CSE463 Motif Finding Assignment

1505048 - Emran Mostofa   1605068 - Abdur Rafi
1805065 - Sayeed Hasan Ovi   1805083 - Fazle Alahi Mukim
1905116 - Md. Sayeeduzzaman

March 14, 2024

Introduction

The goal of our assignment was to explore the domain of motif finding problem. We chose two methods: **Randomized Motif Search** and **Gibbs Sampler** to implement ourselves and check the output of our implementation with the ground truth. Also we explored two tools: **MEME** and **XSTREME** to further explore the motif finding problem.

1 Data

We used the **JASPAR** database for our assignment. **JASPAR** is an open access and widely used database of transcription factor (TF) binding profiles.
We used the following datasets:

- MA0003.2
  5098 sequence each of 115 length
- MA0016.1
  38 sequence each of 20 length
- MA0024.2
  1059 sequence each of 111 length

The genome sequences are available in **FASTA** format and their profiles are available as **PFM**.

2 Methods

2.1 Randomized Motif Search

Randomized motif searching is a heuristic approach used in bioinformatics for identifying conserved patterns within biological sequences. It involves randomly selecting potential motifs from the input sequences and iteratively refining them to identify the most probable motifs.

Algorithm 1 Randomized Motif Search

```
1: Randomly select k-mers from each sequence as initial motifs
2: repeat
3:     Construct a profile matrix from the current motifs
4:     for each k-mer in each sequence do
5:         Find the most probable k-mer using the profile matrix
6:     end for
7:     Update the motifs with the most probable k-mers
8: until no improvement in motif score
```

2.2 Gibbs Sampler

The Gibbs sampler is a probabilistic algorithm used in bioinformatics for motif finding. It iteratively samples motif occurrences from the input sequences based on a statistical model.

Algorithm 2 Gibbs Sampler

```
 1: Randomly select k-mers as initial motifs for each sequence
 2: repeat
 3:     for each sequence do
 4:         Randomly choose a motif in the sequence
 5:         Remove the motif from consideration
 6:         Construct a profile matrix from the remaining motifs
 7:         Sample a new motif from the profile matrix
 8:         Replace the removed motif with the sampled motif
 9:     end for
10:     Update the motifs based on the sampled motifs
11: until convergence or a maximum number of iterations is reached
```

3 Software

We used **MEME** and **XSTREME** tool on our dataset to find motif. **MEME** uses expectation-maximization algorithm to find motifs on a set of sequences. **XSTREME** uses **MEME** to find motifs. It also focuses on maximizing the statistical significance of discovered motifs directly during the motif discovery process. It uses extremal optimization to explore the search space and identify motifs that are highly overrepresented in the input sequences.
MEME is enough to discover motifs but if there is need of extra information about motifs, their position in the sequences and their statistical significance then it is better to use **XSTREME**.

4 Results

4.1 Experiment Configuration

We considered variable lengths of motifs. The minimum and maximum length we considered is 6 and 20. Our code runs the algorithm for each of the considered length. To evaluate the motif we used **consensus score**. As a result shorter length of motifs would have higher score than longer length of motifs. To choose the best length we used a heuristic that if we increase the size of the motif by one and the new alphabet matches at least 30% of the newly added alphabets then we increase the length by one, otherwise we do not increase the length and the current considered length is the best possible length. As there are random elements in the algorithm, our code runs the algorithm multiple times to get the best scoring motif.

4.2 Comparison

- MA0003.2
  True Motif: **CATTGCCTCAGGGCA**
  Motif(RMS): **TGCCTCAGGGCA** Runtime: 1146 sec
  Motif(GS) : **GCCCCAGGGGC** Runtime: 3744 sec
  Motif(MEME): **ATTGCCTCAGGGCA** Runtime: 246 sec

- MA0016.1
  True Motif: **GGGGTCACGG**
  Motif(RMS): **GGGGTCACGG** Runtime: 0.46 sec
  Motif(GS) : **GGGGTCACGG** Runtime: 26 sec
  Motif(MEME): **GGGGTCAC** Runtime: 0.28 sec

- MA0024.2
  True Motif: **CGGGCGGGAGG**
  Motif(RMS): **GGCGGGAGCGCGGGCGGGG** Runtime: 258 sec
  Motif(GS) : **GCGCGGGGGCGGGGGCGGG** Runtime: 782 sec
  Motif(MEME): **GGGCGGGAAGG** Runtime: 333 sec

5 Conclusion

We can see that in the first and third dataset **MEME** has performed much better than our algorithm and it is much faster too. But interestingly in the second dataset **MEME** has come up short on the length of the Motif whereas our algorithm has correctly found out the Motif.
The second dataset has only 38 sequence of shorter length. We can confer that our heuristic is only applicable in short lengths of sequences. To find motifs in much longer lengths we have to explore other heuristics such as expectation-maximization algorithm which is used by **MEME**.
Also **Gibbs Sampler** has performed worst in every aspect, so there is no reason to use this algorithm.