FASTQ Interleaving Assignment Report:

1. Introduction:

The purpose of this assignment is to develop a Python script that interleaves paired-end FASTQ files using Biopython library. Paired-end sequencing generates 2 separate FASTQ files, one for the forward reads (R1) and one for the reverse reads (R2). Interleaving these files ensures that reads from the same DNA fragment appears consecutively in a single output file, which is required by certain bioinformatics Tools.

2. Implementation of the Script:

The script, interleave_fastq.py, performs the following steps:

a. Opens the input FASTQ files (R1 and R2).
b. Reads one record (4 lines) from each file at a time.
c. Writes the record sequentially into interleaved FASTQ file.
d. Handles exceptions, such as different numbers of reads in the output files.

The script makes use of BioPython's Seq10 module.

- Seq10.parse() is used to read FASTQ records.
- Seq10.write() is used to write interleaved records to the output file.

Command to Run the Script:

```
python3 interleave_fastq.py bacterium_R1.fastq bacterium_R2.fastq interleaved.fastq
```

3. Challenges faced and solutions:

a. Missing by a Python library.

Error: ModuleNotFoundError: No module named 'Bio'.

Solution: Installed Biopython using:

```
pip install biopython --user
```

b. GitHub authentication issue.

Issue: GitHub no longer allows password authentication for git push.

Solution: Generated a Personal Access Token (PAT) from GitHub and used it instead of a password.

4. **Verification Steps:**

To ensure that the script functions correctly, we perform the following checks:

a. Checking the Output FASTQ file.

We examined the first 20 lines of the output to verify correct formatting.

```
head -20 interleaved. fastq
```

Correct structure observed (paired reads appear sequentially).

b. Comparing with BBmap's reformat.sh

BBmap's reformat.sh tool was used to generate an interleaved FASTQ file for comparison:

```
reformat.sh in1=bacterium_R1.fastq in2=bacterium_R2.fastq out=temp1.fastq
```

Then, we compared our script's output with BBmap's output using:

```
diff interleaved.fastq temp1.fastq
```

No difference found, confirming that our script correctly interleaves the reads.

5. **Comparison with BBmap's** reformat.sh

1. Difference in implementation.
    - BBmap's reformat.sh is a compiled Java program optimized for large datasets.
    - Our Python script is a lightweight solution using BioPython.
2. Performance.
    - BBmap processes FASTQ files faster because it is optimized for high-performance computing.
    - Python is slower for larger files but is more readable and easier to modify.

Both methods produce the same interleaved output.

6. GitHub Repository:

The script has been uploaded to GitHub for version control.

GitHub Repository URL

https://github.com/Hemalatha18-bio/BCB5250_FASTQ_Interleaving

7. Conclusion:

This assignment provides hands on experience with NGC data handling, BioPython, and GitHub version control. The Python script successfully interleaves paired-end FASTQ files, producing an output that is identical to BBmap's reformat.sh. The script is flexible, lightweight, and easy to integrate into future workflows.