CSE463-Assignment-Motif

Sidratul Muntaha Khan
Roll: 1905009

Md Muhaiminul Islam Nafi
Roll: 1905010

Wasif Hamid
Roll: 1905026

Ajoy Dey
Roll: 1905038

Rakib Abdullah
Roll: 1905047

Abstract

In the fields of genetics and computational biology in particular, motif finding is a vital task in bioinformatics. A conserved pattern of nucleotides or amino acids found in several DNA, RNA, or protein sequences is called a motif. Identifying such conserved patterns in a collection of biological sequences is the goal of motif discovery algorithms. We have implemented four methods Randomized Motif Search, Modified Randomized Motif Search, Gibbs Sampler Motif Search and Modified Gibbs Sampler Motif Search. We also used two web tools[1] named MEME and MEME-ChIP. We compared all of these using different K values and two different motif scoring functions entropy and hamming distance.

Contents

1 Data

1.1 Ground truth

There were three dataset for motif finding. They are hm03,yst04r and yst08r. We could not find the real Motif of the data set. In the "Dataset" folder on github, we can find two types of dataset representations. One is normal sequence format and other is FASTA format.

2 Methods

We used Randomized Motif search and Gibbs Sampler Motif Search as our base two methods. We have also done some modifications to the methods.

2.1 Randomized Motif Search and Modification

We used normal Randomized Motif Search. We have also done some modifications to it. In Randomized Motif Search, when we take motifs from profile, we always take the best probability giving motif from each DNA sequence. In the modification, we have taken top k candidate motifs having top probability scores and chose randomly among them. Also we have done local optimization. This implements a local optimization approach for finding motifs based on entropy or hamming distance.

2.2 Gibbs Sampler Motif Search and Modification

We used normal Gibbs Sampler Motif Search. We have also done some modifications to it. In Gibbs Sampler Motif Search, we chose a random motif that is left out from profile calculation. In modification, we leave out the worst hamming distance giving motif.

We also used two types of scoring functions to compare the different motifs found by a wide range of K values. One is entropy and other is hamming distance. Also we used entropy scoring function for comparing the motifs of Randomized Motif Search, Modified Randomized Motif Search, Gibbs Sampler Motif Search, Modified Gibbs Sampler Motif Search and two web tools.

We iterate 100 times for both the normal and modified version.

The pseudocodes are the following:

Algorithm 1 Randomized Motif Search

```
function RandomizedMotifSearch($DNA, k, t$)
    Initialize $Motifs$ with a random $k$-mer from each string in $DNA$
    $BestMotifs \leftarrow Motifs$
    while true do
        $Profile \leftarrow$ ProfileMatrix($Motifs$)
        $Motifs \leftarrow$ MotifsFromProfile($DNA, Profile$)
        if Score($Motifs$) < Score($BestMotifs$) then
            $BestMotifs \leftarrow Motifs$
        else
            return $BestMotifs$
        end if
    end while
end function
```

Algorithm 2 Modifed Randomized Motif Search

```
function RandomizedMotifSearch($DNA, k, t$)
    Initialize $Motifs$ with a random $k$-mer from each string in $DNA$
    $BestMotifs \leftarrow Motifs$
    while true do
        $Profile \leftarrow$ ProfileMatrix($Motifs$)
        $Motifs \leftarrow$ RandomizedTopKMotifsFromProfile($DNA, Profile$)
        $Motifs \leftarrow$ localOptimization($Motifs$)
        if Score($Motifs$) < Score($BestMotifs$) then
            $BestMotifs \leftarrow Motifs$
        else
            return $BestMotifs$
        end if
    end while
end function
```

Algorithm 3 Gibbs Sampler Motif Search

```
function GibbsSampler($DNA, k, t, N$)
    Initialize $Motifs$ with a random $k$-mer from each string in $DNA$
    $BestMotifs \leftarrow Motifs$
    for $i \leftarrow 1$ to $N$ do
        Choose a random integer $j$ between 1 and $t$
        Remove the $j$-th sequence from $Motifs$
        Build a profile matrix $Profile$ from the remaining motifs in $Motifs$
        Sample a new $k$-mer from the profile $Profile$ for the $j$-th sequence
        Insert the new $k$-mer into the $j$-th position of $Motifs$
        if Score($Motifs$) < Score($BestMotifs$) then
            $BestMotifs \leftarrow Motifs$
        end if
    end for
    return $BestMotifs$
end function
```

Algorithm 4 Modified Gibbs Sampler Motif Search

```
function GibbsSampler($DNA, k, t, N$)
    Initialize $Motifs$ with a random $k$-mer from each string in $DNA$
    $BestMotifs \leftarrow Motifs$
    for $i \leftarrow 1$ to $N$ do
        Choose $j$-th sequence that has the worst hamming distance
        Remove the $j$-th sequence from $Motifs$
        Build a profile matrix $Profile$ from the remaining motifs in $Motifs$
        Sample a new $k$-mer from the profile $Profile$ for the $j$-th sequence
        Insert the new $k$-mer into the $j$-th position of $Motifs$
        if Score($Motifs$) $<$ Score($BestMotifs$) then
            $BestMotifs \leftarrow Motifs$
        end if
    end for
    return $BestMotifs$
end function
```

Algorithm 5 Compute Profile Matrix

```
function ComputeProfileMatrix(motifs)
    $profile\_matrix \leftarrow []$
    for $i$ in range(len(motifs[0])) do
        $column \leftarrow$ [motif[i] for motif in motifs]
        $profile\_matrix$.append(
        'A': ($column$.count('A') + 1) / (len($column$) + 4),
        'C': ($column$.count('C') + 1) / (len($column$) + 4),
        'G': ($column$.count('G') + 1) / (len($column$) + 4),
        'T': ($column$.count('T') + 1) / (len($column$) + 4)
        )
    end for
    return $profile\_matrix$
end function
```

Algorithm 6 Motif score entropy

```
function ComputeScore(motifs)
    $profile\_matrix \leftarrow$ ComputeProfileMatrix(motifs)
    $entropy \leftarrow []$
    for $i$ in range(len(profile_matrix)) do
        if $profile\_matrix[i]$['A'] $\neq 0$ then
            $entropy.append(-$profile_matrix[i]['A'] $\cdot \log_2(profile\_matrix[i]$['A']$))$
        end if
        if $profile\_matrix[i]$['C'] $\neq 0$ then
            entropy.$append(-$profile_matrix[i]['C'] $\cdot \log_2(profile\_matrix[i]$['C']$))$
        end if
        if $profile\_matrix[i]$['G'] $\neq 0$ then
            entropy.$append(-$profile_matrix[i]['G'] $\cdot \log_2(profile\_matrix[i]$['G']$))$
        end if
        if $profile\_matrix[i]$['T'] $\neq 0$ then
            entropy.$append(-$profile_matrix[i]['T'] $\cdot \log_2(profile\_matrix[i]$['T']$))$
        end if
    end for
    return $\sum(entropy)$
end function
```

Algorithm 7 Motif score hamming distance

```
function ComputeScore1(motifs)
    $score \leftarrow 0$
    $consensus \leftarrow$ GetConsensusMotif($motifs$)
    for each motif in $motifs$ do
        $score \leftarrow score +$ GetHammingDistance($consensus, motif$)
    end for
    return $score$
end function
```

3 Software

3.1 Commands to run

3.1.1 Randomized Motif Search & Modifed Randomized Motif Search

For “Step1_Randomized_and_modification.py” file in “Randomized_Motif_Search” folder on github,

```
$ python3 Step1\_Randomized\_and\_modification.py <input\_file>
```

For “Step2_Randomized_and_modification.py” file in “Randomized_Motif_Search” folder on github,

```
$ python3 Step2\_Randomized\_and\_modification.py <input\_file> <k\_value1> <k\_value2>
```

3.1.2 Gibbs Sampler Motif Search & Modified Gibbs Sampler Motif Search

For “Step1_Gibbs_and_modification.py” file in “Gibbs_Sampler” folder on github,

```
$ python3 Step1\_Gibbs\_and\_modification.py <input\_file>
```

For “Step2_Gibbs_and_modification.py” file in “Gibbs_Sampler” folder on github,

```
$ python3 Step2\_Gibbs\_and\_modification.py <input\_file> <k\_value1> <k\_value2>
```

3.1.3 Webtool[1] - MEME & MEMEChIP

For output comparison, in “main.py” python file in the“Web_Tools” folder on github,

```
    $ python3 main.py
```

3.2 Scripts to run

We used python language for implementing the four methods Randomized Motif Search, Modifed Randomized Motif Search, Gibbs Sampler Motif Search and Modified Gibbs Sampler Motif Search. Also for comparing all the four methods and two web tools, we created csvs from runing python files.

3.2.1 Randomized Motif Search & Modifed Randomized Motif Search

In the "Randomized_Motif_Search" folder on github, there are two python files that are needed to be run for required csv generation.. The files are "Step1_Randomized_and_modification.py" and "Step2_Randomized_and_modification.py".

3.2.2 Gibbs Sampler Motif Search & Modified Gibbs Sampler Motif Search

In the "Gibbs_Sampler" folder on github, there are two python files that are needed to be run for required csv generation.. The files are "Step1_Gibbs_and_modification.py" and "Step2_Gibbs_and_modification.py".

3.2.3 Webtool[1] - MEME & MEMEChIP

We can submit the FASTA files of the dataset in the webtools `https://meme-suite.org/meme/` and `https://meme-suite.org/meme/tools/meme-chip`. It queues the jobs and gives us the result after 1-2 hours. Then "main.py" python file in the "Web_Tools" folder on github is needed to be run for required csv generation.

4 Results

4.1 Exp. configuration

Firstly, we have implemented the four algorithms: Randomized Motif Search, Modified Randomized Motif Search, Gibbs Sampler Motif Search, Modified Gibbs Sampler Motif Search.

Then, we run it for different K(Length of the motif) values(ranging from 10-49). We used entropy motif scoring during the running. We also compared all four methods using the entropy motif scores for each K value. We divided the motif scores by K and got the avg motif scores. We used the avg motif scores for the comparison.

Then, we run each of the four methods for a particular K using two scoring functions entropy and hamming distance.

Then, we compared all four methods with entropy motif scoring, all four methods with hamming distance motif scoring, two web tool results(In total 10) using the entropy motif scoring as a metric.
We did it in two phases. One is all four methods with entropy motif scoring, all four methods with hamming distance motif scoring and MEME web tool. Other one is all four methods with entropy motif scoring, all four methods with hamming distance motif scoring and MEME-ChIP web tool.

Lastly, we compared the two webtools using he entropy motif scoring as a metric.

We did all of the above for all three dataset.

4.2 Comparison

4.2.1 Comparison of all four methods scores vs K values

We ran all four methods with a range(10-49) of k values for three dataset.
hm03:
Results for Randomized and its modification can be found at : Supplementary file
Results for Gibbs Sampler and its modification can be found at : Supplementary file
For chart, We can find the results from: Supplementary file
yst04r:
Results for Randomized and its modification can be found at : Supplementary file
Results for Gibbs Sampler and its modification can be found at : Supplementary file
For chart, We can find the results from: Supplementary file
yst08r:
Results for Randomized and its modification can be found at : Supplementary file
Results for Gibbs Sampler and its modification can be found at : Supplementary file
For chart, We can find the results from: Supplementary file

From the charts, we can see that for hm03 and yst08r, Gibbs Sampler performs overall well. For yst04r, Gibbs Sampler and Modified Gibbs Sampler perform overall well.

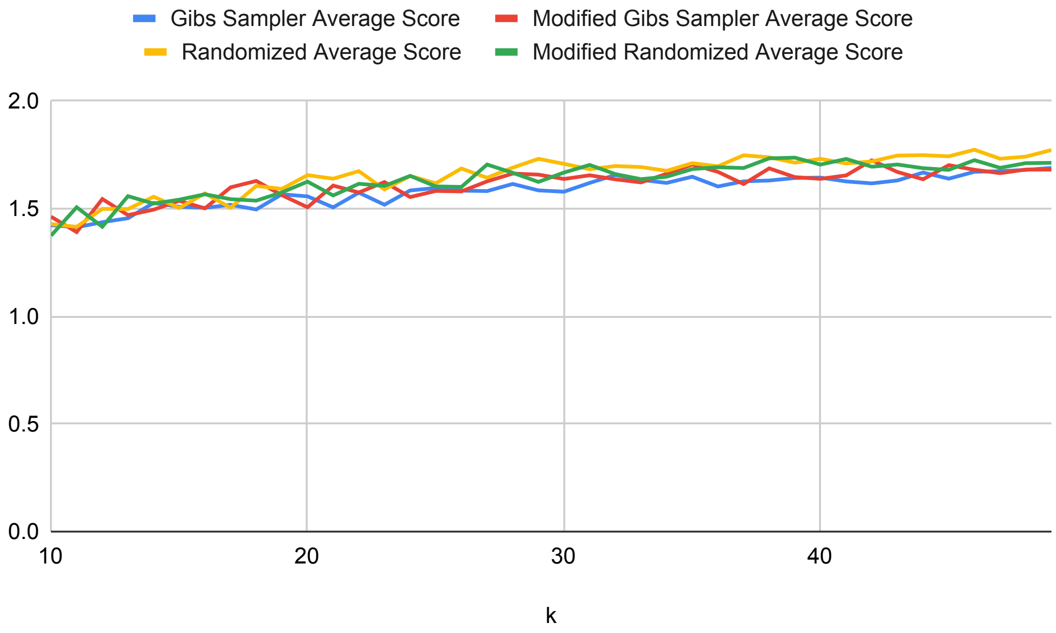

Figure 1: hm03

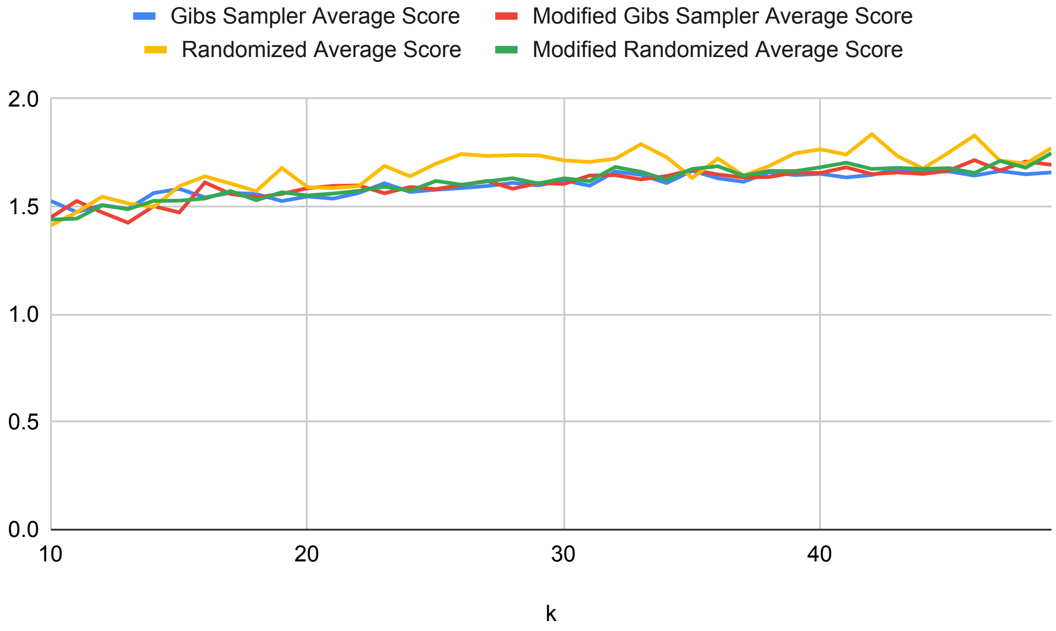

Figure 2: yst04r

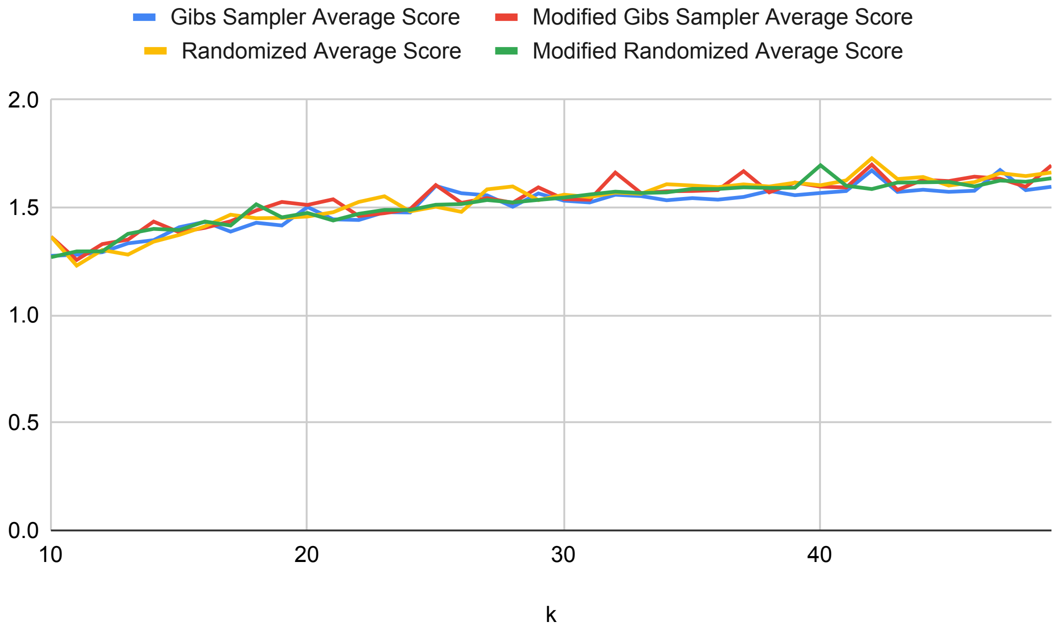

Figure 3: yst08r

4.2.2 Comparison of all four methods time vs K values

For hm03, the results can be found at: Supplementary file
For yst04r, the results can be found at: Supplementary file
For yst08r, the results can be found at: Supplementary file

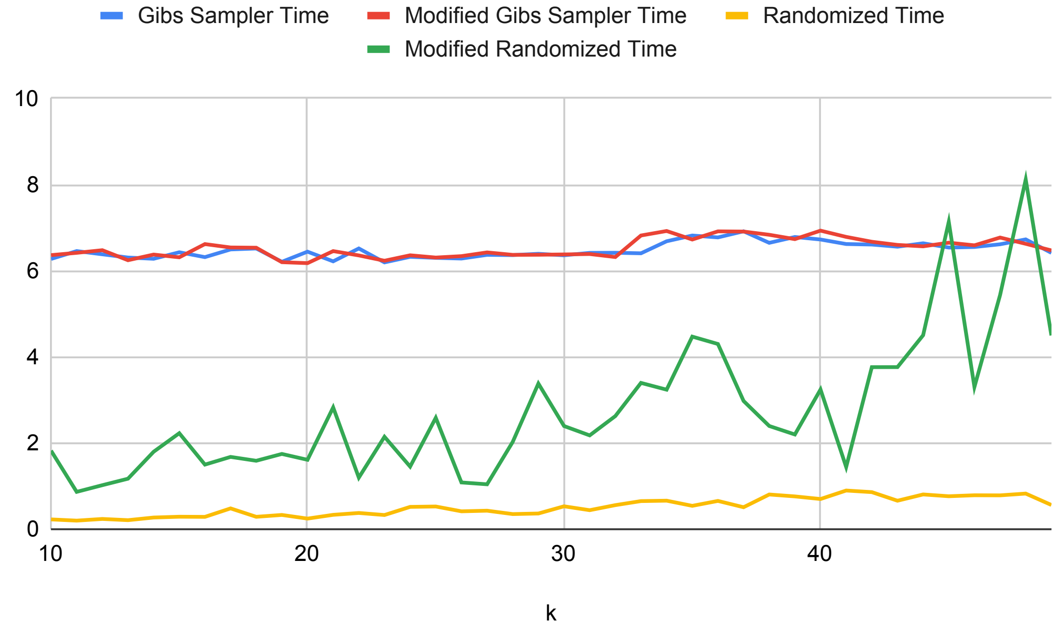

Figure 4: hm03

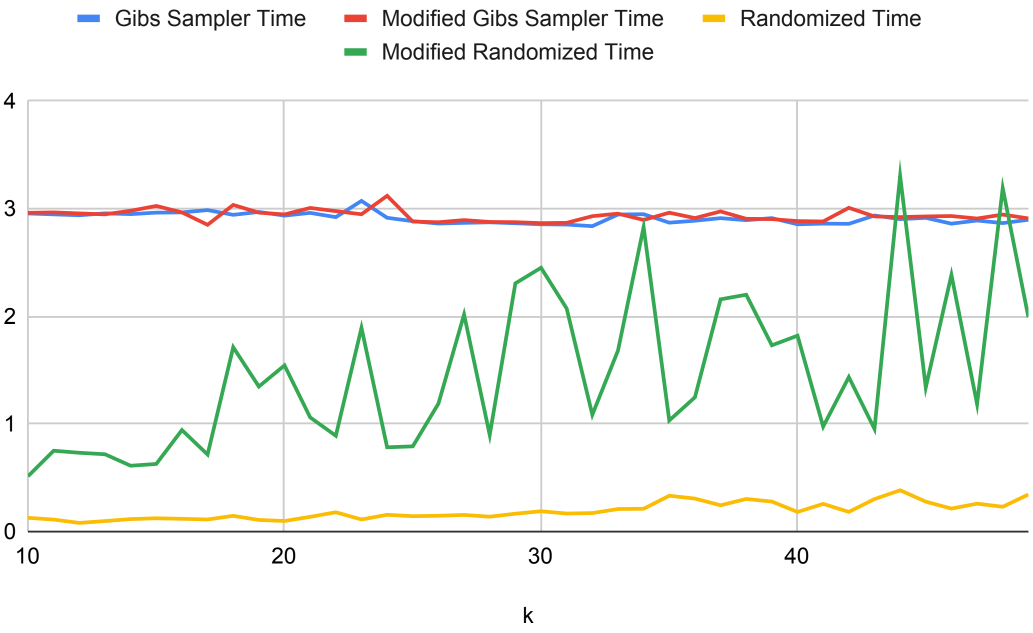

Figure 5: yst04r

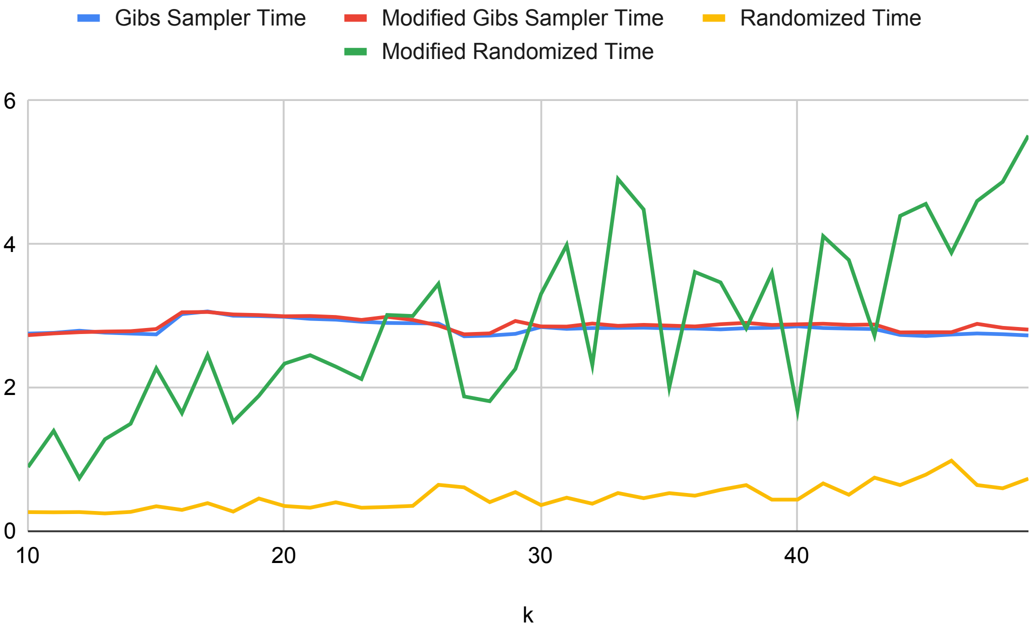

Figure 6: yst08r

4.2.3 All 8 methods

Detailed results for first 4 (Randomized Score With Entropy, Randomized Score With Hamming, Modified Randomized Score With Entropy, Modified Randomized Score With Hamming) can be found at:Supplementary file
Detailed results for last 4 (Gibs Sampler Score With Entropy, Gibs Sampler Score With Hamming, Modified Gibs Sampler Score With Entropy, Modified Gibs Sampler Score With Hamming) can be found at:Supplementary file

4.2.4 Comparison of all eight methods and MEME

For all three dataset, we generated results for all 8 methods(Randomized Score With Entropy, Randomized Score With Hamming, Modified Randomized Score With Entropy, Modified Randomized Score With Hamming, Gibs Sampler Score With Entropy, Gibs Sampler Score With Hamming, Modified Gibs Sampler Score With Entropy, Modified Gibs Sampler Score With Hamming) and MEME. Results can be found from: Supplementary file. It was used for chart generation.
From the graph we can see that, MEME performed better for all three dataset.

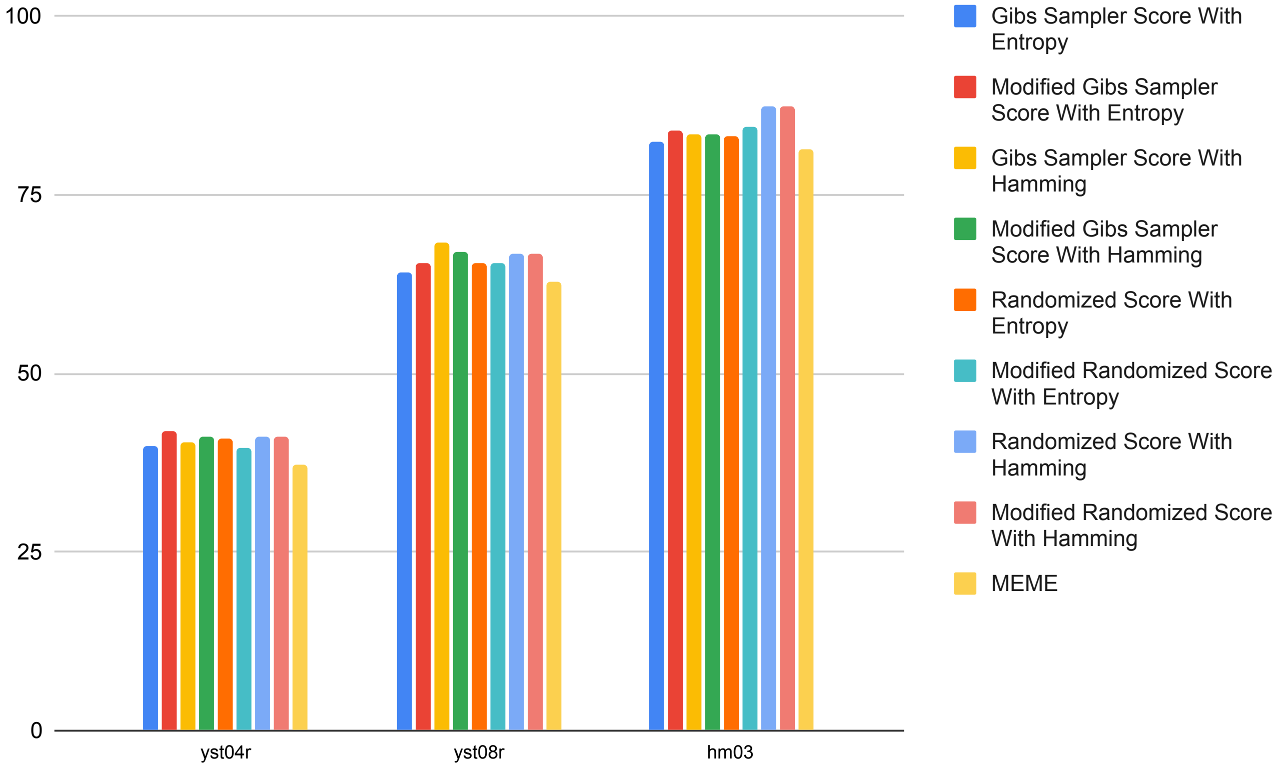

Figure 7: All 8 and MEME

4.2.5 Comparison of all eight methods and MEME-ChIP

For all three dataset, we generated results for all 8 methods(Randomized Score With Entropy,Randomized Score With Hamming, Modified Randomized Score With Entropy, Modified Randomized Score With Hamming, Gibs Sampler Score With Entropy, Gibs Sampler Score With Hamming, Modified Gibs Sampler Score With Entropy, Modified Gibs Sampler Score With Hamming) and MEME-ChIP. Results can be found from: Supplementary file. It was used for chart generation.
From the chart, we can see that Modified Randomized Score With Entropy performed better for yst04r and yst08r. Gibs Sampler Score With Entropy performed better for hm03.

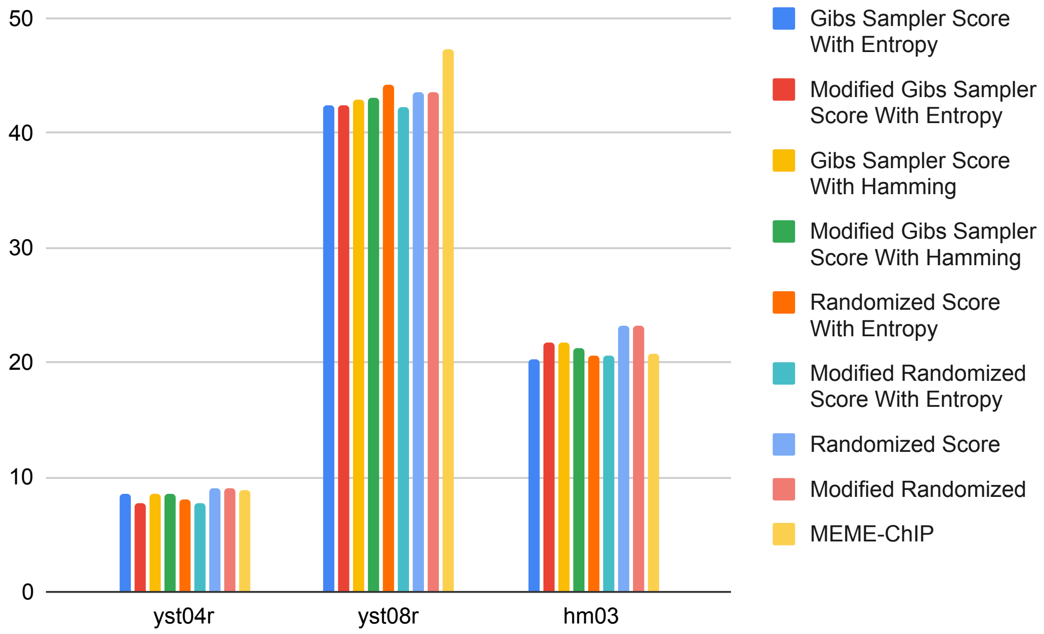

Figure 8: All 8 and MEME-ChIP

4.2.6 MEME vs MEME-ChIP

We comapred it based on avg entropy motif score. Avg entropy score is calculated using whole entropy divided by k. We can see that MEME perfoms better for yst08r and MEME-ChIP performs better for yst04r and hm03.

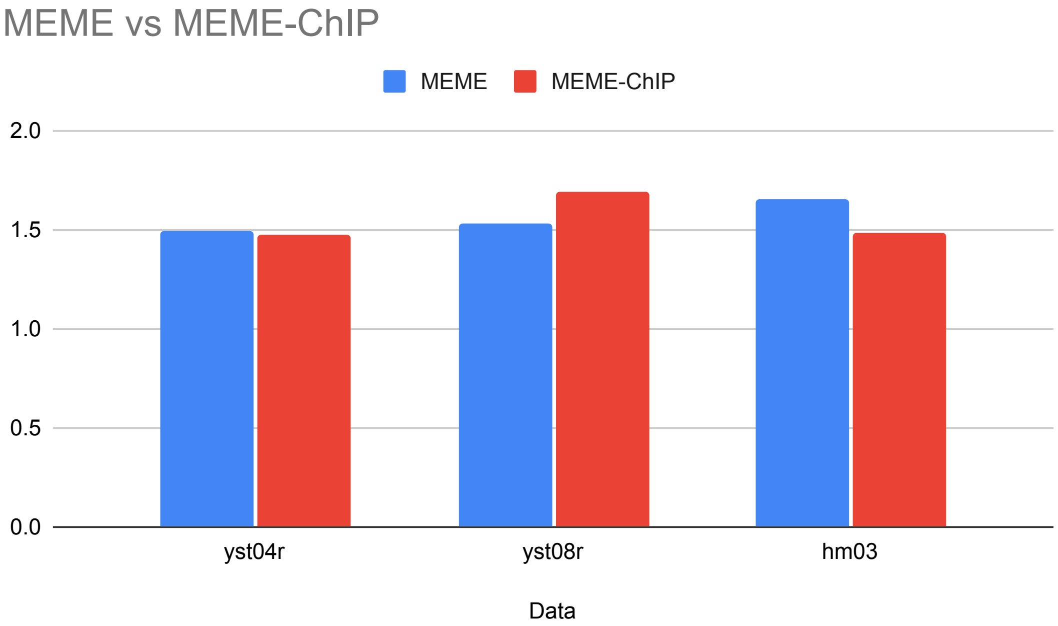

Figure 9: MEME vs MEME-ChIP

5 Conclusion

In the comparison of the main four methods scores with wide range of k values, we can see that Randomized Motif Search perform relatively worse than others for all three dataset and Gibbs Sampler perform well in overall for all three dataset.
In the comparison of the main four methods times with wide range of k values, we can see that Randomized Motif Search perform faster than the others. Gibbs Sampler and Modified Gibbs Sampler go with overall constant time. So if we want a faster search, we can go with Randomized Motif Search. If we want a moderate speed search with a constant speed across all k values and better motif, we can go with Gibbs Sampler.
In the comparison with all 8 and MEME, MEME performs better. But MEME-ChIP performs worse than the others in all 8 and MEME-ChIP chart. We see, MEME-ChIP perform better than MEME for two dataset in MEME vs MEME-ChIP chart. The contradictions occur due to the less k value of MEME-ChIP. As we have taken avg score entropy values of both MEME and MEME-ChIP motif set and MEME-ChIP has less k value than MEME, MEME-ChIP takes the advantage in the metric. But on the real score evaluations, MEME perform better than MEME-ChIP.

References

[1] Timothy L. Bailey, James Johnson, Charles E. Grant, and William S. Noble. The MEME Suite. *Nucleic Acids Research*, 43(W1):W39–W49, 05 2015.