TILAK PUBLIC SCHOOL JAIPUR

AISSCE- IP Project 2023-24

Project Report on

Data Wallah

Submitted to:
Mr. Praveen Gupta
(PGT, IP)

Submitted By:
Arnav Vijay

TILAK PUBLIC SCHOOL

CERTIFICATE

This is to certify that the Informatics Practices titled **"Data_Wallah"** has been Successfully completed by **Arnav vijay** of class **XII - A,** Under the guidance of **Mr. Praveen Gupta.** This project is submitted as partial fulfillment for **AISSCE** 2023 Practical Examination

Date:____

Mr. Praveen Gupta
(Internal Examiner)

Raksha Shekhawat
(Principal)

ACKNOWLEDGEMENT

I solemnly take the opportunity to thank all the helping hands who made me to complete this project. First of all I thank the Almighty for keeping me hale and healthy in order to successfully complete my work.

I wish to express my sincere gratitude to **Mrs. Raksha Shekhawat**, Principal of Tilak Public School, Triveni Nagar, Jaipur for permitting me to carry out the project and for the facilities he has provided for the fulfillment of this project work.

I am greatly indebted to **Mr. Praveen Gupta**, Teacher in Informatics Practices who gave me immense support and guidance throughout the completion of this project.

Sincerely,

Arnav Vijay

TABLE OF CONTENTS

Problem definition

With the exponential growth of genomic data, there is a need for a comprehensive bioinformatics toolkit to facilitate genomic analysis using Python.

The project aims to develop a versatile toolkit that covers a range of functionalities for the analysis and interpretation of genomic data and generation of several genetic codes.

Team Details

1. ***Arnav Vijay***
2. ***Divyansh Sharma***

The Project ***"Data Wallah"*** is going too developed by *Arnav Vijay* and *Divyansh Sharma*. It take approx. 30 days to develop this project, working approx. 1 Hour daily. All modules will be completed by us only as per our view and knowledge.

Reason for choosing this topic

Understanding DNA and RNA manipulation is fundamental in the field of molecular biology and bioinformatics. Creating a program that performs basic operations on these sequences can serve as a practical educational tool.

1. **Bioinformatics Exploration:**
    - Bioinformatics involves the use of computational tools to analyse biological data. Developing a program that handles DNA and RNA sequences allows individuals to explore bioinformatics concepts and gain hands-on experience.
2. **Relevance to Biology:**
    - DNA and RNA are essential components of living organisms, and studying their properties and interactions provides insights into genetics, evolution, and disease. A program focusing on DNA and RNA manipulation aligns with the biological relevance of these molecules.
3. **Programming Practice:**
    - Developing a program that involves string manipulation, data analysis, and user interaction provides an opportunity for individuals to practice their programming skills in Python.
4. **Applicability in Research:**
    - The ability to manipulate and analyse DNA and RNA sequences is crucial in various research areas, including genetics, genomics, and medicine. A program that performs basic operations on these sequences can serve as a starting point for more complex bioinformatics tools.
5. **Interactive Learning:**
    - Creating a user-friendly program with an interactive menu makes it accessible to a wide audience, including students, researchers, and enthusiasts. It promotes hands-on learning and experimentation.
6. **Open-Ended Nature:**
    - DNA and RNA manipulation programs can be expanded with additional features and functionalities. This allows developers to continuously improve and extend the program, making it a flexible and ongoing project.
7. **Combining Biology and Programming:**
    - For individuals with an interest in both biology and programming, a DNA and RNA sequence manipulation program provides a unique opportunity to bridge these two disciplines, fostering interdisciplinary skills.

Objective

- **User**

Introducing users to the field of bioinformatics by incorporating fundamental concepts such as sequence manipulation and analysis.

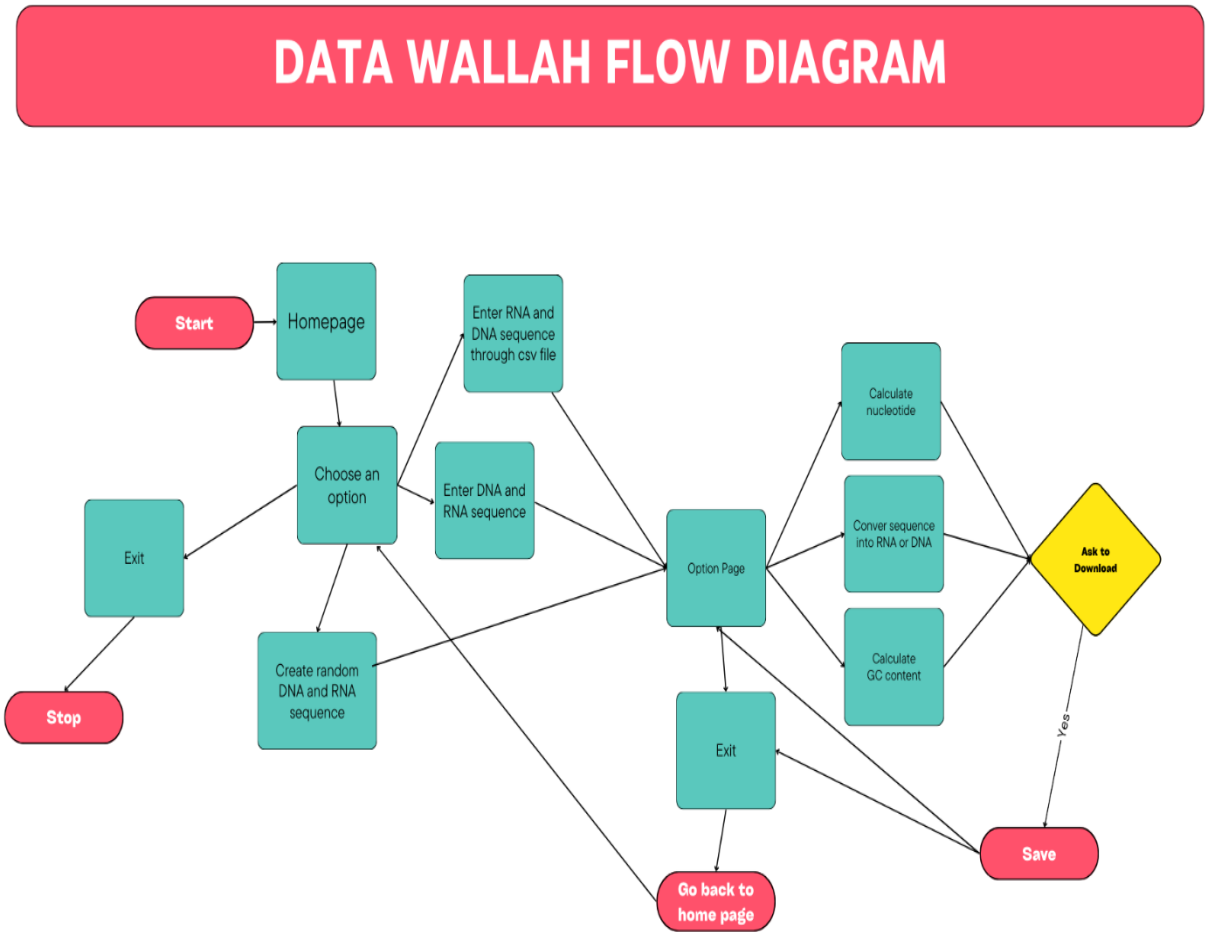
- **Environment**

Foster an environment where users can contribute to the program's improvement, either by suggesting enhancements or directly contributing code.

- **Download**

We are providing the facility handling biological data, such as reading sequences from CSV files and saving results.

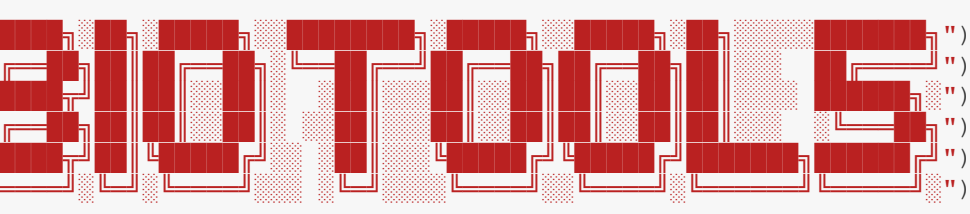
Designing Tools



CODE

In [ ]:

```
import random
import pandas as pd

def generate_random_dna_sequence(length):
    return ''.join(random.choice('ACGT') for _ in range(length))

def generate_random_rna_sequence(length):
    return ''.join(random.choice('ACGU') for _ in range(length))

def calculate_gc_content(dna_sequence):
    gc_count = dna_sequence.count('G') + dna_sequence.count('C')
    return (gc_count / len(dna_sequence)) * 100

def calculate_nucleotide_frequency(dna_sequence):
    nucleotide_count = {'A': 0, 'C': 0, 'G': 0, 'T': 0}
    for nucleotide in dna_sequence:
        if nucleotide in nucleotide_count:
            nucleotide_count[nucleotide] += 1
    return nucleotide_count

def dna_to_rna(dna_sequence):
    return dna_sequence.replace('T', 'U')

def rna_to_dna(rna_sequence):
    return rna_sequence.replace('U', 'T')

def read_csv(file_path):
    try:
        df = pd.read_csv(file_path)
        return df['DNA Sequence'].tolist()
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        return None

def read_rna_sequence(file_path):
    try:
        df = pd.read_csv(file_path)
        return df['RNA Sequence'].tolist()
    except FileNotFoundError:
        print(f"Error: File '{file_path}' not found.")
        return None

def save_results_to_csv(result, file_path):
    df = pd.DataFrame([result], columns=['DNA Sequence', 'GC Content', 'RNA Sequence'])
    df.to_csv(file_path, index=False)
    print(f"Result saved to '{file_path}'.")

def print_separator():
    print("\n" + "=" * 60 + "\n")

def get_user_sequence(sequence_type):
    sequence = input(f"Enter the {sequence_type} sequence: ").upper()
    if all(base in 'ACGTU' for base in sequence):
        return sequence
    else:
        print(f"Invalid {sequence_type} sequence. Please use only A, C, G, T, or U.")
        return None

def main():
    while True:
        print("
        print("
        print("
        print("
        print("
        print("
```



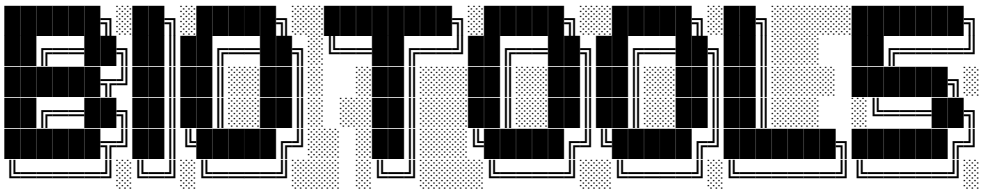
```
        print("\n")

        print("╔═══════════════════════════════════════╗")
        print("║ 1. Enter DNA Sequence Manually        ║")
        print("║ 2. Enter RNA Sequence Manually        ║")
        print("║ 3. Generate Random DNA Sequence       ║")
        print("║ 4. Generate Random RNA Sequence       ║")
        print("║ 5. Read DNA Sequence from CSV         ║")
        print("║ 6. Read RNA Sequence from CSV         ║")
        print("║ 7. Exit                               ║")
        print("╚═══════════════════════════════════════╝")


        main_choice = input("Choose an option to get started.  : ")

        if main_choice == '1':
            random_sequence = get_user_sequence("DNA")
            if random_sequence:
                print("\nDNA Sequence entered successfully.")

        elif main_choice == '2':
            random_sequence = get_user_sequence("RNA")
            if random_sequence:
                print("\n RNA Sequence entered successfully.")

        elif main_choice == '3':
            random_sequence = generate_random_dna_sequence(int(input("\nEnter the length
of the DNA sequence: ")))
            print("\n Random DNA Sequence generated successfully.")

        elif main_choice == '4':
            random_sequence = generate_random_rna_sequence(int(input("Enter the length o
f the RNA sequence: ")))
            print("\n Random RNA Sequence generated successfully.")

        elif main_choice == '5':
            file_path = input("Enter the CSV file path: ")
            dna_sequences = read_csv(file_path)
            if dna_sequences:
                random_sequence = dna_sequences[0]
                print("\n DNA Sequence read from CSV successfully.")

        elif main_choice == '6':
            file_path = input("Enter the CSV file path: ")
            rna_sequences = read_rna_sequence(file_path)
            if rna_sequences:
                random_sequence = rna_sequences[0]
                print("\n RNA Sequence read from CSV successfully.")

        elif main_choice == '5':
            random_sequence = get_user_sequence("DNA")
            if random_sequence:
                print("\n DNA Sequence entered successfully.")

        elif main_choice == '6':
            random_sequence = get_user_sequence("RNA")
            if random_sequence:
                print("\n RNA Sequence entered successfully.")


        elif main_choice == '7':
            print("Exiting the program. Goodbye!")
            print_separator()
            break

        else:
            print("\n Invalid choice. Please enter a number between 1 and 7.")
            print_separator()
            continue

        perform_operations(random_sequence)
```

```
def perform_operations(random_sequence):
    gc_content = 0
    rna_sequence = ""

    while True:
        print_separator()

        print("OPTIONS")
        print("")

        print("\n")

        print("╔═══════════════════════════════════════════╗")
        print("║1. Calculate GC Content                    ║")
        print("║2. Calculate Nucleotide Frequency          ║")
        print("║3. Convert DNA to RNA                      ║")
        print("║4. Convert RNA to DNA                      ║")
        print("║5. Go Back to Main Menu                     ")
        print("╚═══════════════════════════════════════════╝")



        operation_choice = input("□□□□□□ □□□□ □□□□□□ : ")

        if operation_choice == '1':
            gc_content = calculate_gc_content(random_sequence)
            print("\n GC Content calculated successfully:", gc_content)

        elif operation_choice == '2':
            nucleotide_frequency = calculate_nucleotide_frequency(random_sequence)
            print("\n Nucleotide Frequency calculated successfully:")
            for nucleotide, count in nucleotide_frequency.items():
                print(f"{nucleotide}: {count}")

        elif operation_choice == '3':
            rna_sequence = dna_to_rna(random_sequence)
            print("\n DNA to RNA conversion successful. RNA Sequence:", rna_sequence)

        elif operation_choice == '4':
            dna_sequence = rna_to_dna(random_sequence)
            print("\n RNA to DNA conversion successful. DNA Sequence:", dna_sequence)

        elif operation_choice == '5':
            print("\n Returning to the Main Menu.")
            print_separator()
            break

        else:
            print("\n Invalid choice. Please enter a number between 1 and 5.")
            print_separator()
            continue

        download_choice = input("\n Do you want to download results? (y/n): ")
        if download_choice.lower() == 'y':
            result = (random_sequence, gc_content, rna_sequence)
            save_results_to_csv(result, 'results.csv')
            print("\n Results downloaded successfully.")

        more_operations_choice = input("\n Do you want to continue for more operations?
(y/n): ")
        if more_operations_choice.lower() != 'y':
            break

if __name__ == "__main__":
    main()
```

OUTPUT

In [1]:

```
#INSERTING SEQUENCE MANUALLY
```




```
╔═══════════════════════════════════════╗
║ 1. Enter DNA Sequence Manually        ║
║ 2. Enter RNA Sequence Manually        ║
║ 3. Generate Random DNA Sequence       ║
║ 4. Generate Random RNA Sequence       ║
║ 5. Read DNA Sequence from CSV         ║
║ 6. Read RNA Sequence from CSV         ║
║ 7. Exit                               ║
╚═══════════════════════════════════════╝
```

Choose an option to get started. : 1
Enter the DNA sequence: ATGC

DNA Sequence entered successfully.

═══════════════════════════════════════


```
╔═══════════════════════════════════════╗
║1. Calculate GC Content                ║
║2. Calculate Nucleotide Frequency      ║
║3. Convert DNA to RNA                  ║
║4. Convert RNA to DNA                  ║
║5. Go Back to Main Menu                ║
╚═══════════════════════════════════════╝
```

□□□□□ □□□□ □□□□□□ : 1

GC Content calculated successfully: 50.0

Do you want to download results? (y/n): n

Do you want to continue for more operations? (y/n): y

═══════════════════════════════════════


```
╔═══════════════════════════════════════╗
║1. Calculate GC Content                ║
║2. Calculate Nucleotide Frequency      ║
║3. Convert DNA to RNA                  ║
║4. Convert RNA to DNA                  ║
║5. Go Back to Main Menu                ║
╚═══════════════════════════════════════╝
```

□□□□□ □□□□ □□□□□□ : 2

Nucleotide Frequency calculated successfully:
A: 1
C: 1
G: 1
T: 1

Do you want to download results? (y/n): n

Do you want to continue for more operations? (y/n): y

1. Calculate GC Content
2. Calculate Nucleotide Frequency
3. Convert DNA to RNA
4. Convert RNA to DNA
5. Go Back to Main Menu

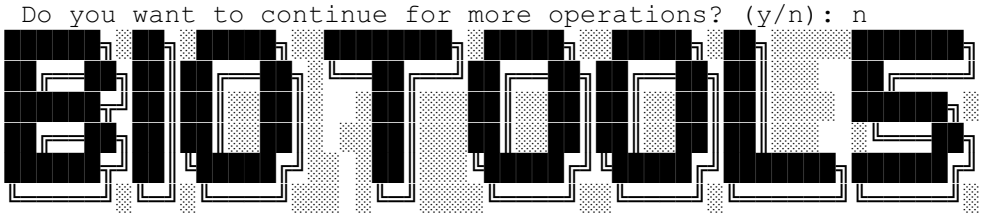
□□□□□ □□□□ □□□□□□ : 3

DNA to RNA conversion successful. RNA Sequence: AUGC

Do you want to download results? (y/n): y
Result saved to 'results.csv'.

Results downloaded successfully.

Do you want to continue for more operations? (y/n): n

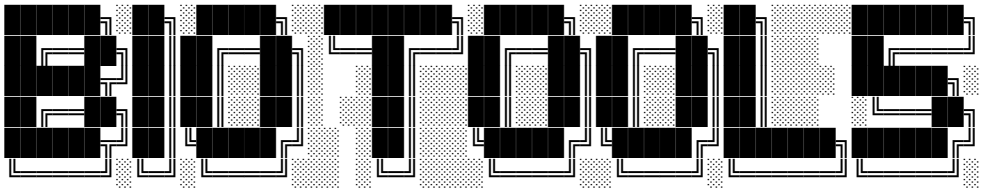


1. Enter DNA Sequence Manually
2. Enter RNA Sequence Manually
3. Generate Random DNA Sequence
4. Generate Random RNA Sequence
5. Read DNA Sequence from CSV
6. Read RNA Sequence from CSV
7. Exit

Choose an option to get started. : 7
Exiting the program. Goodbye!

In [2]:

```
#RANDOM SEQUENCE GENERATION
```



1. Enter DNA Sequence Manually
2. Enter RNA Sequence Manually
3. Generate Random DNA Sequence
4. Generate Random RNA Sequence
5. Read DNA Sequence from CSV
6. Read RNA Sequence from CSV
7. Exit

Choose an option to get started. : 3

```
Enter the length of the DNA sequence: 4

 Random DNA Sequence generated successfully.
```

OPTIONS

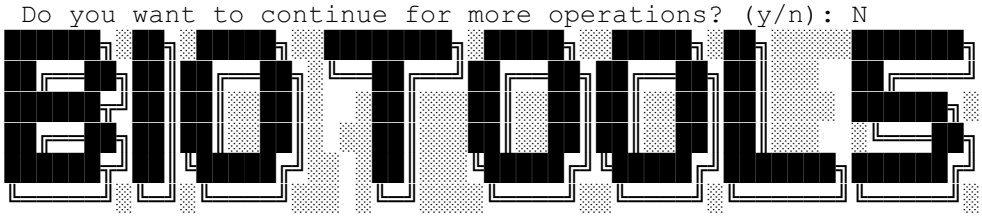
```
1. Calculate GC Content
2. Calculate Nucleotide Frequency
3. Convert DNA to RNA
4. Convert RNA to DNA
5. Go Back to Main Menu
```

```
□□□□□ □□□□ □□□□□□ : 1

 GC Content calculated successfully: 0.0

 Do you want to download results? (y/n): N

 Do you want to continue for more operations? (y/n): Y
```

OPTIONS

```
1. Calculate GC Content
2. Calculate Nucleotide Frequency
3. Convert DNA to RNA
4. Convert RNA to DNA
5. Go Back to Main Menu
```

```
□□□□□ □□□□ □□□□□□ : 3

 DNA to RNA conversion successful. RNA Sequence: UAAA

 Do you want to download results? (y/n): Y
Result saved to 'results.csv'.

 Results downloaded successfully.

 Do you want to continue for more operations? (y/n): N
```

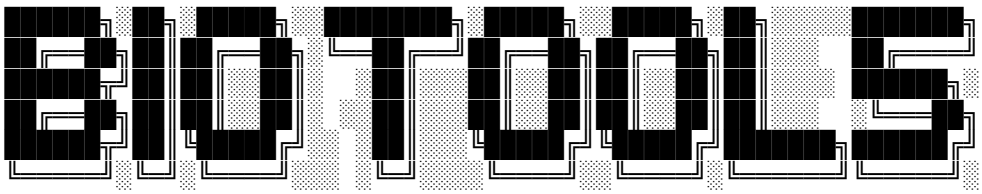



```
 1. Enter DNA Sequence Manually
 2. Enter RNA Sequence Manually
 3. Generate Random DNA Sequence
 4. Generate Random RNA Sequence
 5. Read DNA Sequence from CSV
 6. Read RNA Sequence from CSV
 7. Exit
```

Choose an option to get started. : 7
Exiting the program. Goodbye!

In [3]:

```
#INSERTING THROUGH CSV FILE
```



```
 1. Enter DNA Sequence Manually
 2. Enter RNA Sequence Manually
 3. Generate Random DNA Sequence
 4. Generate Random RNA Sequence
 5. Read DNA Sequence from CSV
 6. Read RNA Sequence from CSV
 7. Exit
```

Choose an option to get started. : 5
Enter the CSV file path: C:\Users\PCL\Desktop\Untitled Folder\results1.csv

DNA Sequence read from CSV successfully.

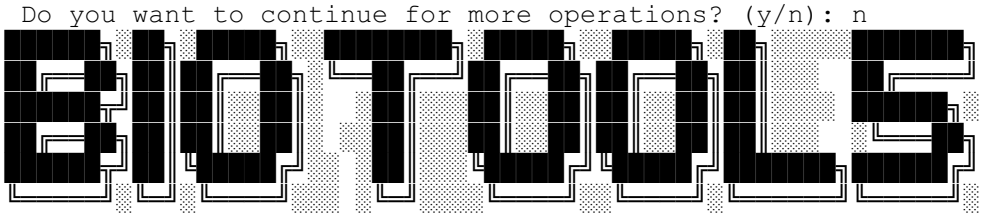
```
1. Calculate GC Content
2. Calculate Nucleotide Frequency
3. Convert DNA to RNA
4. Convert RNA to DNA
5. Go Back to Main Menu
```

□□□□□ □□□□ □□□□□□ : 1

GC Content calculated successfully: 50.0

Do you want to download results? (y/n): n

Do you want to continue for more operations? (y/n): n



```
 1. Enter DNA Sequence Manually
 2. Enter RNA Sequence Manually
 3. Generate Random DNA Sequence
 4. Generate Random RNA Sequence
 5. Read DNA Sequence from CSV
 6. Read RNA Sequence from CSV
 7. Exit
```

Choose an option to get started. : 7
Exiting the program. Goodbye!

MANUAL RESULT

```
DNA Sequence,GC Content,RNA Sequence
ATGC,50.0,AUGC
```

RANDOM RESULT

```
DNA Sequence,GC Content,RNA Sequence
TAAA,0.0,UAAA
```

HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS:

OPERATING SYSTEM	:	WINDOWS 7 AND ABOVE
PROCESSOR	:	PENTIUM(ANY) OR AMD ATHALON(3800+ 4200+ DUAL CORE)
MOTHERBOARD	:	1.845 OR 915,995 FOR PENTIUM 0R MSI K9M VVIA K8M800+8237R PLUS CHIPSET FOR AMD ATHALON
RAM	:	512 MB+

SOFTWARE REQUIREMENTS:

Windows OS

Python

Limitations

1. Needs more customization to fulfill the need of every customer.

2. More functionality can be added as per requirement.

3. No provision to print hard copies.

BIBLIOGRAPHY

- Computer science With Python - Class XII By : Sumita Arora
- Website: https://docs.python.org/3/library/random.html
- Website: https://www.google.com
- Website: https://github.com/belugaop/data_wallah/tree/MAIN/school_src

****