**Description of motifs.sh, gibbs16.cpp, PSSM\_MOTIF\_FINDER.py and logo.py**

**motifs.sh:**

This file contains a script that will run a Gibbs Sampler c++ script, a PSSM python script, and a Sequence Logo script. The script is show below.

#!/bin/bash

# Compile and run the C++ program

g++ gibbs16.cpp -o gb

./gb E.coliRpoN-sequences-16-100nt.fasta 22

# Run the first Python program

python3 PSSM\_MOTIF\_FINDER.py FruR.txt ecoK12-MG1655.fasta 18

# Run the second Python program

python3 logos.py

**gibbs16.cpp:**

./gb [fasta\_sequence\_chunks\_here] [starting\_motif\_length]

The program gibbs16.cpp implements a Gibbs Sampler algorithm. It requires a file containing segments from a .fasta file and a starting motif length (e.g., 22 bases). The program aims to identify the optimal motif position and length within each provided .fasta chunk. It iteratively scans each segment to optimize motif positions and adjust sizes for the most suitable motif length. After iterations, the program provides statistics regarding the best motif length, position, and overall score. The resulting ideal motif sequences are then saved to a .txt file for further analysis, including the creation of a sequence logo.

**PSSM\_MOTIF\_FINDER.py:**

[.txt file containing a certain amount of motifs] [a backfround sequence file] [minimum score]

The program PSSM\_MOTIF\_FINDER.py utilizes a Position-Specific Scoring Matrix (PSSM) function. It requires a .txt file containing motifs with similar sequences and a .fasta file containing nucleotide bases. The program searches through the background file to find sequences resembling the provided motifs. Users can input a minimum score threshold (e.g., 18), and the program identifies and saves all motif sequences surpassing this score. The resulting common motif sequences are stored in a .txt file, ready for further analysis, including the creation of a sequence logo.

**logo.py:**

The script logo.py, written in Python, generates a sequence logo for the aligned DNA sequences produced by the PSSM\_MOTIF\_FINDER.py and gibbs16.cpp scripts. It utilizes base counts files generated from both programs to create the logo. The sequence logo provides visual representation and insight into the conserved regions identified by the PSSM and Gibbs Sampler algorithms.

To Create the logos, use logomaker module and follow respective syntax to create a sequence logo. Installation for logomaker is show below:

pip install logomaker

**Gibbs Sampler Sequence Logo:**

A colorful letters on a white background

Description automatically generated

**PSSM sequence LOGO:**

A colorful text with letters

Description automatically generated with medium confidence