TIE2030 Programming Methodology with Python
Take-Home Assignment (40 marks)

Release Date: 12 November, 2021 (~2pm)

Deadline: 16 November, 2021, 11pm

Late Submission Deadline (25% Penalty): 17 November, 2021, 11pm

No submissions will be accepted after 17 November, 2021, 11pm

INSTRUCTIONS – READ THE INSTRUCTIONS CAREFULLY BEFORE YOU START.

This is an individual assignment.

✓ **Download the following from TIE2030_TakeHomeAssignment folder:**

- Take home assignment (PDF) file – This is your assignment.
- **sequences.txt, motifs.txt** files from the same directory (data needed).
- **Template** (report) file (word doc) for you to capture all the results.

✓ **Do NOT share your code with anyone.** As this is a take-home assignment replacing your final exam, if you are found copying or have copied anyone's code or from the internet or from any other source, **marks will NOT be awarded for both the source and copiers**. No justifications will be accepted. As mentioned earlier, each one of you will have a different logic, design approach, and code. Hence, detecting plagiarism is easy in coding, as we did in your lab sessions. We will escalate the case to **disciplinary action** with the SCALE/Universityand you may face severe penalties. In the past, we have done so. **Do NOT DISCUSS on your logic and code with others.**

✓ As you have done all the lab assignments on your own, then you can handle this assignment alone and your design would be unique.

✓ This is an assignment that calls for your own effort in implementing your idea and design and you also need to decide on a few other things. If you need to assume anything you can do so, but you need to specify explicitly.

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore

You can consult us through the special **FORUM** (see **TIE2030_TakeHomeAssng**) which will be opened on – 13 November, 2021, 9am to 12 noon. After this time window it will be opened for anyone to visit and see the earlier queries and responses. New queries after 12 noon will not be addressed.

✓ Use the Forum in a wise way! You can consult **only** on anything that needs technical clarification and we will **NOT address anything related to the design and possible solution approaches**. Our answers would be predominantly, "Yes" / "May be" / "You can decide", if your questions point to solution approaches and design! So, unless it is an extremely important technical concern, do not post any queries related to your assumptions (Our answer will be "you can decide"). Only generic technical doubts and minor issues will be clarified.

✓ **EMAIL** **-** Individual emails from you will **NOT** be answered on this assignment. Use **-** **TIE2030_TakeHomeAssng** forum.

✓ Remember that this is an alternative assessment for your final exam and hence the **DEADLINE is STRICT**. You need to submit on or before the mentioned date above. After the above date, we will consider that you have not submitted it. **Suggestion** – Complete and try to submit **one/half-a day before the deadline**, to be safe. **No excuses if you do not submit within the deadline.**

✓ **Submission is only via LumiNUS folder (Use: TIE2030_StudentSubmission_TakeHomeAssng OR TIE2030_LATE Submission_TakeHomeAssng).**

✓ **Marks will be awarded for a variety of cases according to the Rubrics given on Page 9.**

✓ **Details on what you need to upload can be found on Page 9.**

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore

Problem Statement: In this assignment, you will be **searching for a given list of motifs** (small size fixed patterns) **in a given list of DNA sequences**. **A DNA sequence** is made up of fundamental Amino Acids – **A (adenine), G (guanine), C (cytosine), T (thymine)**. Motif finding is an important problem in the Bioinformatics domain. Motif finding helps to understand several common features between species, allows us to understand human diseases, and helps drug manufacturers to target towards manufacturing certain drugs.

Input and Output Data:

Following are **given** to you in this assignment.

- **List of DNA sequences***:* To be read from the **sequences.txt** file. Each line of the file contains one DNA sequence.
- **List of Motifs***:* To be read from the **motifs.txt** file. Each line of the file contains one motif.

Write a Python program to perform the following analysis. Your outputs must be written with clear messages to the **DNA_analysis_results.txt** file.

What do you need to do in this take-home assignment?

You need to write a program that **searches for ALL the motifs given to you in each DNA sequence,** generate the output and **store in a dictionary**. Also, **compare the list of sequences given to a target sequence** and report the statistics. You may use any built-in library function, if needed.

ANSWER ALL THE FOLLOWING:

Read through all the questions and the skeleton code in **Page 7** before you start coding. Use the skeleton code. All of the functions specified from Questions (2), (3), (4), (5), (7) must be **called** from your main() function [**Note**: No parameters are passed into main() function].

(1) In your **main()** function, read the motifs from the file **motifs.txt** and store them in a list. Create a dictionary ***Motif_Count_Dictionary*** that takes the motifs you have read from the file **motifs.txt** as keys, and their values are initialized to zero. Read the DNA sequences from the file **sequences.txt**. Write each DNA sequence and its length with a clear meaningful message

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore

to your output file **DNA_analysis_results.txt**. See the sample output in **Page 8**.

(5 Marks)

(2) **Write a Python function Nucleo_Counter(…)** and pass each of your DNA sequence and other parameters needed for this function. Call this function from the main() function. The function must count the **number of occurrences** (frequencies) of **each nucleotide A, G, C, T** in the DNA sequence you pass in. Write your counted values with a clear meaningful message to your output file **DNA_analysis_results.txt**. See the sample output in **Page 8**.

(5 Marks)

(3) **Write a Python function Motif_Counter(…)** and pass each of your DNA sequence, your ***Motif_Count_Dictionary***, and other parameters needed for this function. Call this function from the main() function. The function must count the **number of occurrences** (frequencies) of **each motif** that you have read from the file **motifs.txt** and accumulate the counts to the corresponding fields in your ***Motif_Count_Dictionary***. For example, the number of occurrences of motif **TC** must be added to the entry with key **TC** in your ***Motif_Count_Dictionary***.

Write your counted values with a clear meaningful message to your output file **DNA_analysis_results.txt**. See the sample output.

(5 Marks)

(4) **Write a Python function Freq_Counter(…)** to determine which motif **most frequently occurs** (maximum frequency) and which motif **least frequently occurs** (minimum frequency) in the given DNA sequences. Pass your ***Motif_Count_Dictionary*** and other parameters needed for this function. Call this function from the main() function. Write your results – **corresponding motifs** and their **frequencies**, with a clear meaningful message to your output file **DNA_analysis_results.txt**. See the sample output.

Important Note: If there are **more than one** motifs that occur most frequently and least frequently, write all of them to your output file.

(5 Marks)

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore

(5) Define a **target sequence** ***Target_Seq*** as following (refer to the skeleton code on **Page 7**):

Target_Seq = ‘ATGGGGAATGCGCAATGCAACGTAATTTAGAGGAGCCCCAGTTTGAAAGT’

Write a Python function Target_Search(…) to compare each sequence in your given DNA sequences against the target sequence ***Target_Seq***. Pass each of your DNA sequence and other parameters needed for this function. Call this function from your main() function. The function must perform the following:

Count the **number of elements matching exactly in the respective locations** between the DNA sequence you passed in and ***Target_Seq***. This gives the **“similarity”** between that DNA sequence and the target sequence ***Target_Seq.*** Return this value from your **Target_Search(…)** function to your main() function.

For example, given a target sequence:

ATGTAAAGCCTATAGTGGGGC

and a DNA sequence, say:

ATGTTTTGCCTATAGTATGGCATAGTAGTA

the **similarity score** between above example sequences is **16**.

After finding all the similarities, **in your main() function**, **find the sequences** that are **most similar** and the sequences that are **least similar** from ***Target_Seq***. Print your results with clear meaningful messages to your output file **DNA_analysis_results.txt**. Refer to the sample output.

Important Note: If there are **more than one sequences** that are most/least similar to the target sequence, write all of them to your output file.

(10 Marks)

(6) In your **main()** function, **measure the time taken to run your analysis as required from Question 1 to Question 5** (the time to run from the start of your program to the end of your code for Question 5). Write the time

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore

you measured with a clear meaningful message to your output file **DNA_analysis_results.txt.** See the sample output.

(2 Marks)

(7) **Write a Python function Plot_Chart(…)** to plot a **bar chart** that shows the **total number of occurrences** of each motif in all of the DNA sequences given in **sequences.txt** (the counts that you have accumulated for each motif in your ***Motif_Count_Dictionary***). Clearly present your chart with all the required information, title, and axis labels, as shown in the sample output. Pass your ***Motif_Count_Dictionary*** and other parameters needed for this function. Call this function from the main() function.

(3 Marks)

IMPORTANT FEATURES:

- Displaying your results with **meaningful messages** and clarity, writing **meaningful comments** (in addition to the comments given in the skeleton), and using **meaningful variable naming** also carry marks. Refer to the rubrics.

- In your output file, you need to print the length, counts of nucleotides, counts of motifs, and similarity to the target sequence for each DNA sequence **before proceeding to the next DNA sequence**. Refer to the sample output.

- Your code should be able to **give the correct results for different DNA sequences and motifs** (which also consist of **A, T, G, C** nucleotides), **without changing the code**. That is, if we change some motifs and sequences in **motifs.txt** and **sequences.txt,** and run your code, we expect the **correct results** for the new input data in your output file **DNA_analysis_results.txt.**

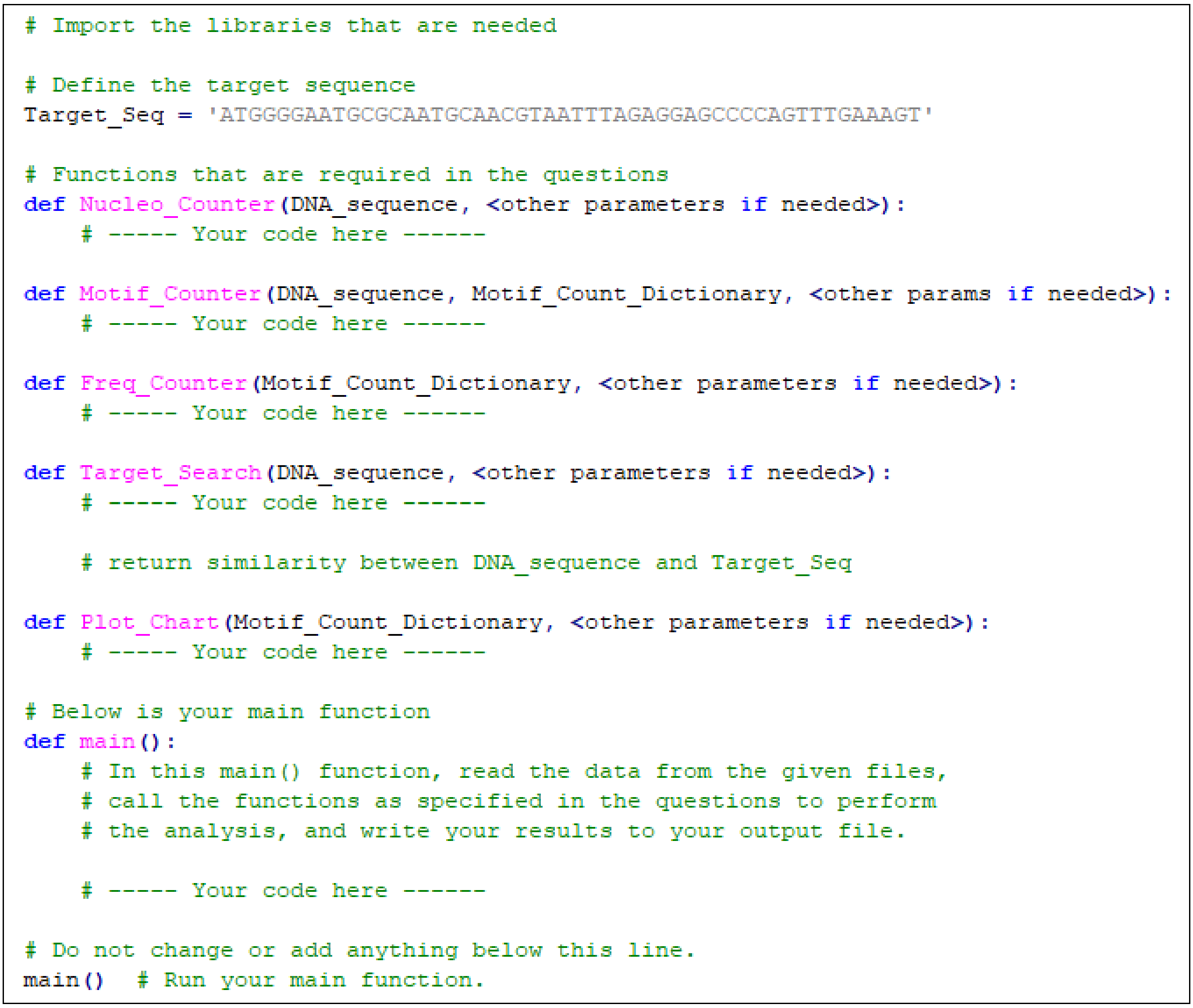
- For file writing, you can either write to the output file while processing or store your output messages in a list of strings and write to the output file at the end of the program.

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021.

SKELETON CODE:

Use the skeleton code below.

Note: you must NOT declare any variables outside the functions other than ***Target_Seq***. You are allowed to write additional functions if needed.

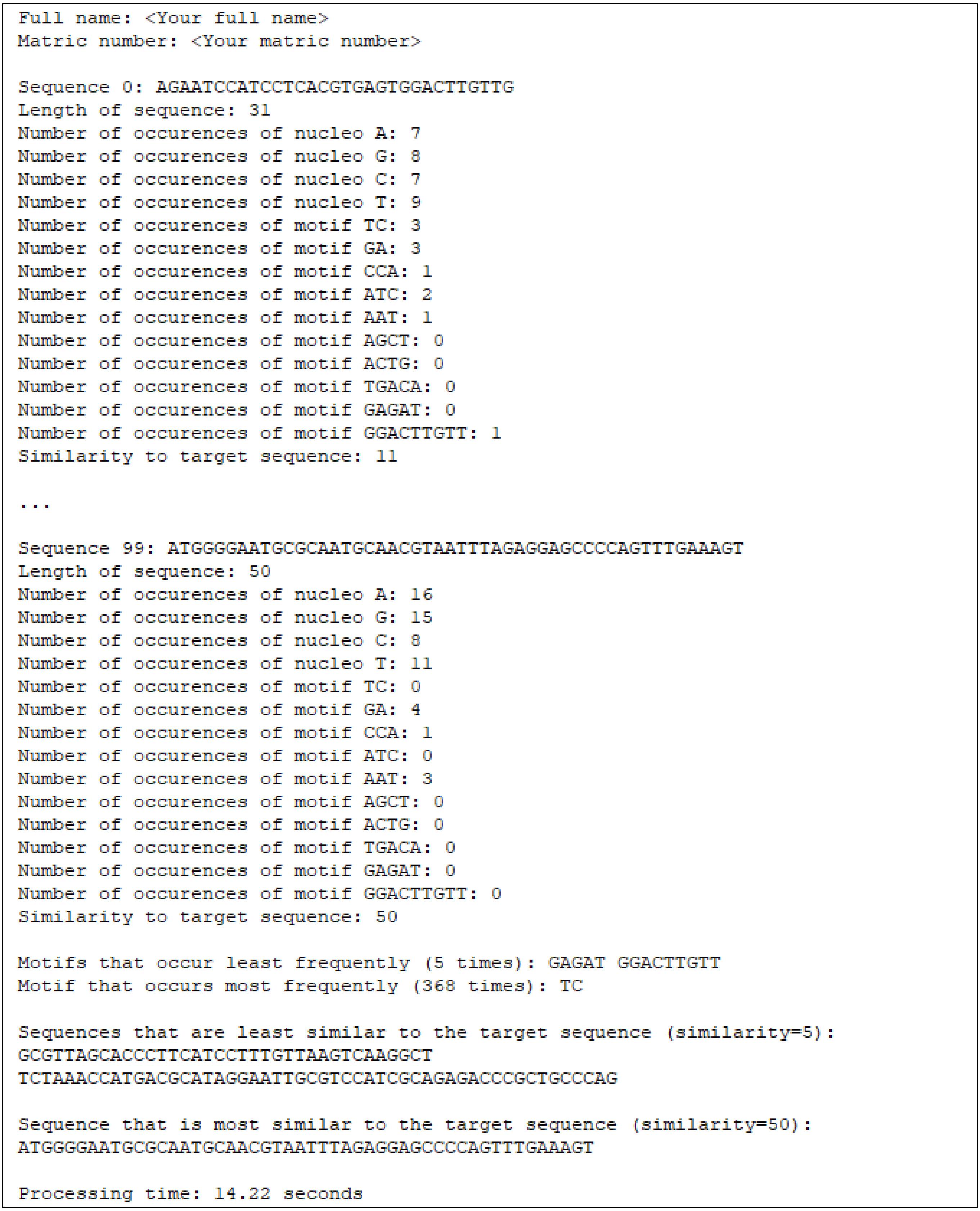


```
# Import the libraries that are needed

# Define the target sequence
Target_Seq = 'ATGGGGAATGCGCAATGCAACGTAATTTAGAGGAGCCCCAGTTTGAAAGT'

# Functions that are required in the questions
def Nucleo_Counter(DNA_sequence, <other parameters if needed>):
    # ----- Your code here ------

def Motif_Counter(DNA_sequence, Motif_Count_Dictionary, <other params if needed>):
    # ----- Your code here ------

def Freq_Counter(Motif_Count_Dictionary, <other parameters if needed>):
    # ----- Your code here ------

def Target_Search(DNA_sequence, <other parameters if needed>):
    # ----- Your code here ------

    # return similarity between DNA_sequence and Target_Seq

def Plot_Chart(Motif_Count_Dictionary, <other parameters if needed>):
    # ----- Your code here ------

# Below is your main function
def main():
    # In this main() function, read the data from the given files,
    # call the functions as specified in the questions to perform
    # the analysis, and write your results to your output file.

    # ----- Your code here ------

# Do not change or add anything below this line.
main()   # Run your main function.
```

SAMPLE OUTPUT:

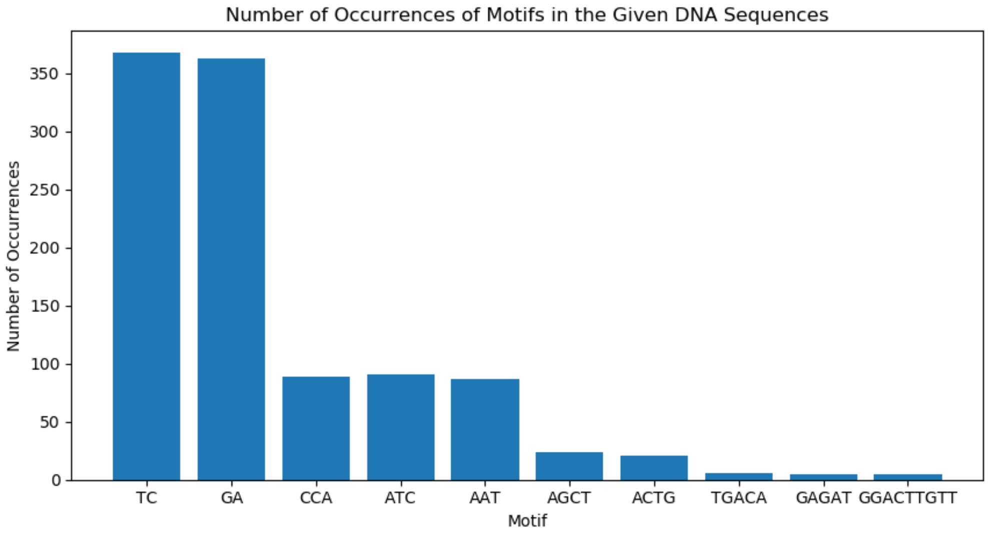
Important note: The results shown in the sample output below are obtained using DIFFERENT data from the files given to you. Therefore, the results shown below DO NOT match the results you obtain using the data in **sequences.txt** and **motifs.txt**. You should compute and check your results by yourself using the data in the two given files.

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021.
Lecturer: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore



```
Full name: <Your full name>
Matric number: <Your matric number>

Sequence 0: AGAATCCATCCTCACGTGAGTGGACTTGTTG
Length of sequence: 31
Number of occurences of nucleo A: 7
Number of occurences of nucleo G: 8
Number of occurences of nucleo C: 7
Number of occurences of nucleo T: 9
Number of occurences of motif TC: 3
Number of occurences of motif GA: 3
Number of occurences of motif CCA: 1
Number of occurences of motif ATC: 2
Number of occurences of motif AAT: 1
Number of occurences of motif AGCT: 0
Number of occurences of motif ACTG: 0
Number of occurences of motif TGACA: 0
Number of occurences of motif GAGAT: 0
Number of occurences of motif GGACTTGTT: 1
Similarity to target sequence: 11

...

Sequence 99: ATGGGGAATGCGCAATGCAACGTAATTTAGAGGAGCCCCAGTTTGAAAGT
Length of sequence: 50
Number of occurences of nucleo A: 16
Number of occurences of nucleo G: 15
Number of occurences of nucleo C: 8
Number of occurences of nucleo T: 11
Number of occurences of motif TC: 0
Number of occurences of motif GA: 4
Number of occurences of motif CCA: 1
Number of occurences of motif ATC: 0
Number of occurences of motif AAT: 3
Number of occurences of motif AGCT: 0
Number of occurences of motif ACTG: 0
Number of occurences of motif TGACA: 0
Number of occurences of motif GAGAT: 0
Number of occurences of motif GGACTTGTT: 0
Similarity to target sequence: 50

Motifs that occur least frequently (5 times): GAGAT GGACTTGTT
Motif that occurs most frequently (368 times): TC

Sequences that are least similar to the target sequence (similarity=5):
GCGTTAGCACCCTTCATCCTTTGTTAAGTCAAGGCT
TCTAAACCATGACGCATAGGAATTGCGTCCATCGCAGAGACCCGCTGCCCAG

Sequence that is most similar to the target sequence (similarity=50):
ATGGGGAATGCGCAATGCAACGTAATTTAGAGGAGCCCCAGTTTGAAAGT

Processing time: 14.22 seconds
```

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore



RUBRICS:

Part	Description	Marks
1	For reading data from files; initiate the ***Motif_Count_Dictionary***; printing the DNA sequences and their lengths to the output file.	5
2	For the logic and execution of the **Nucleo_Counter(…)** function.	5
3	For the logic and execution of the **Motif_Counter(…)** function.	5
4	For the logic and execution of the **Freq_Counter(…)** function.	5
5	For the logic and execution of the **Target_Search(…)** function; for finding the DNA sequences that are most and least similar to the target sequence.	10
6	For measuring the processing time.	2
7	For the logic and execution of the **Plot_Chart(…)** function; displaying the chart clearly with all the required information.	3
Coding quality, output display	For writing clear code and comments; using meaningful variable names; printing outputs with clear messages; following skeleton code and other requirements in the questions and template; use of function and parameter passing.	5
Total		**40**

WHAT IS THAT YOU NEED TO UPLOAD?

Upload a **.zip** file in the **StudentSubmission_TakeHomeAssng_TIE2030** folder with name: **<MATRIC_NUMBER>_FINAL_ASSIGNMENT_< FIRST_NAME>.zip** containing the following files:

(1) Your working code **CODE_<MATRIC_NUMBER>_FINAL_ASSIGNMENT_<FIRST_NAME>.py**

(2) Your report (convert to PDF after you complete)
REPORT_<MATRIC_NUMBER>_FINAL_ASSIGNMENT_<FIRST_NAME>.pdf

(3) Your output file **DNA_analysis_results.txt**.

Refer to the template. Please follow the file naming strictly.

✓ [1] Take-Home Assignment as an alternative assessment for TIE2030 Final Exam for AY2122, Semester I, Nov 2021. **Lecturer**: A/Prof Bharadwaj Veeravalli, Dept of ECE, NUS, Singapore