Scientific Programming and Statistical Computing (SCM7047)
Assignment 2

Instructions:

You are reminded that this assignment is to be completed individually - do not share your solutions with others. Students found to be colluding on this assignment may be asked to appear before a School Academic Offences Panel.

Note on the use of AI tools: You are encouraged to use AI tools responsibly and with integrity during the completion of this assignment. For example, you could use AI to explain to you the underpinning concepts of the functions in Python that you need to complete each of the tasks. AI should not be used to do the assignment for you - this constitutes "contract cheating" and is an academic offence. If it is suspected that you have used AI to complete the bulk of this assignment you may have to appear before a School Academic Offences Panel. **Uploading of the data files provided for this assignment to AI tools is not permitted.** You are required to declare each instance of how you have used AI, if at all, in each section of the assignment and on the accompanying assessment cover sheet.

1. Assignment and associated datasets are available as downloadable links on this page.
2. Read each question carefully for a clear understanding of what it demands.
3. The submission should take the form of a **single Jupyter python notebook** (example: filename.ipynb) file, including code [in code cells] and accompanying narrative explanations [in Markdown Cells]. Appropriately mark each cell in your notebook, either as code or markdown. Don't forget to number your solutions.

Q1 [20 pts]

Write a function-based program in Python, which calculates a "*prognostic Score"* for cancer patients to stratify them into "*High/Low risk"* groups. The cancer patient prognostic score is derived from three variables: biomarker A (range 0.5 to 11.2), staging (I, II, III, or IV) and age (range 25-75).

a. Your program should have an outer function taking three arguments from a user: biomarker A, staging and age. **[4 pts]**

b. Create an inner function within this outer function, which divides the biomarker A value by patient age.

Next, it adds 1 to this result if the patient has been classed as Stage I or II; else add 10 to the result if classed as Stage III or IV. This is the final patient score. The outer function should then check whether the result of the inner function (final patient score) is greater or less than 6.185. If the value > threshold, the outer function should return 'High risk' along with the patient score. If the value < or equal to the threshold, the outer function should return 'low risk' together with the patient score. **[8 pts]**

c. Include exception handling if the variables entered by the user are outside the specified ranges. **[4 pts]**

d. Write an accompanying narrative to explain your implementation. **[4 pts]**

Q2 (30 pts)

Write a Python program (which may consist of several functions) to analyse the restriction sites in a DNA sequence. Specifically, the program should:

a. **Identify and count restriction sites** for the following enzymes in a given DNA sequence:

- EcoRI (site: "GAATTC")
- HindIII (site: "AAGCTT")

[5pts]

b. **Determine which of these enzymes has the higher number of restriction sites** in the sequence.

- If EcoRI has more sites, additionally count the restriction sites for:
- BamHI (site: "GGATCC")
- XhoI (site: "CTCGAG")

- Otherwise, if HindIII has more sites, count the restriction sites for:
  - PstI (site: "CTGCAG")
  - NotI (site: "GCGGCCGC")

[5pts]

c. **Output the results** as follows:

- Report the number of sites identified for EcoRI and HindIII.
- Report which enzyme between EcoRI and HindIII cuts the most often.
- Based on the results of step 2, report the number of sites for either the pair BamHI and XhoI or the pair PstI and NotI.

[4pts]

d. **Simulate Digestion with Multiple Enzymes:**

- Allow the user to specify multiple enzymes to "digest" the sequence.
- Then, identify sites for all specified enzymes and determine where they would cut/the size of resulting fragments.
- Filter fragments for those <=2000bp and plot the distribution as a histogram

[9pts]

e. Test the full program using the sequence files chr22.fasta and chr21.fasta **[3pts]**

f. Write an accompanying narrative to explain your implementation and the results using chr21.fasta and chr22.fasta. **[4pts]**

Note you may use the python standard library '**re'** and library **'matplotlib'**

Q3 [50 pts]

You are part of a Data Science team in a Bio-analytics company. As a consultant, you are helping a client company to understand datasets on Lung cancer to gain new insights. The first dataset (Q3-Lung_cancer_clinical) has clinical data of Lung cancer patients and was downloaded from the Cancer Genome Atlas portal (https://portal.gdc.cancer.gov/projects/TCGA-LUAD).

To find out more about the individual attributes in the clinical dataset, you can copy and search the term in the CDE browser. While the second dataset (Q3-LUADcancer_protein) has the protein expression data for the same set of patients and was downloaded from the Cancer Proteome portal (https://www.tcpaportal.org/tcpa/download.html).

To find out more about the proteins in this dataset, you can copy and search for the term on the My Protein section of the Cancer Proteome portal. The two datasets contain information on common patients.

However, the clinical dataset contains a greater number of patient records than corresponding protein expression data in the second dataset.

The clinical dataset has patient_barcode as the unique identifier, whereas, in the protein expression dataset the Sample_ID is used. In both datasets, the patient_barcode can be derived as: tissue_source_site"-"patient_id", e.g. TCGA-2H-A9GF.

Write a program in Python to help the client company perform the following tasks and include an accompanying narrative to explain your implementation for each section:

a. Link the two datasets by patient barcode, removing those patient records which appear in only one dataset. Create a new dataset combining both clinical and protein data for common patients. Give users the option to export out result either in CSV or TXT file format. **[15 pts.]**

b. Allow the user to select the combined dataset and then a minimum of two attributes. In an output to the user provide a data summary of the user-selected attributes. **[7 pts.]**

c. Allow the user to create a graphical representation (an appropriate plot/s) between the two user-selected attributes from the combined dataset. **[7 pts.]**

d. Allow the user to report on the number of missing values in the combined dataset. Report results, using percentages, by attribute and then by patient. **[7 pts.]**

e. Allow the user to specify a maximum number of attributes with missing values to tolerate. Allow the user to select and remove those patient samples which have attributes with missing values greater than the specified threshold. **[7 pts.]**

f. Write an accompanying narrative to explain your implementation **[7 pts.]**