CSE 4065 –Computational Genomics Programming Assignment #1

Project Report

Group Members:
Cem Güleç: 150117828
Büşra Gökmen: 150116027
Ömer Faruk Çakı: 150117821

Introduction:

First of all, in order to generate random DNA sequences we write a very basic `generate_sequence(k, bases='ACGT')` function. This function takes the length and the alphabet and creates the random sequences. Also we have another function called `generate_mutated_sequence()` which is used to randomly place a mutated string to each sequence. After the random DNA sequences are generated and mutated k-mers are randomly placed, we have saved the output in the `"input.txt"`.

I. Randomized Motif Search:

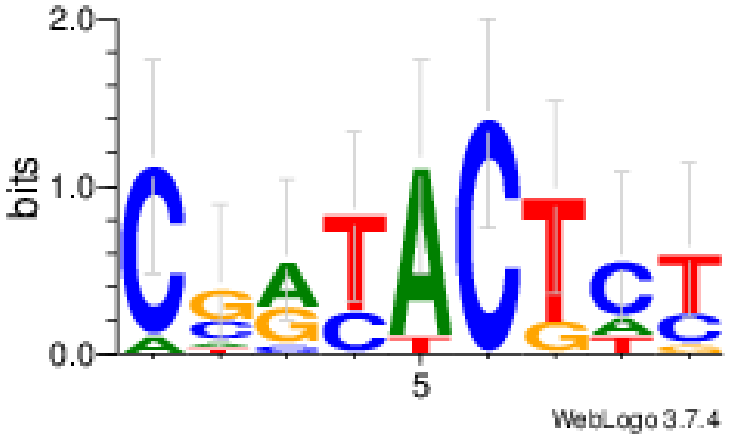
In the program's main, we call the `randomized_motif_search(k, file)` function with the arguments of k and the name of the input file.

In the `randomized_motif_search()` function, first of all we read the 10 instances of DNA strings of length 500 from the provided input file. After that, by creating random index numbers we choose random motifs from each instance. In order to determine the consensus string of selected motifs we call the `generate_consensus(motif_list)` function and it returns the string with the highest probability. After we get our consensus string this time we call the `calculate_score(motif_list, consensus_str)` to calculate the score and we print some useful information on each iteration. In the following for loop we are counting the pairs and calculating the profile matrix based on counts. Using the *Profile(Motifs)*, we calculate probabilities of each possible k-mers of the DNA sequences and determine new motifs.
At the end of all these in each iteration, if the score is reduced we keep continuing with the new motifs, until it does not reduce the score anymore.

Consensus strings acquired using different k values:

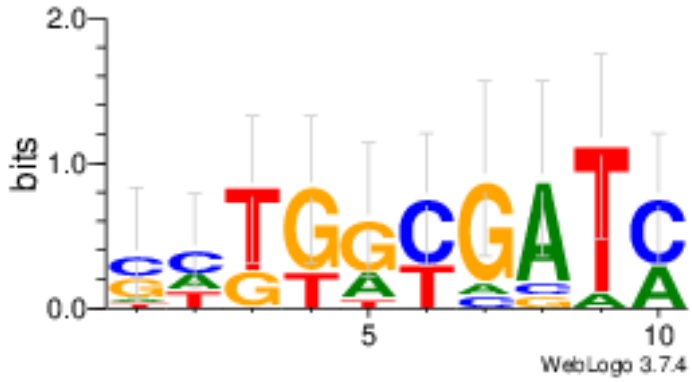
k = 9: *CGATACTCT*

```
PS C:\Users\Cem\Documents\GitHub\Genomics-Project1> python randomized.py
consensus string:  CAATACTCT
Iteration number: 1
consensus string: CAATACTCT

Motif List:
['A', 'A', 'C', 'C', 'C', 'A', 'G', 'A', 'A']
['A', 'G', 'A', 'C', 'A', 'C', 'G', 'C', 'T']
['A', 'C', 'A', 'T', 'T', 'C', 'A', 'C', 'G']
['C', 'T', 'G', 'A', 'A', 'G', 'G', 'A', 'T']
['C', 'A', 'C', 'C', 'A', 'C', 'T', 'C', 'T']
['C', 'A', 'G', 'T', 'G', 'T', 'A', 'T', 'C']
['T', 'G', 'G', 'G', 'G', 'T', 'T', 'T', 'G']
['C', 'T', 'T', 'G', 'T', 'C', 'T', 'A', 'C']
['T', 'C', 'A', 'G', 'A', 'G', 'C', 'G', 'T']
['G', 'C', 'A', 'T', 'G', 'C', 'T', 'C', 'C']

initial score: 55

**************************************************
Iteration number:  1

New motif list: ['CTATACGTT', 'AGACACGCT', 'CGGCACTAT', 'CCATACTTT', 'CACCACTCT', 'CCATACTCT', 'CGGTACTCC', 'CGATACTCC', 'CCGTTCTAC',
 'CGGTACTCG']

new score:
 25
Profile matrix:
A: 0.3 | 0.3 | 0.4 | 0.1 | 0.4 | 0.1 | 0.2 | 0.3 | 0.1 |
T: 0.2 | 0.2 | 0.1 | 0.3 | 0.2 | 0.2 | 0.4 | 0.2 | 0.4 |
G: 0.1 | 0.2 | 0.3 | 0.3 | 0.3 | 0.2 | 0.3 | 0.1 | 0.2 |
C: 0.4 | 0.3 | 0.2 | 0.3 | 0.1 | 0.5 | 0.1 | 0.4 | 0.3 |

initial score was: 55
best score possible: 25
stopped at score: 25
number of iterations passed: 2
Runtime of Randomized Motif Search: 0.1850873 sec
```



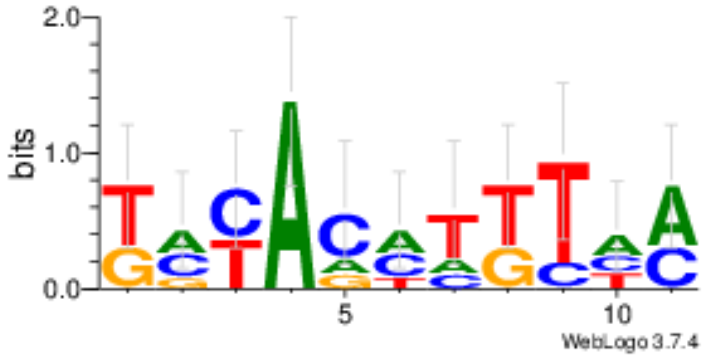
k = 10: *CCTGGCGATC*

```
PS C:\Users\Cem\Documents\GitHub\Genomics-Project1> python randomized.py
consensus string:  GCTGGTGATA
Iteration number: 1
consensus string: GCTGGTGATA

Motif List:
['A', 'T', 'T', 'G', 'G', 'T', 'G', 'C', 'A', 'A']
['T', 'C', 'T', 'T', 'C', 'G', 'C', 'C', 'T', 'C']
['C', 'C', 'T', 'G', 'G', 'C', 'G', 'G', 'T', 'A']
['C', 'A', 'T', 'T', 'T', 'C', 'G', 'A', 'A', 'A']
['G', 'C', 'G', 'A', 'T', 'T', 'T', 'A', 'T', 'G']
['C', 'A', 'C', 'T', 'A', 'C', 'G', 'T', 'C', 'C']
['A', 'T', 'G', 'G', 'G', 'C', 'G', 'G', 'T', 'C']
['G', 'C', 'G', 'G', 'C', 'T', 'C', 'C', 'C', 'G']
['G', 'T', 'A', 'G', 'G', 'G', 'A', 'A', 'A', 'C']
['T', 'A', 'C', 'G', 'A', 'T', 'A', 'A', 'G', 'A']

initial score: 58

**************************************************
Iteration number:  1

New motif list: ['GAGTGTGATC', 'GCTGGTGATA', 'CCTGGCGGTA', 'GATGATGATC', 'CTTGACAATA', 'AATGGCGAAC', 'GCGGTCGCTC', 'CTGGGTCATC', 'CTTT
GCGATC', 'TCTTACGATA']

new score:
 35
Profile matrix:
A: 0.2 | 0.3 | 0.1 | 0.1 | 0.2 | 0.0 | 0.2 | 0.4 | 0.3 | 0.4 |
T: 0.2 | 0.3 | 0.4 | 0.3 | 0.2 | 0.4 | 0.1 | 0.1 | 0.4 | 0.0 |
G: 0.3 | 0.0 | 0.3 | 0.6 | 0.4 | 0.2 | 0.5 | 0.2 | 0.1 | 0.2 |
C: 0.3 | 0.4 | 0.2 | 0.0 | 0.2 | 0.4 | 0.2 | 0.3 | 0.2 | 0.4 |

initial score was: 58
best score possible: 35
stopped at score: 36
number of iterations passed: 2
Runtime of Randomized Motif Search: 0.1635014 sec
```



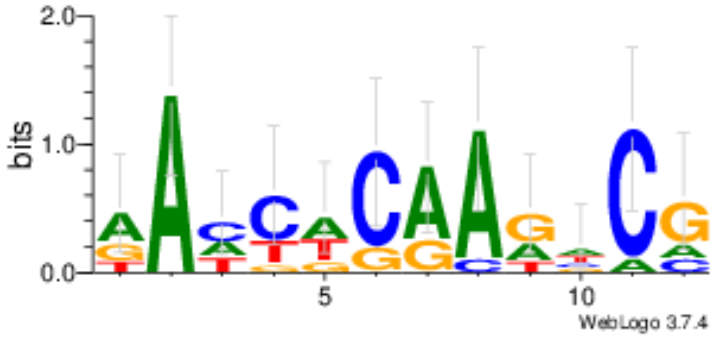
k = 11: *TACACATTTAA*

```
PS C:\Users\Cem\Documents\GitHub\Genomics-Project1> python randomized.py
consensus string:  TATACAATTTA
Iteration number: 1
consensus string: TATACAATTTA

Motif List:
['T', 'A', 'T', 'A', 'C', 'G', 'T', 'T', 'T', 'C', 'A']
['G', 'C', 'A', 'T', 'C', 'T', 'A', 'T', 'C', 'T', 'C']
['G', 'T', 'C', 'A', 'G', 'C', 'C', 'T', 'A', 'C', 'A']
['C', 'G', 'A', 'A', 'A', 'G', 'T', 'A', 'C', 'C', 'A']
['A', 'A', 'G', 'A', 'C', 'C', 'A', 'A', 'T', 'T', 'T']
['T', 'T', 'T', 'C', 'A', 'A', 'T', 'G', 'G', 'T', 'A']
['G', 'G', 'C', 'G', 'G', 'T', 'C', 'G', 'C', 'T', 'C']
['T', 'C', 'C', 'G', 'C', 'A', 'A', 'C', 'T', 'A', 'T']
['T', 'C', 'T', 'A', 'T', 'C', 'C', 'T', 'T', 'A', 'A']
['G', 'A', 'T', 'C', 'A', 'A', 'G', 'G', 'C', 'A', 'C']

initial score: 67

**************************************************
Iteration number:  1

New motif list: ['GATACAATTCA', 'TACAACTTCAC', 'TCTACTATTCA', 'TCCAATTTTTA', 'GACAGCTTTTA', 'GGTAGATGTTA', 'TGTACCCGTAA', 'TATACATTTCC
', 'GCCACCTGTAC', 'TCCACACGCAC']

new score:
 45
Profile matrix:
A: 0.1 | 0.3 | 0.2 | 0.5 | 0.3 | 0.3 | 0.3 | 0.2 | 0.1 | 0.3 | 0.5 |
T: 0.4 | 0.2 | 0.4 | 0.1 | 0.1 | 0.2 | 0.3 | 0.4 | 0.4 | 0.4 | 0.2 |
G: 0.4 | 0.2 | 0.1 | 0.2 | 0.2 | 0.2 | 0.1 | 0.3 | 0.1 | 0.0 | 0.0 |
C: 0.1 | 0.3 | 0.3 | 0.2 | 0.4 | 0.3 | 0.3 | 0.1 | 0.4 | 0.3 | 0.3 |

initial score was: 67
best score possible: 45
stopped at score: 45
number of iterations passed: 2
Runtime of Randomized Motif Search: 0.1686849 sec
```



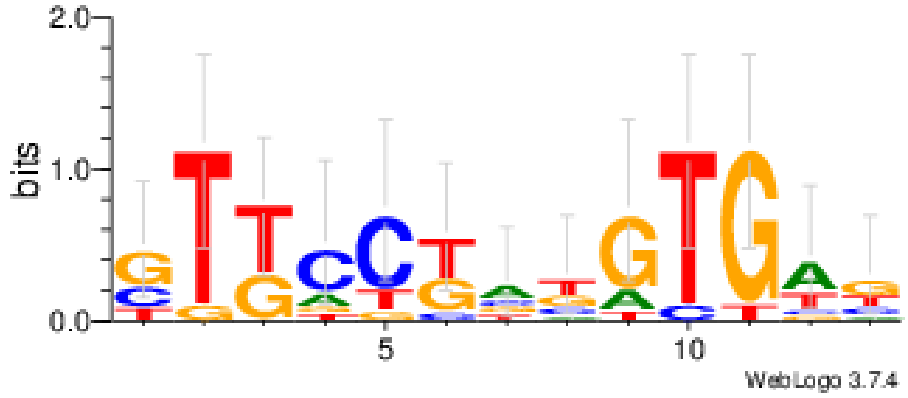
k = 12: *AACCACAAGACG*

```
PS C:\Users\Cem\Documents\GitHub\Genomics-Project1> python randomized.py
consensus string:  AATTGCAAGGCG
Iteration number: 1
consensus string: AATTGCAAGGCG

Motif List:
['C', 'T', 'T', 'T', 'G', 'G', 'T', 'A', 'G', 'C', 'C', 'T']
['A', 'A', 'G', 'C', 'T', 'T', 'G', 'A', 'C', 'T', 'C', 'C']
['G', 'A', 'C', 'T', 'T', 'C', 'G', 'A', 'T', 'G', 'T', 'A']
['A', 'A', 'T', 'G', 'G', 'C', 'G', 'C', 'A', 'G', 'A', 'G']
['G', 'G', 'T', 'T', 'G', 'A', 'C', 'G', 'G', 'A', 'C', 'A']
['T', 'A', 'A', 'C', 'A', 'C', 'A', 'A', 'A', 'G', 'G', 'G']
['A', 'A', 'A', 'T', 'A', 'G', 'A', 'C', 'G', 'T', 'T', 'G']
['T', 'C', 'C', 'T', 'T', 'G', 'T', 'A', 'A', 'G', 'C', 'C']
['A', 'A', 'G', 'C', 'G', 'C', 'A', 'A', 'T', 'C', 'C', 'C']
['T', 'A', 'C', 'G', 'A', 'T', 'A', 'A', 'G', 'A', 'A', 'G']

initial score: 65

**************************************************
Iteration number:  1

New motif list: ['AACCGGGAGCCC', 'TAATACAAAACG', 'GAATGCAAATCA', 'GATCTCGAATCG', 'GACTTCAATGCG', 'AACCACAAGGCA', 'AATCAGACGACG', 'TATG
TCGAGACG', 'AAACTCAAGTCC', 'AACCACAATCAG']

new score:
 44
Profile matrix:
A: 0.4 | 0.7 | 0.2 | 0.0 | 0.3 | 0.1 | 0.4 | 0.7 | 0.3 | 0.2 | 0.2 | 0.2 |
T: 0.3 | 0.1 | 0.3 | 0.5 | 0.3 | 0.2 | 0.2 | 0.0 | 0.2 | 0.2 | 0.2 | 0.1 |
G: 0.2 | 0.1 | 0.2 | 0.2 | 0.4 | 0.3 | 0.3 | 0.1 | 0.4 | 0.4 | 0.1 | 0.4 |
C: 0.1 | 0.1 | 0.3 | 0.3 | 0.0 | 0.4 | 0.1 | 0.2 | 0.1 | 0.2 | 0.5 | 0.3 |

initial score was: 65
best score possible: 44
stopped at score: 44
number of iterations passed: 2
Runtime of Randomized Motif Search: 0.1582083 sec
```



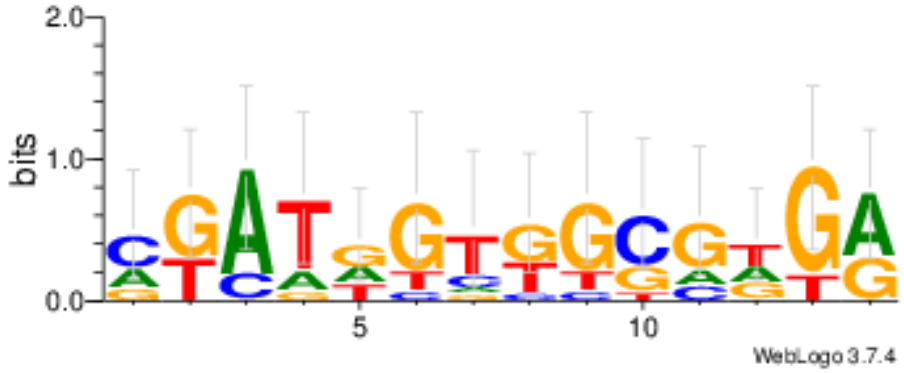
k = 13: *GTTCCTATGTGAG*

```
PS C:\Users\Cem\Documents\GitHub\Genomics-Project1> python randomized.py
consensus string:  GTGGCGATGTGAT
Iteration number: 1
consensus string: GTGGCGATGTGAT

Motif List:
['C', 'T', 'T', 'T', 'G', 'G', 'T', 'A', 'G', 'C', 'C', 'T', 'C']
['C', 'G', 'C', 'C', 'T', 'G', 'A', 'C', 'G', 'T', 'G', 'A', 'G']
['T', 'T', 'T', 'G', 'C', 'G', 'G', 'G', 'C', 'A', 'G', 'T', 'A']
['G', 'T', 'G', 'T', 'T', 'C', 'C', 'T', 'A', 'T', 'T', 'G', 'T']
['G', 'A', 'G', 'G', 'C', 'A', 'T', 'C', 'G', 'C', 'G', 'C', 'A']
['G', 'T', 'C', 'C', 'T', 'T', 'A', 'T', 'A', 'T', 'G', 'C', 'G']
['A', 'G', 'G', 'G', 'C', 'G', 'C', 'T', 'T', 'C', 'T', 'A', 'T']
['G', 'A', 'T', 'A', 'C', 'T', 'A', 'G', 'G', 'G', 'G', 'G', 'G']
['T', 'G', 'G', 'C', 'C', 'C', 'G', 'A', 'A', 'T', 'T', 'T', 'C']
['T', 'T', 'G', 'A', 'G', 'T', 'A', 'G', 'T', 'T', 'G', 'A', 'T']

initial score: 76

**************************************************
Iteration number:  1

New motif list: ['TTTCTGACGTGTG', 'GTGAGCTGGTGAT', 'CTTCCTCAATGGT', 'GGTCCGATGTGAG', 'CTTCCGTGGTTTG', 'GTGACTCTTTGTT', 'GTGTTTACGTGAG'
, 'TTTCCTAGACGAC', 'CTGGCTGTGTGAC', 'GTTCCGGTGTGCA']

new score:
 50
Profile matrix:
A: 0.1 | 0.2 | 0.0 | 0.2 | 0.0 | 0.1 | 0.4 | 0.2 | 0.3 | 0.1 | 0.0 | 0.3 | 0.2 |
T: 0.3 | 0.5 | 0.3 | 0.2 | 0.3 | 0.3 | 0.2 | 0.3 | 0.2 | 0.5 | 0.3 | 0.3 | 0.3 |
G: 0.4 | 0.3 | 0.5 | 0.3 | 0.2 | 0.4 | 0.2 | 0.3 | 0.4 | 0.1 | 0.6 | 0.2 | 0.3 |
C: 0.2 | 0.0 | 0.2 | 0.3 | 0.5 | 0.2 | 0.2 | 0.2 | 0.1 | 0.3 | 0.1 | 0.2 | 0.2 |

initial score was: 76
best score possible: 50
stopped at score: 50
number of iterations passed: 2
Runtime of Randomized Motif Search: 0.1681565 sec
```



k = 14: ***CGATGGTGGCGTGA***

```
initial score: 86

**************************************************
Iteration number:  1

New motif list: ['AGAATGGTTCGTGG', 'GTGAGCTGGTGATA', 'GTATTGTCGCCAGA', 'CGAAAGTACCAATG', 'ATAATATTCGAGGC', 'ATCTGCTTTGGATA', 'CGCTGGTG
GCGAGA', 'CGAGGGTGGCGTGG', 'AGATAGATGGGGGG', 'GGATTTCGGCAGGA']

new score:
 62
Profile matrix:
A: 0.3 | 0.1 | 0.3 | 0.4 | 0.2 | 0.2 | 0.1 | 0.3 | 0.0 | 0.1 | 0.3 | 0.4 | 0.1 | 0.3 |
T: 0.1 | 0.4 | 0.1 | 0.3 | 0.5 | 0.2 | 0.4 | 0.3 | 0.3 | 0.2 | 0.1 | 0.2 | 0.3 | 0.0 |
G: 0.3 | 0.4 | 0.3 | 0.3 | 0.3 | 0.4 | 0.2 | 0.3 | 0.4 | 0.3 | 0.4 | 0.3 | 0.4 | 0.4 |
C: 0.3 | 0.1 | 0.3 | 0.0 | 0.0 | 0.2 | 0.3 | 0.1 | 0.3 | 0.4 | 0.2 | 0.1 | 0.2 | 0.3 |

**************************************************
Iteration number:  2

New motif list: ['AGAATGGTTCGTGG', 'CTAAAGTGCTGTGA', 'GTATTGTCGCCAGA', 'CGATGTTTGCATTG', 'CTATAGCGGGCGGA', 'ATCTGCTTTGGATA', 'CGCTGGTG
GCGAGA', 'CGAGGGTGGCGTGG', 'AGATAGATGGGGGG', 'GGATTTCGGCAGGA']

new score:
 55
Profile matrix:
A: 0.4 | 0.0 | 0.7 | 0.4 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 | 0.0 | 0.3 | 0.5 | 0.0 | 0.5 |
T: 0.0 | 0.4 | 0.0 | 0.5 | 0.4 | 0.1 | 0.7 | 0.4 | 0.2 | 0.1 | 0.0 | 0.2 | 0.3 | 0.0 |
G: 0.3 | 0.6 | 0.1 | 0.1 | 0.4 | 0.6 | 0.1 | 0.4 | 0.6 | 0.3 | 0.6 | 0.3 | 0.7 | 0.4 |
C: 0.3 | 0.0 | 0.2 | 0.0 | 0.0 | 0.2 | 0.1 | 0.1 | 0.2 | 0.6 | 0.1 | 0.0 | 0.0 | 0.1 |

initial score was: 86
best score possible: 55
stopped at score: 55
number of iterations passed: 3
Runtime of Randomized Motif Search: 0.2333518 sec
```



II. Gibbs Sampler:

In the program's main, we call the `Gibbs_Sampler(k, file)` function with the arguments of k and the name of the input file.

In the `Gibbs_Sampler()` function, first of all we read the 10 instances of DNA strings of length 500 from the provided input file. After that, by creating random index numbers we choose random motifs from each instance. Then we randomly removed one of the motifs from this motifs list. In order to determine the consensus string of selected motifs we call the `generate_consensus(motif_list)` function and it returns the string with the highest probability.

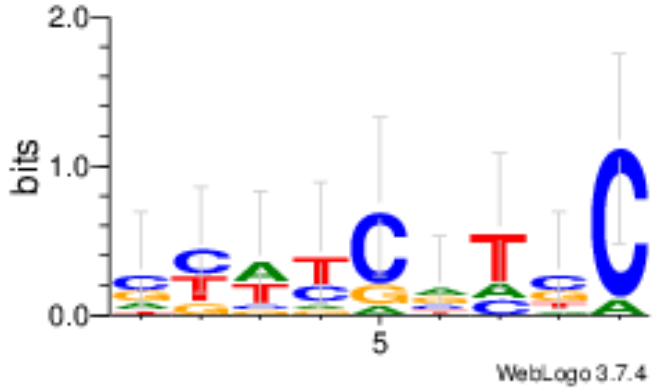
After we get our consensus string this time we call the `calculate_score(motif_list, consensus_str)` to calculate the score and we print some useful information on each iteration. We determine best score by initial motif list before the iterations. In the following while loop we remove one the motifs from the motif list and calculate the profile matrix based on counts. Using the *Profile(Motifs)*, we calculate probabilities of each possible k-mers of the DNA sequences and determine new motifs. We take the full sequence from the DNA list that has the same index with removed from motif list. We create new substrings that have k length from this sequence and calculate probability of them with `calculate_single_prob(profile_list, variation)` function. Then we use `die(list)` function to choose randomly k-mer from these substrings by a biased die. We add a randomly chosen k-mer to motif_list instead of removing the k-mer motif's position.

At the end of all these in each iteration, if the score is reduced we keep continuing with the new motifs. We control while loop iterations number with a threshold value that is equal to 500 for 1000 iterations. If the best score is the same through 500 iterations then we stop the while loop without 1000 iterations.

Consensus strings

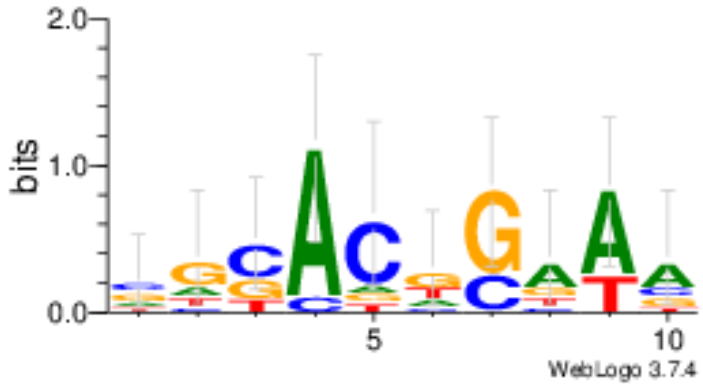
k = 9: *CCATCATCC*

```
new score:  45
best score: 33
33
consensus string:  CTATCATCC
Final Motif List: ['CCACGACGC', 'ACAACTTCA', 'AGACCATCT', 'GCTTGAATC', 'ATTTCAAGC', 'GTCTCGTTC', 'CTTCCTTCC', 'CCACGGTTC', 'CTGTACAGC'
, 'GGTTCGCCC']
best score initial: 53
new score:  46
best score: 33
33
consensus string:  CTATCATCC
Final Motif List: ['CCACGACGC', 'ACAACTTCA', 'AGACCATCT', 'GGTTCATCC', 'ATTTCAAGC', 'GTCTCGTTC', 'CTTCCTTCC', 'CCACGGTTC', 'CTGTACAGC'
, 'GGTTCGCCC']
best score initial: 53
new score:  43
best score: 33
33
consensus string:  CTATCATCC
Final Motif List: ['CCACGACGC', 'ACAACTTCA', 'TCAGCCTAC', 'GGTTCATCC', 'ATTTCAAGC', 'GTCTCGTTC', 'CTTCCTTCC', 'CCACGGTTC', 'CTGTACAGC'
, 'GGTTCGCCC']
best score initial: 53
new score:  44
best score: 33
33
number of iterations: 511
Runtime of Gibbs Sampler Search: 4.0451891 sec
```



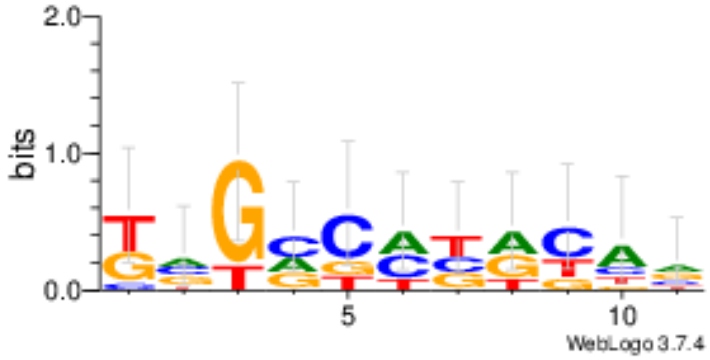
k = 10: *CGCACGGAAA*

```
consensus string:  GGCACTGAAA
Final Motif List: ['CAGGCTGAGT', 'AGGACAGGTA', 'CGCACTGATA', 'GACACGCCAA', 'GGGAGTTAAG', 'TTCAATGGTA', 'GGCCCTGTAC', 'ACCACCGAAG', 'CG
TAGGGAAA', 'TGGATGCAAT']
best score initial: 59
new score:  43
best score: 39
39
consensus string:  GGCACTGAAA
Final Motif List: ['GTTACGCTAC', 'AGGACAGGTA', 'CGCACTGATA', 'GACACGCCAA', 'GGGAGTTAAG', 'TTCAATGGTA', 'GGCCCTGTAC', 'ACCACCGAAG', 'CG
TAGGGAAA', 'TGGATGCAAT']
best score initial: 59
new score:  43
best score: 39
39
consensus string:  GGCACGGAAA
Final Motif List: ['GTTACGCTAC', 'AGGACAGGTA', 'CGCACTGATA', 'GACACGCCAA', 'CAGACAGAAG', 'TTCAATGGTA', 'GGCCCTGTAC', 'ACCACCGAAG', 'CG
TAGGGAAA', 'TGGATGCAAT']
best score initial: 59
new score:  43
best score: 39
39
number of iterations: 509
Runtime of Gibbs Sampler Search: 3.0093202 sec
```



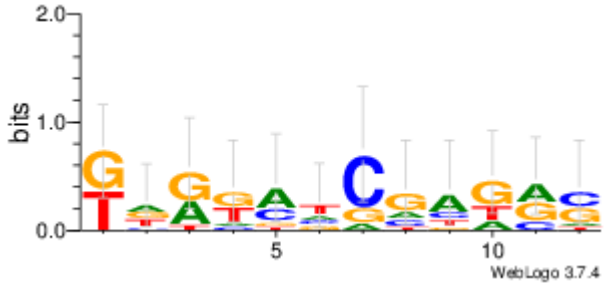
k = 11: *TAGCCATACAA*

```
consensus string:  GCGACATACAA
Final Motif List: ['TTGGTCCGCAC', 'TCGGGCCATAG', 'TCGCCAGACCA', 'TGGACATTCCA', 'GATACATACGC', 'GAGCGTGGTAA', 'GCGAGAAGCGT', 'CCGCCACGG
TT', 'GGGCCCGAGTT', 'GCTACTTTGCG']
best score initial: 65
new score:  59
best score: 48
48
consensus string:  TCGACATACAA
Final Motif List: ['TTGGTCCGCAC', 'TCGGGCCATAG', 'TCGCCAGACCA', 'TGGACATTCCA', 'GATACATACGC', 'GAGCGTGGTAA', 'GCGAGAAGCGT', 'CCGCCACGG
TT', 'GGGCCCGAGTT', 'TATACCTTTAG']
best score initial: 65
new score:  59
best score: 48
48
consensus string:  TAGCCATACAA
Final Motif List: ['TTGGTCCGCAC', 'TCGGGCCATAG', 'TCGCCAGACCA', 'TGGACATTCCA', 'GATACATACGC', 'GAGCGTGGTAA', 'GGGGTTTGCAG', 'CCGCCACGG
TT', 'GGGCCCGAGTT', 'TATACCTTTAG']
best score initial: 65
new score:  59
best score: 48
48
number of iterations: 506
Runtime of Gibbs Sampler Search: 3.947945 sec
```



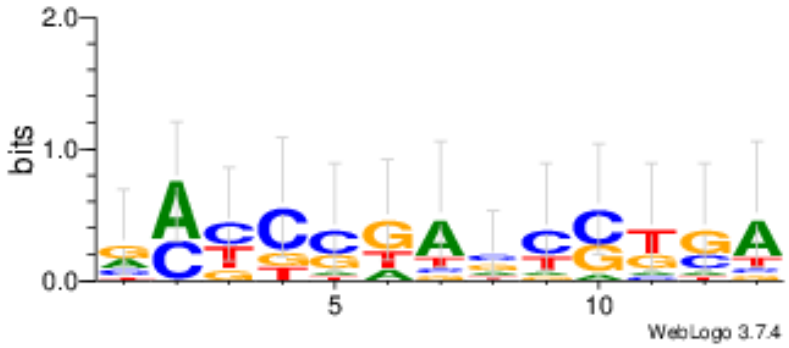
k = 12: *GAGGATCGAGAC*

```
consensus string:  TAGTATCGAGGG
Final Motif List: ['GCGTATCGGAGC', 'TGAGGCCGAGAC', 'TTGTCGGCCGCC', 'TGAAAGCAATGG', 'GGGCCCCGGAGG', 'TTGGATACCGCG', 'GCGTTACGTTGC', 'TA
AGCTCCAGAC', 'TAAGATAGATGG', 'GATTAACTCAAG']
best score initial: 64
new score:  63
best score: 46
46
consensus string:  TAGTATCGAGGC
Final Motif List: ['GCGTATCGGAGC', 'TGAGGCCGAGAC', 'TTGTCGGCCGCC', 'TGAAAGCAATGG', 'GGGCCCCGGAGG', 'GTGTATCATGAA', 'GCGTTACGTTGC', 'TA
AGCTCCAGAC', 'TAAGATAGATGG', 'GATTAACTCAAG']
best score initial: 64
new score:  62
best score: 46
46
consensus string:  TAGTATCGAGAG
Final Motif List: ['GTGGATGGAGCT', 'TGAGGCCGAGAC', 'TTGTCGGCCGCC', 'TGAAAGCAATGG', 'GGGCCCCGGAGG', 'GTGTATCATGAA', 'GCGTTACGTTGC', 'TA
AGCTCCAGAC', 'TAAGATAGATGG', 'GATTAACTCAAG']
best score initial: 64
new score:  64
best score: 46
46
number of iterations: 507
Runtime of Gibbs Sampler Search: 4.320262 sec
```



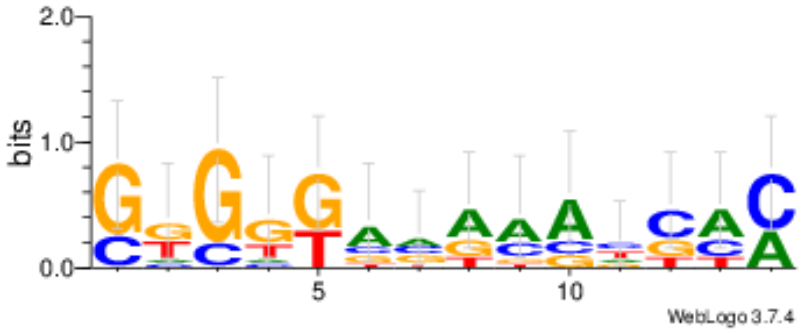
k = 13: *GACCCGACCCTGA*

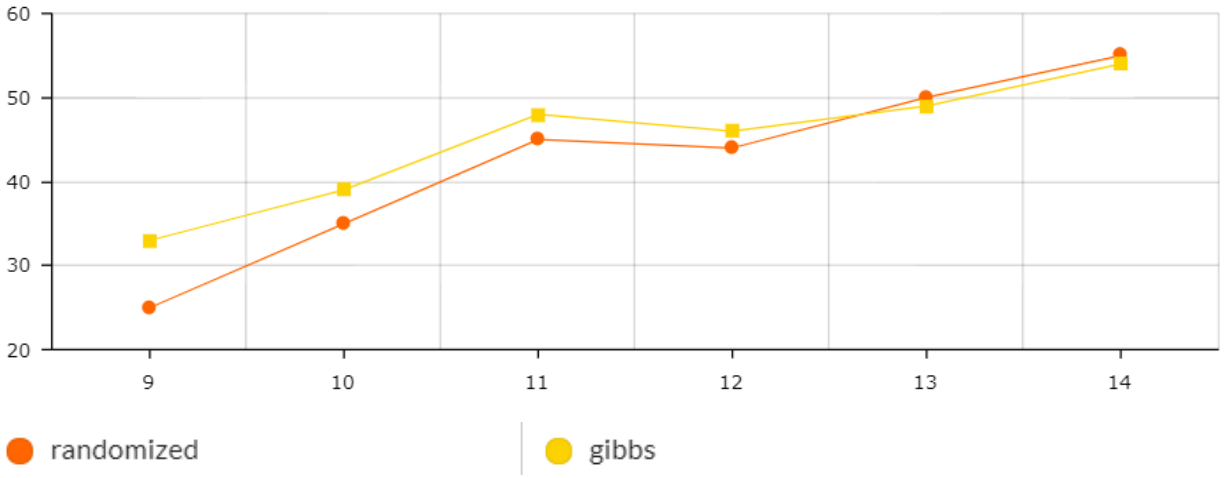
```
consensus string:  GATCCGATCCTGA
Final Motif List: ['GCTGCGTTTCTGA', 'ACTCCGATCCTCA', 'CCTCGTGACCTGG', 'AACGCATGCGGAA', 'GACTTGTCTGCTT', 'GCTCCTAGGCTCA', 'GCCTAGACCCGG
A', 'CAGCGAATTGTCT', 'GAGCGTAAAATGC', 'AATCCGAGCCGGA']
best score initial: 74
new score:  62
best score: 49
49
consensus string:  GATCCGATCCTGA
Final Motif List: ['GCTGCGTTTCTGA', 'ACTCCGATCCTCA', 'CCTCGTGACCTGG', 'AACGCATGCGGAA', 'GACTTGTCTGCTT', 'GCTCCTAGGCTCA', 'GCCTAGACCCGG
A', 'CAGCGAATTGTCT', 'GAGCGTAAAATGC', 'AATCCGAGCCGGA']
best score initial: 74
new score:  62
best score: 49
49
consensus string:  GATCCGAGCCTGA
Final Motif List: ['TACGCGCCTGAGA', 'ACTCCGATCCTCA', 'CCTCGTGACCTGG', 'AACGCATGCGGAA', 'GACTTGTCTGCTT', 'GCTCCTAGGCTCA', 'GCCTAGACCCGG
A', 'CAGCGAATTGTCT', 'GAGCGTAAAATGC', 'AATCCGAGCCGGA']
best score initial: 74
new score:  65
best score: 49
49
number of iterations: 513
Runtime of Gibbs Sampler Search: 3.1961971 sec
```



k = 14: *GGGGGAAAAACCAC*

```
consensus string:  GTGGTAGAAAGCAA
Final Motif List: ['CGGGAGCCCGGCAA', 'CTGGTGAGACCGCA', 'GCCGTACATCTCAC', 'GTGTGACACGCCAA', 'GGGAGGGGAGTTAA', 'GTGCGCGTAAGGTA', 'GGGGTT
TGCAGGAC', 'GGGGTAGAAACCAC', 'CAGTGAAAAATCTC', 'ATGGATTTCGGCAG']
best score initial: 76
new score:  69
best score: 54
54
consensus string:  GTGGGAGAAATCAC
Final Motif List: ['CGCGGACTCAACCC', 'CTGGTGAGACCGCA', 'GCCGTACATCTCAC', 'GTGTGACACGCCAA', 'GGGAGGGGAGTTAA', 'GTGCGCGTAAGGTA', 'GGGGTT
TGCAGGAC', 'GGGGTAGAAACCAC', 'CAGTGAAAAATCTC', 'ATGGATTTCGGCAG']
best score initial: 76
new score:  69
best score: 54
54
consensus string:  GTGGGAAAAATCAC
Final Motif List: ['CGCGGACTCAACCC', 'CTGGTGAGACCGCA', 'GCCGTACATCTCAC', 'GTGTGACACGCCAA', 'GGGAGGGGAGTTAA', 'GTGCGCGTAAGGTA', 'GGGGTT
TGCAGGAC', 'GGGGTAGAAACCAC', 'CAGTGAAAAATCTC', 'GTGTGCAAGAATCC']
best score initial: 76
new score:  67
best score: 54
54
number of iterations: 511
Runtime of Gibbs Sampler Search: 6.2816871999999995 sec
```





Graph-1: Stacked line representation for score considering both algorithms.

Conclusion

When we apply the Random Motif Search and the Gibbs Sampler, Gibbs sampler tries and retrieves all probabilities randomly,it tries and allows all probabilities. Better to see every possibility and score better than Random Motif Search. However, the Randomized Motif Search runs faster than the Gibbs Sampler because it returns the best motif it finds first. Finally both of Randomized Motif Search and Gibbs Sampler scores are increased proportionally to k-mer's length.