PRACTICAL VII

DATE: 16/03/24

BioPython

AIM: To learn and execute Biopython.

INTRODUCTION:

Biopython is the most popular molecular biology package for computation. Brad Chapman and Jeff Chang developed it in 1999. It is mainly written in python but some C code is there to solve complex optimization. Biopython is capable of a lot like it can do protein structure, sequence motifs, sequence alignment also machine learning.

Biopython has a lot of libraries for the help of biologists in their work as it is portable, easy, and clear. In some cases Google images use Biopython, BioPerl, BioJAVA, BioRuby are the siblings of the Biopython project.

Reasons for development

- Creating high quality, reusable classes for the complex bio-informatics problems.
- Read genetic databases like Swissport, FASTA, and many more. After reading, it parse them to python utilizable data structure.
- For genomic data analysis.
- Tools for protein structure and have BigSQL-storing a lot of data.

Advantages

- Good read and write for tree-view files.
- Support micro-array data type for clustering.
- Parsing bio-informatics files into a formalized object or in a generic class.

Sample Cases and Features

- **RNA structure:** RNA is one of the major macromolecules in our day-to-day life, others are DNA and Protein. DNA is called the blueprint of a cell and RNA acts as 'DNA photocopy' in the cell. For making this structural work easy *Biopython* has Bio.sequence object for DNA, RNA block representation.
- **Genome Diagram:** This module provides visualization of sequences in the PDF or JPG format. Multiple sequence compare is possible using this module.

- **SeqRecord:** Bio.sequence provides only the sequence. *SeqRecord* class provide name, description, feature along with the sequence.
- **Population Genetics:** It is the study of checking genetic variation in a population. It works with the examination, modeling of genes. *Bio.PopGen* module is there to simplify the work.
- **Phylogeny:** As I mentioned earlier *Biopython* has the advantage of using tree-view files. It uses Bio. *Phylo* module visualizing tree as well as manipulation and traversal from getting input in *Newick*, *PhyloXML* file format.

Q1. Write a biopython script to store DNA sequence using alphabet class. Perform following operation:

a. To get the second value in sequence

b. To print the first two values

c. To perform length and count number of 'g'

d. Create second sequence and to add two sequences and display

e. Turn a Seq object into a string

```
from Bio.Seq import Seq

dna_sequence = input("Enter DNA sequence: ")

seq_obj = Seq(dna_sequence)

second_value = seq_obj[1]

print("Second value in sequence:", second_value)

first_two_values = seq_obj[:2]

print("First two values:", first_two_values)

sequence_length = len(seq_obj)

print("Length of the sequence:", sequence_length)

count_g = seq_obj.count('G')

print("Number of 'G' in the sequence:", count_g)

second_sequence = Seq("ATCGATCG")

added_sequence = seq_obj + second_sequence

print("Added sequences:", added_sequence)

seq_string = str(seq_obj)

print("Seq object turned into a string:", seq_string)
```

```
Enter DNA sequence: attggcttc
Second value in sequence: t
First two values: at
Length of the sequence: 9
Number of 'G' in the sequence: 0
Added sequences: attggcttcATCGATCG
Seq object turned into a string: attggcttc
```

Figure 1: Output of Q1

Q2. Write a biopython script to store dna sequence. Perform following operation:

a. Find reverse complement

b. Calculate GC percentage in dna sequence

c. Convert dna to rna

d. Convert dna to protein

```
from Bio.Seq import Seq
# Get DNA sequence from user
dna_sequence = input("Enter DNA sequence: ")
# Create a Seq object from the input sequence
seq_obj = Seq(dna_sequence)
# Find reverse complement
reverse_complement = seq_obj.reverse_complement()
print("Reverse complement:", reverse_complement)
# Calculate GC percentage
gc_content = (seq_obj.count("G") + seq_obj.count("C")) / len(seq_obj) * 100
print("GC Percentage:", gc_content)
# Convert DNA to RNA
rna_sequence = seq_obj.transcribe()
print("RNA Sequence:", rna_sequence)
# Convert DNA to protein
protein_sequence = seq_obj.translate()
print("Protein Sequence:", protein_sequence)
```

```
Enter DNA sequence: ATAATGAACAGC
Reverse complement: GCTGTTCATTAT
GC Percentage: 33.3333333333333
RNA Sequence: AUAAUGAACAGC
Protein Sequence: IMNS
```

Figure 2: Output of Q2

Q3. Write a biopython script to store protein sequence using alphabet class and calculate molecular weight.

```
from Bio.SeqUtils import molecular_weight

from Bio.Seq import Seq

str = input("\nEnter a DNA sequence: ")

seq = Seq(str)

print("DNA sequence is", seq)

print("Converting DNA to RNA sequence :", seq.transcribe())

print("Converting RNA sequence to protein:", seq.translate())

print("Molecular weight of the sequence is", molecular_weight(seq))
```

```
Enter a DNA sequence: AGAACGGATTAG
DNA sequence is AGAACGGATTAG
Converting DNA to RNA sequence : AGAACGGAUUAG
Converting RNA sequence to protein: RTD*
Molecular weight of the sequence is 3798.4396
```

Figure 3: Output of Q3

Q4. Write a biopython script to create sequence using sequence record by adding parameters like seq,id, Name,description and display the whole sequence record and also access only sequence id and sequence.

```
from Bio.Seq import Seq

from Bio.SeqRecord import SeqRecord

seq_id = input("Enter sequence ID: ")

seq_name = input("Enter sequence name: ")

seq_description = input("Enter sequence description: ")
```

```
sequence = input("Enter sequence: ")

seq_record = SeqRecord(Seq(sequence), id=seq_id, name=seq_name,
description=seq_description)

print("\nEntire Sequence Record:")

print(seq_record)

print("\nAccessing Sequence ID and Sequence:")

print("Sequence ID:", seq_record.id)

print("Sequence:", seq_record.seq)
```

```
Enter sequence ID: SEQ001
Enter sequence name: Homosapein
Enter sequence description: This is a dna sequence
Enter sequence: AATTAGAGATAG

Entire Sequence Record:
ID: SEQ001
Name: Homosapein
Description: This is a dna sequence
Number of features: 0
Seq('AATTAGAGATAG')

Accessing Sequence ID and Sequence:
Sequence ID: SEQ001
Sequence: AATTAGAGATAG
```

Figure 4: Output of Q4

Q5. Write a biopython script to read sequence which is downloaded from genbank i.e fasta file and save named as NC_005816.fna. Peform the following:

a. Display id and seq

b. Find reverse complement

c. Calculate GC percentage in dna sequence

d. Convert dna to rna

e. Convert dna to protein

```
from Bio import SeqIO

from Bio.Seq import Seq

file_path = "C:\\Users\\user\\Desktop\\New folder (2)\\NC_005816.fna"  # Update the file
path accordingly

record = SeqIO.read(file_path, "fasta")
```

```
print("Sequence ID:", record.id)
print("Sequence:", record.seq)
reverse_complement = record.seq.reverse_complement()
print("\nReverse Complement:", reverse_complement)
gc_percentage = record.seq.count("G") + record.seq.count("C")
gc_percentage /= len(record.seq)
gc_percentage *= 100
print("\nGC Percentage:", gc_percentage)

rna_sequence = record.seq.transcribe()
print("\nRNA Sequence:", rna_sequence)
protein_sequence = record.seq.translate()
print("\nProtein Sequence:", protein_sequence)
```

```
Sequence ID: HSBGPG
Sequence: GGCAGATTCCCCCTAGACCCGCCCGCACCATGGTCAGGCATGCCCCTCCTCATCGCTGGGCACAGCCCAGAGGGTATAAACAGTGCTGGAGGCTGGCGGGGCAGGCCAGCTGAGTCCTGAGCAGCAGCCCAGCGCAGCCACCGAGACACCATG
AGAGCCCTCACACTCCTCGCCCTATTGGCCCTGGCCGCACTTTGCATCGCTGGCCAGGCAGGTGAGTGCCCCCACCTCCCCTCAGGCCGCATTGCAGTGGGGGCTGAGAGGAGGAAGCACCATGGCCCACCTCTTCTCACCCCTTTGGCTGGCAGTCCCTTTG
CAGTCTAACCACCTTGTTGCAGGCTCAATCCATTTGCCCCAGCTCTGCCCTTGCAGAGGGAGAGGAGGGAAGAGCAAGCTGCCCGAGACGCAGGGGAAGGAGGATGAGGGCCCTGGGGATGAGCTGGGGTGAACCAGGCTCCCTTTCCTTTGCAGGTGCGAAG
CCCAGCGGTGCAGAGTCCAGCAAAGGTGCAGGTATGAGGATGGACCTGATGGGTTCCTGGACCCTCCCCTCTCACCCTGGTCCCTCAGTCTCATTCCCCCACTCCTGCCACCTCCTGTCTGGCCATCAGGAAGGCCAGCCTGCTCCCCACCTGATCCTCCCAA
ACCCAGAGCCACCTGATGCCTGCCCCTCTGCTCCACAGCCTTTGTGTCCAAGCAGGAGGGCAGCGAGGTAGTGAAGAGACCCAGGCGCTACCTGTATCAATGGCTGGGGTGAGAGAAAAGGCAGAGCTGGGCCAAGGCCCTGCCTCTCCGGGATGGTCTGTGG
GGGAGCTGCAGCAGGGAGTGGCCTCTCTGGGTTGTGGTGGGGGTACAGGCAGCCTGCCCTGGTGGGCACCCTGGAGCCCCATGTGTAGGGAGAGGAGGGATGGGCATTTTGCACGGGGGCTGATGCCACCACGTCGGGTGTCTCAGAGCCCCAGTCCCCTACC
CGGATCCCCTGGAGCCCAGGAGGGAGGTGTGTGAGCTCAATCCGGACTGTGACGAGTTGGCTGACCACATCGGCTTTCAGGAGGCCTATCGGCGCTTCTACGGCCCGGTCTAGGGTGTCGCTCTGCTGGCCTGGCCGGCAACCCCAGTTCTGCTCCTCTCCAG
GCACCCTTCTTTCCTCTTCCCCTTGCCCTTGCCCTGACCTCCCAGCCCTATGGATGTGGGGTCCCCATCATCCCAGCTGCTCCCAAATAAACTCCAGAAG

Reverse Complement: CTTCTGGAGTTTATTTGGGAGCAGCTGGGATGATGGGGACCCCACATCCATAGGGCTGGGAGGTCAGGGCAAGGGCAAGGGGAAGAGGAAAGAAGGGTGCCTGGAGAGGAGCAGAACTGGGGTTGCCGGCCAGGCCAGCAGAG
CGACACCCTAGACCGGGCCGTAGAAGCGCCGATAGGCCTCCTGAAAGCCGATGTGGTCAGCCAACTCGTCACAGTCCGGATTGAGCTCACACACCTCCCTCCTGGGCTCCAGGGGATCCGGGTAGGGGACTGGGGCTCTGAGACACCCGACGTGGTGGCATCA
GCCCCCGTGCAAAATGCCCATCCCTCCTCTCCCTACACATGGGGCTCCAGGGTGCCCACCAGGGCAGGCTGCCTGTACCCCCACCACAACCCAGAGAGGCCACTCCCTGCTGCAGCTCCCCCACAGACCATCCCGGAGAGGCAGGGCCTTGGCCCAGCTCTGC
CTTTTCTCTCACCCCAGCCATTGATACAGGTAGCGCCTGGGTCTCTTCACTACCTCGCTGCCCTCCTGCTTGGACACAAAGGCTGTGGAGCAGAGGGGCAGGCATCAGGTGGCTCTGGGTTTGGGAGGATCAGGTGGGGAGCAGGCTGGCCTTCCTGATGGCC
AGACAGGAGGTGGCAGGAGTGGGGGAATGAGACTGAGGGACCAGGGTGAGAGGGGAGGGTCCAGGAACCCATCAGGTCCATCCTCATACCTGCACCTTTGCTGGACTCTGCACCGCTGGGCTTCGCACCTGCAAAGGAAAGGGAGCCTGGTTCACCCCAGCTC
ATCCCCAGGGCCCTCATCCTCCTTCCCCTGCGTCTCGGGCAGCTTGCTCTTCCCTCCTCTCCCTCTGCAAGGGCAGAGCTGGGGCAAATGGATTGAGCCTGCAACAAGGTGGTTAGACTGCAAAGGGACTGCCAGCCAAAGGGGTGAGAAGAGGTGGGCCATG
GTGCTTCCTCCTCTCAGCCCCCACTGCAATGCGGCCTGAGGGGAGGTGGGGGCACTCACCTGCCTGGCCAGCGATGCAAAGTGCGGCCAGGGCCAATAGGGCGAGGAGTGTGAGGGCTCTCATGGTGTCTCGGTGGCTGCGCTGGGCTGCTGCTCAGGACTCA
GCTGGCCTGCCCCGCCAGCCTCCAGCACTGTTTATACCCTCTGGGCTGTGCCCAGCGATGAGGAGGGGCATGCCTGACCATGGTGCGGGCGGGTCTAGGGGGAATCTGCC

GC Percentage: 63.52558895207149

RNA Sequence: GGCAGAUUCCCCCUAGACCCGCCCGCACCAUGGUCAGGCAUGCCCCUCCUCAUCGCUGGGCACAGCCCAGAGGGUAUAAACAGUGCUGGAGGCUGGCGGGGCAGGCCAGCUGAGUCCUGAGCAGCAGCCCAGCGCAGCCACCGAGACAC
CAUGAGAGCCCUCACACUCCUCGCCCUAUUGGCCCUGGCCGCACUUUGCAUCGCUGGCCAGGCAGGUGAGUGCCCCCACCUCCCCUCAGGCCGCAUUGCAGUGGGGGCUGAGAGGAGGAAGCACCAUGGCCCACCUCUUCUCACCCCUUUGGCUGGCAGUCCC
UUUGCAGUCUAACCACCUUGUUGCAGGCUCAAUCCAUUUGCCCCAGCUCUGCCCUUGCAGAGGGAGAGGAGGGAAGAGCAAGCUGCCCGAGACGCAGGGGAAGGAGGAUGAGGGCCCUGGGGAUGAGCUGGGGUGAACCAGGCUCCCUUUCCUUUGCAGGUGC
GAAGCCCAGCGGUGCAGAGUCCAGCAAAGGUGCAGGUAUGAGGAUGGACCUGAUGGGUUCCUGGACCCUCCCCCUCUCACCCUGGUCCCUCAGUCUCAUUCCCCCACUCCUGCCACCUCCUGUCUGGCCAUCAGGAAGGCCAGCCUGCUCCCCACCUGAUCCUC
CCAAACCCAGAGCCACCUGAUGCCUGCCCCUCUCUGCUCCACAGCCUUUGUGUCCAAGCAGGAGGGCAGCGAGGUAGUGAAGAGACCCAGGCGCUACCUGUAUCAAUGGCUGGGGUGAGAGAAAAGGCAGAGCUGGGCCAAGGCCCUGCCUCUCCCGGGAUGGUCU
GUGGGGGAGCUGCAGCAGGGAGUGGCCUCUCUGGGUUGUGGUGGGGGUACAGGCAGCCUGCCCUGGUGGGCACCCUGGAGCCCCAUGUGUAGGGAGAGGAGGGAUGGGCAUUUUGCACGGGGGCUGAUGCCACCACGUCGGGUGUCUCAGAGCCCCAGUCCCC
UACCCGGAUCCCCUGGAGCCCAGGAGGGAGGUGUGUGAGCUCAAUCCGGACUGUGACGAGUUGGCUGACCACAUCGGCUUUCAGGAGGCCUAUCGGCGCUUCUACGGCCCGGUCUAGGGUGUCGCUCUGCUGGCCUGGCCGGCAACCCCAGUUCUGCUCCUCU
CCAGGCACCCUUCUUUCCUCUUCCCCUUGCCCUUGCCCUGACCUCCCAGCCCUAUGGAUGUGGGGUCCCCAUCAUCCCAGCUGCUCCCAAAUAAACUCCAGAAG

Warning (from warnings module):
  File "C:\Users\user\AppData\Local\Programs\Python\Python312\Lib\site-packages\Bio\Seq.py", line 2880
    warnings.warn(
BiopythonWarning: Partial codon, len(sequence) not a multiple of three. Explicitly trim the sequence or add trailing N before translation. This may become an error
in future.

Protein Sequence: GRFPLDPPAPWSGMPLLIAGHSPEGINSAGGWRGRPAES*AAAQRSHRDTMRALTLLALLALAALCIAGQAGECPHLPSGRIAVGAERRKHHGPPLLTPLAGSPFAV*PPCCRLNPFAPALPLQRERREEQAARDAGEGG*GPWG
*AGVNQAPFPLQVRSPAVQSPAKVQV*GWT*WVPGPSPLTLVPQSHSPTPATSCLAIRKASLLPT*SSQTQSHLMPAPLLHSLCVQAGGQRGSEETQALPVSMAGVREKAELGQGPASPGWSVGELQQGVASLGCGGGTGSLEWWAPWSPMCRERRDGHFARG
LMPPRRVSQSPSPLPGSPGAQEGGV*AQSGL*RVG*PHRLSGGLSALLRPGLGCRSAGLAGNPSSAPLQAPFFPLPLALALTSQPYGCGVPIIPAAPK*TPE
```

Figure 5: Output of Q5

Q6. Write a biopython script to access sequence record from genbank directly

```
from Bio import Entrez
from Bio import SeqIO
# Provide your email address to NCBI
```

```
Entrez.email = "your.email@example.com"

# Accession number of the sequence record in GenBank

accession_number = "DJ484126.1"

# Fetch the sequence record from GenBank

handle = Entrez.efetch(db="nucleotide", id=accession_number, rettype="fasta",
retmode="text")

record = SeqIO.read(handle, "fasta")

handle.close()

# Display the sequence record

print("Sequence ID:", record.id)

print("Sequence Description:", record.description)

print("Sequence:", record.seq)
```

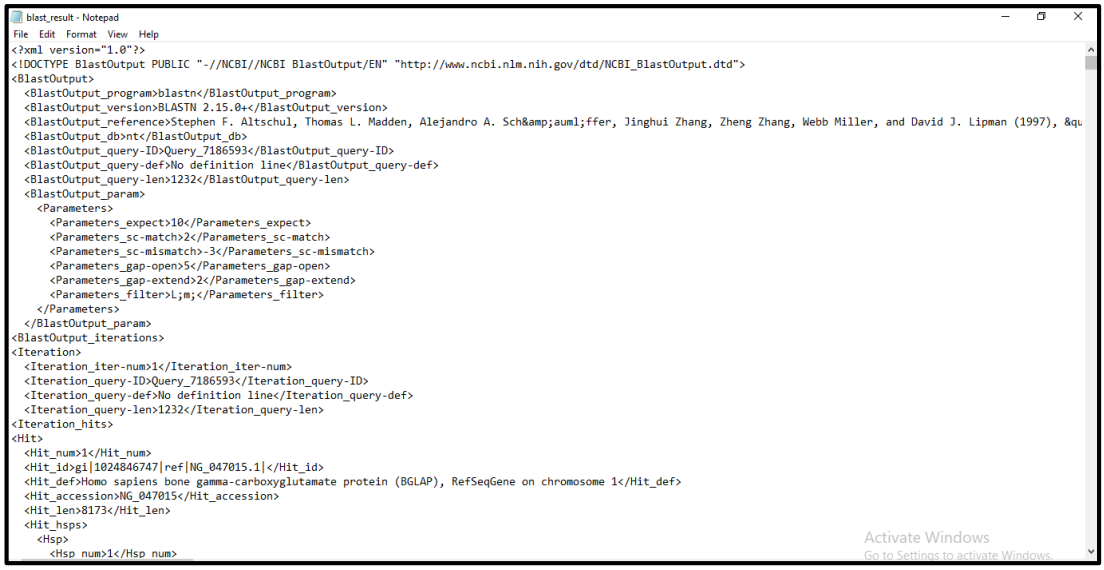
```
Sequence ID: DJ484126.1
Sequence Description: DJ484126.1 MYOSIN-LIKE GENE EXPRESSED IN HUMAN HEART AND MUSCLE
Sequence: GCCACAGATACTATGAG
```

Figure 6: Output of Q6

Q7 Write a biopython script to any read fasta file query using sequence I/O and run that query using blast with any blast algorithm. Save the output file in xml format.

```
from Bio import SeqIO

from Bio.Blast import NCBIWWW

query = SeqIO.read("C:\\Users\\user\\Desktop\\New folder (2)\\test.fasta", format='fasta')

result_handle = NCBIWWW.qblast("blastn", "nt", query.seq)

blast_file = open("C:\\Users\\user\\Desktop\\New folder (2)\\blast_result.xml", "w")

blast_file.write(result_handle.read())

blast_file.close()

result_handle.close()
```

blast_result	3/17/2024 7:51 AM	Microsoft Edge H...	215 KB
test	3/17/2024 7:38 AM	FASTA File	2 KB



blast_result - Notepad

File Edit Format View Help


```
<?xml version="1.0"?>
<!DOCTYPE BlastOutput PUBLIC "-//NCBI//NCBI BlastOutput/EN" "http://www.ncbi.nlm.nih.gov/dtd/NCBI_BlastOutput.dtd">
<BlastOutput>
  <BlastOutput_program>blastn</BlastOutput_program>
  <BlastOutput_version>BLASTN 2.15.0+</BlastOutput_version>
  <BlastOutput_reference>Stephen F. Altschul, Thomas L. Madden, Alejandro A. Sch&amp;amp;auml;ffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), &qu
  <BlastOutput_db>nt</BlastOutput_db>
  <BlastOutput_query-ID>Query_7186593</BlastOutput_query-ID>
  <BlastOutput_query-def>No definition line</BlastOutput_query-def>
  <BlastOutput_query-len>1232</BlastOutput_query-len>
  <BlastOutput_param>
    <Parameters>
      <Parameters_expect>10</Parameters_expect>
      <Parameters_sc-match>2</Parameters_sc-match>
      <Parameters_sc-mismatch>-3</Parameters_sc-mismatch>
      <Parameters_gap-open>5</Parameters_gap-open>
      <Parameters_gap-extend>2</Parameters_gap-extend>
      <Parameters_filter>L;m;</Parameters_filter>
    </Parameters>
  </BlastOutput_param>
<BlastOutput_iterations>
<Iteration>
  <Iteration_iter-num>1</Iteration_iter-num>
  <Iteration_query-ID>Query_7186593</Iteration_query-ID>
  <Iteration_query-def>No definition line</Iteration_query-def>
  <Iteration_query-len>1232</Iteration_query-len>
<Iteration_hits>
<Hit>
  <Hit_num>1</Hit_num>
  <Hit_id>gi|1024846747|ref|NG_047015.1|</Hit_id>
  <Hit_def>Homo sapiens bone gamma-carboxyglutamate protein (BGLAP), RefSeqGene on chromosome 1</Hit_def>
  <Hit_accession>NG_047015</Hit_accession>
  <Hit_len>8173</Hit_len>
  <Hit_hsps>
    <Hsp>
      <Hsp_num>1</Hsp_num>
```


Activate Windows
Go to Settings to activate Windows.

Figure 7: Output of Q7