Motif Finding Algorithm

CS412 Final Project Report

Alberto Alvarez Aldea (alberto6)
Jiangyan Feng (jf8)
Sayed Shazeb Hussain (sayedsh2)
University of Illinois at Urbana-Champaign
May 4, 2019

1 Algorithm Description

The whole process of the motif finder is shown in Figure 1. The two required inputs are “sequences.fa" and “motiflength.txt", which define two parameters sequence count (*SC*) and motif length (*ML*).

(1) **Generate a preliminary list of candidate motifs**. Based on the first sequence in the “sequences.fa", we enumerate all the ML-mers and generate a list of candidate motifs.

(2) **Find the most similar sites from the remaining sequences**. For each candidate motif, we find the most similar ML-mers from the remaining *SC-1* sequences. In total, we can generate a SC*1 index list for each candidate motif. To measure the similarity between two sites, we employ a python package **SequenceMatcher**. The similarity is measured by the formula:

$$Dro = \frac{2K_m}{\mid S_1 \mid + \mid S_2 \mid}, Dro \in [0,1] \qquad (1)$$

where $\mid S_1 \mid$ and $\mid S_2 \mid$ refer to the number of characters in two strings S_1, S_2, respectively. K_m refers to the number of matching characters. The larger the Dro, the more similar between two sites.

(3) **Calculate the sum of ICPC for each candidate motif**. For each candidate motif, we calculate the sum of ICPC over all the columns from the previously found *SC*1* index list.

(4) **Select the motif**. Based the sum of ICPC for each candidate motif, we select the candidate motif with the highest sum of ICPC as the predicted motif.

(5) **Output “predictedmotif.txt", “predictedsites.txt"**. Based on the predicted motif and the recorded *SC*1* index list information for the motif, we calculate the motif probability matrix and generate two outputs: “predictedmotif.txt", “predictedsites.txt".

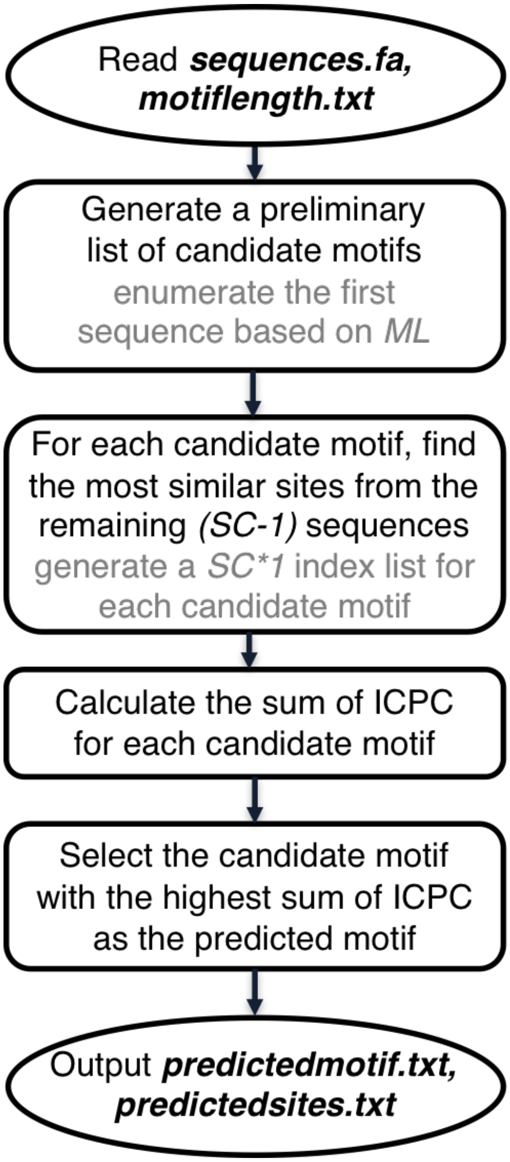
Read ***sequences.fa, motiflength.txt***
Generate a preliminary list of candidate motifs
enumerate the first sequence based on *ML*
For each candidate motif, find the most similar sites from the remaining *(SC-1)* sequences
generate a *SC*1* index list for each candidate motif
Calculate the sum of ICPC for each candidate motif
Select the candidate motif with the highest sum of ICPC as the predicted motif
Output ***predictedmotif.txt, predictedsites.txt***

Figure 1: The flow chart showing the whole process of the motif finder.

2 Performance Evaluation

To evaluate the effectiveness of the motif finder, we applied it to 70 data sets, 10 data sets for each parameter combination (as shown in Table 1). The performance for each parameter combination is averaged over 10 data sets. In total, 4 evaluation criteria (Kullback-Leibler divergence, number of overlapping positions, number of overlapping sites, and running time) are applied. We find that (1) Information content per column (ICPC) affects the accuracy of the motif finder the most in terms of Kullback-Leibler divergence, the normalized number of overlapping positions, and the normalized number of overlapping sites. The higher the ICPC (the sharper the motif), the better the performance. (2) Motif length (ML) has little effect on the accuracy of predicted positions. The larger the ML (the larger the motif), the better the performance. (3) Sequence count (SC) has no effect on the accuracy of the motif finder in terms of Kullback-Leibler divergence, the normalized number of overlapping positions, and the normalized number of overlapping sites. (4) The running time increases linearly with ML, whereas the running time increases exponentially with SC. It suggests that it will become computationally expensive to find a motif from a large set of sequences.

Table 1: Results of four evaluation metrics for seven parameter combinations.

Index	ICPC	ML	SL	SC	KL divergence	overlapping positions	overlapping sites	running time
1	2	8	500	10	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	87.30 ± 0.62
2	1	8	500	10	2.22 ± 0.43	0.01 ± 0.03	0.09 ± 0.11	87.51 ± 0.89
3	1.5	8	500	10	0.79 ± 1.02	0.66 ± 0.35	0.73 ± 0.34	87.39 ± 1.08
4	2	6	500	10	0.00 ± 0.00	0.92 ± 0.10	1.00 ± 0.00	70.88 ± 0.55
5	2	7	500	10	0.00 ± 0.00	0.98 ± 0.04	1.00 ± 0.00	78.87 ± 1.05
6	2	8	500	5	0.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	39.15 ± 0.42
7	2	8	500	20	0.00 ± 0.00	0.99 ± 0.01	1.00 ± 0.00	183.23 ± 1.28

2.1 Metric 1: Kullback-Leibler (KL) divergence

Kullback-Leibler (KL) divergence measures the difference between two probability distributions, which can be expressed as equation 2. The smaller the KL divergence, the more similar the two probability distributions. In our case, the smaller the KL divergence, the better the predictions. $D_{KL} = 0$ indicates a perfect prediction. Comparing “motif.txt" and “predictedmotif.txt", we calculated the KL divergence for each position in the motif and then averaged KL divergence over all the positions in the motif.

$$D_{KL}(P \parallel Q) = \sum P(x) ln(\frac{P(x)}{Q(x)}), D_{KL} \geq 0 \tag{2}$$

Figure 2 shows the KL divergence varying with 3 experimental parameters (information content per column (ICPC), motif length (ML) and sequence count (SC)). In Figure 2a, KL divergence drops abruptly with the increasing ICPC. When $ICPC = 2$, $D_{KL} = 0$ which indicates a perfect prediction. It suggests that it is easier for the motif finder to detect a very 'sharp' pattern (high ICPC). When the $ICPC$ is small, the motif finder may fail to detect the correct motif due to the weak signal from the pattern. In Figure 2b and 2c, KL divergence stays at 0, suggesting that there is little or no effect of ML and SC on KL divergence. Therefore, we suggest that (1) ICPC determines the performance of the motif finder in terms of KL divergence, and the higher the ICPC (the sharper the motif), the better the performance. (2) ML and SC have no effect on the performance of the motif finder in terms of the KL divergence.

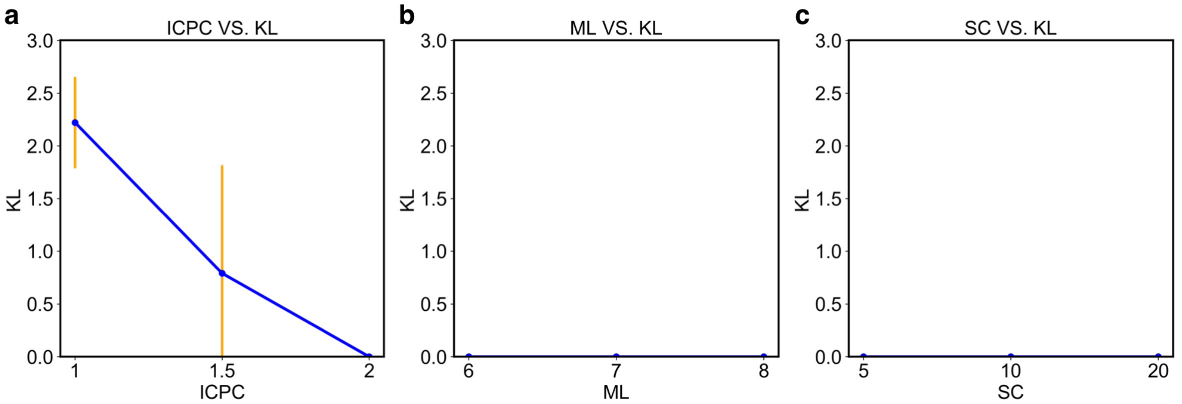

Figure 2: The Kullback-Leibler (KL) divergence varying with (a) information content per column (ICPC), (b) motif length (ML) and (c) sequence count (SC). Data for all experiments are mean (blue) ± standard deviation (orange) of ten datasets.

2.2 Metric 2: number of overlapping positions

Comparing "sites.txt" and "predictedsites.txt", we calculated the overlapping positions for each sequence and then normalized the total number overlapping positions by $ML \times SC$ to ensure the number of overlapping positions falls into the range of [0, 1]. The higher the normalized overlapping positions, the better the performance.

$$normalized\,overlapping\,positions = \frac{\sum_i^{SC}\sum_j^{ML} overlapping\,position}{ML \times SC} \tag{3}$$

where overlapping position is 1 if the predicted index matches with the known index; otherwise, overlapping position is 0. The total number of overlapping positions is normalized by $ML \times SC$.

Figure 3 shows the normalized number of overlapping positions varying with 3 experimental parameters (information content per column (ICPC), motif length (ML) and sequence count (SC)). In Figure 3a, the normalized number of overlapping positions increases from 0 to 1 as the ICPC increases from 1 to 2. This suggests the performance of the motif finder increases with the ICPC and the motif finder fails to predict the accurate positions when ICPC is low ($ICPC = 1$). In Figure 3b, the normalized number of overlapping positions is higher than 0.8 in three cases which indicates a good performance of the motif finder in all 3 cases. However, we observe the normalized number of overlapping positions increases as the ML increases from 6 to 8. This suggests that the larger the motif, the easier of the motif to be detected. In Figure 3c, the normalized number of overlapping positions stays at 1 which indicates that SC has no effect on the performance of the motif finder. Therefore, we suggest (1) ICPC determines the performance of the motif finder in terms of the normalized number of overlapping positions, and the higher the ICPC (the sharper the motif), the better the performance; (2) the larger the ML (the larger the motif), the easier of the motif to be detected by the motif finder; (3) SC has no effect on the performance of the motif finder in terms of the normalized number of overlapping positions.

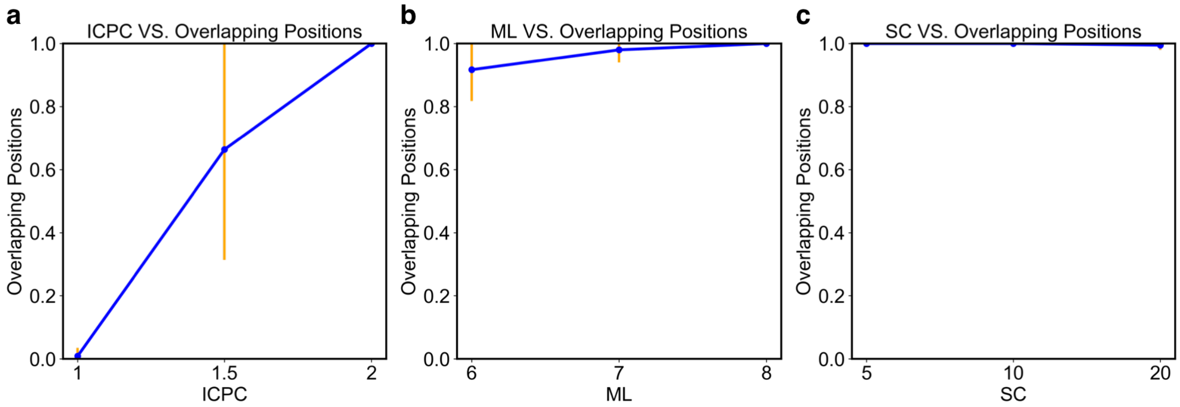

Figure 3: The normalized number of overlapping positions varying with (a) information content per column (ICPC), (b) motif length (ML) and (c) sequence count (SC). Data for all experiments are mean (blue) ± standard deviation (orange) of ten datasets.

2.3 Metric 3: number of overlapping sites

Comparing “sites.txt" and “predictedsites.txt", we checked whether the site is overlapping for each sequence and then normalized the total number overlapping sites by SC to ensure the number of overlapping sites falls into the range of [0, 1]. The higher the normalized overlapping sites, the better the performance.

$$normalized\, overlapping\, sites = \frac{\sum_i^{SC} overlapping\, site}{SC} \tag{4}$$

where overlapping site is 1 if at least $\frac{ML}{2}$ of the predicted positions in the predicted site match with the known site; otherwise, overlapping position is 0. The total number of overlapping sites is normalized by SC.

Figure 4 shows the normalized number of overlapping sites varying with 3 experimental parameters (information content per column (ICPC), motif length (ML) and sequence count (SC)). In Figure 4a, the normalized number of overlapping sites increases abruptly with the increasing ICPC. When $ICPC = 2$, $normalized\, overlapping\, sites = 1$ which indicates a perfect prediction. Similar to the finding in KL divergence metric, it suggests that it is easier for the motif finder to detect a very ‘sharp’ pattern (high ICPC). When the $ICPC$ is small, the motif finder may fail to detect the correct site due to the weak signal from the pattern. In Figure 4b and 4c, the normalized number of overlapping sites stays at 1, suggesting that there is little or no effect of ML and SC on the normalized number of overlapping sites. Therefore, we suggest that (1) ICPC determines the accuracy of predicted sites detected by the motif finder, and the higher the ICPC (the sharper the motif), the better the performance; (2) ML and SC have no effect on the performance of the motif finder in terms of the normalized number of overlapping sites.

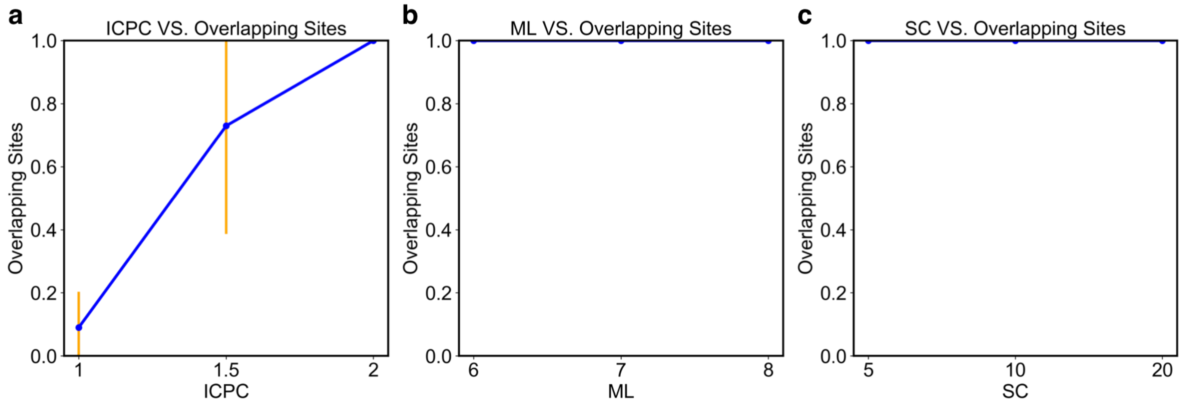

Figure 4: The normalized number of overlapping sites varying with (a) information content per column (ICPC), (b) motif length (ML) and (c) sequence count (SC). Data for all experiments are mean (blue) ± standard deviation (orange) of ten datasets.

2.4 Metric 4: running time

In order to check the effect of experimental parameter on the efficiency of the motif finder, we compared the running time required for each parameter combination (as shown in Figure 5). In Figure 5a, running time does not vary a lot with the increase of ICPC. This suggests that ICPC will not affect the running time. In Figure 5b, the running time increases linearly as the increase of ML. This suggests that the running time of motif finder will scale linearly with the increase of ML. In Figure 5c, the running time increases nearly exponentially as the increase of SC. This indicates that the computational cost will not scale well with SC and it will become computationally expensive if the SC is very large.

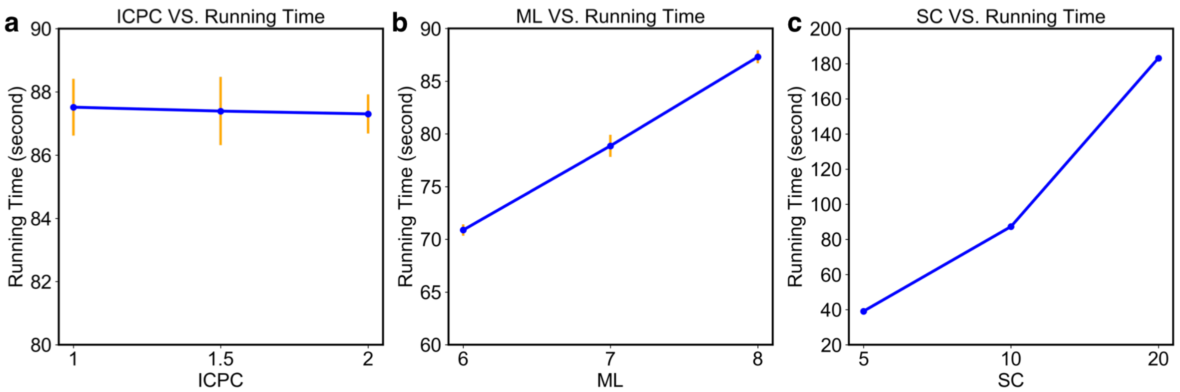

Figure 5: Running time varying with (a) information content per column (ICPC), (b) motif length (ML) and (c) sequence count (SC). Data for all experiments are mean (blue) ± standard deviation (orange) of ten datasets.