-20 frequency matrix. Column names should represent the first amino acid of di-amino acid words, and row names represent the second amino acid. For example, the frequency of di-amino acid word AG should be stored in column "A" and row "G"

5. Empirically measure di-amino acid frequencies.

Output: A 20-by-20 frequency matrix. Column names should represent the first amino acid of di-amino acid words, and row names represent the second amino acid. For example, the frequency of di-amino acid word AG should be stored in column "A" and row "G"

6. Using your empirically measured di-amino acid frequencies, calculate conditional amino acid frequencies where, for each vector, the conditioning is on a different one of the 20 possible preceding amino acids. Specifically,
    a. If the preceding amino acid is Isoleucine (I), what is the frequency of seeing each of the 20 amino acids in the next position?
    b. If the preceding amino acid is Glutamine (Q), what is the frequency of seeing each of the 20 amino acids in the next position?

*Output: 2 * 20 frequencies each labelled with the corresponding amino acid*

7. Compare the conditional amino acid frequencies from Step 6 to the empirical marginal amino acid frequency from Step 2. Do they agree with each other? How did you determine that? Please offer a potential explanation for the largest differences you see between marginal and conditional frequencies.

Tip: When answering the questions, please make sure to explain your rationale.
Output: A paragraph describing the comparison and answering the questions.

Notes

1. You can use any libraries to parse the FASTA file and to perform pattern matching.
2. You are NOT allowed to use functions, libraries or packages that generate single- or di-amino acid frequencies directly from protein sequences. This provides you an opportunity to understand how these frequencies are generated, which is the purpose of this part of the assignment.

Submission

Please submit one zip file (first_lastname.zip) with:

1. Source code (in .R or .py format) and a README file on how to run your program.
2. Your program's output (in .txt format).
3. A document (in .doc or .pdf format) with answers to questions in Part I and II.

References

Ptacek, J., Devgan, G., Michaud, G., Zhu, H., Zhu, X., Fasolo, J., Guo, H., Jona, G., Breitkreutz, A., Sopko, R., et al. (2005). Global analysis of protein phosphorylation in yeast. Nature *438*, 679–684.

Xia, X. (2018). String Mathematics, BLAST, and FASTA. In Bioinformatics and the Cell: Modern Computational Approaches in Genomics, Proteomics and Transcriptomics, X. Xia, ed. (Cham: Springer International Publishing), pp. 1–31.

Zhang, J. (2000). Protein-length distributions for the three domains of life. Trends Genet. *16*, 107–109.