

Project Report

1. Algorithm description

Before describing the main motif-finder algorithm, we shall first give a brief introduction to motif-generator algorithm for testbench generator, which is a two-step random process: (1) For each column of probability weight matrix (PWM), first randomly generate two probabilities, then solve equation for the rest two probability with the constraints of ICPC and Unitarity of probability distribution; (2) Rearrange each column of PWM, to guarantee each gene AGCT has equal expectation in motif-generator algorithm.

For the main motif-finder algorithm, we have complemented two algorithms: Expectation Maximization (EM) and Expectation Brute Force (E-Brute). The first one EM is learnt from the literature¹, the second one E-Brute is developed by ourselves.

We will start with EM first, and then describe how it evolves into E-Brute. The final evaluation and results are accomplished with E-Brute.

1.1 EM algorithm

Algorithm 2: Expectation Maximization¹ (EM)

Input: SM = Sequence Matrix, ML = Motif length

Output: MS (Motif Site), PWM (Probability weight matrix)

While not convergence **do**:

```
#####  
      E-step  
#####
```

for sequence i **do**:

for position j **do**:

$$\Pr(X_i | Z_{ij} = 1, p) = \underbrace{\prod_{k=1}^{j-1} p_{c_k,0}}_{\text{before motif}} \underbrace{\prod_{k=j}^{j+W-1} p_{c_k,k-j+1}}_{\text{motif}} \underbrace{\prod_{k=j+W}^L p_{c_k,0}}_{\text{after motif}}$$

¹ <https://www.biostat.wisc.edu/~craven/776/lecture9.pdf>

$$Z_{ij}^{(t)} = \frac{\Pr(X_i | Z_{ij} = 1, p^{(t)}) \Pr(Z_{ij} = 1)}{\sum_{k=1}^{L-W+1} \Pr(X_i | Z_{ik} = 1, p^{(t)}) \Pr(Z_{ik} = 1)}$$

#####

M-step

#####

for position k **do**:

for character c **do**:

$$p_{c,k}^{(t+1)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$

pseudo-counts

$$n_{c,k} = \begin{cases} \sum_i \sum_{\{j | X_{i,j+k-1}=c\}} Z_{ij} & k > 0 \\ n_c - \sum_{j=1}^W n_{c,j} & k = 0 \end{cases}$$

total # of c's in data set

Where

$$PWM = [[p_{c,k}]]$$

#####

Extract MS

#####

for sequence i **do**:

$$MS_i = MS_i = \arg \max_j Z_{ij}$$

1.2 From EM to E-Brute

a. Overcome local optimum:

The major problem of EM we observed is that this algorithm often falls into local optimum. This means that EM algorithm is not robust, and depends heavily on the initial condition. In this way, if we want to find the global optimum, we would probability need to traverse all possible initial condition.

Now that every initial condition required to be traversed, we come out with the idea to just eliminate the M-step in EM algorithm, and directly compare the expectations after

assigning initial condition. This idea is applied with our project case, where only one probability of the total four gene is dominant.

b. Refine expectation term:

We made a refinement for the expectation term in EM algorithm, to make it best fit our E-Brute algorithm, as well as cutting down the computational complexity.

The refinement is to ignore the probability (0.25) bring by the other location more than the assumed motif when calculating the expectation of each site. The feasibility of this simplification is due to the normalization step before calculating out the final expectation. This simplification cutting down the complexity from $O(*S)$ to $O(*M)$. Where S is sequence length, M is motif length, * is the complexity from other parts of the algorithm.

c. Sharpen PWM matrix:

The PWM predicted by EM algorithm has the tendency to be uniform distribution, because it counts the pattern for the whole sequence, more than just the site where motif is planted. The large number of irrelevant sequence patterns brings overwhelming noise to the predicted PWM.

Our method is to first extract the motifs from sequences based on the motif sited we predicted. Then PWM is calculating merely on these extracted motifs. This method can block out the noise from other irrelevant sites.

1.3 E-Brute Algorithm

Algorithm 2: Expectation Brute Force (E-Brute)

Input: SM = Sequence Matrix, ML = Motif length

Output: MS (Motif Site), PWM (Probability weight matrix)


```
#####  
 Traverse the initial_PWM_all  
#####
```

Generate all the possible initial PWMs and saved as initial_PWM_all. In each column of initial PWM, the probability weight of the dominate gene is 10 times of other gene. (For example, in one column, the probability weight vector could be [1, 1, 10, 1] corresponding to the alphabets [A G C T])

```
#####  
 Find the maximum expectation  
#####
```

for initial_PWM_i in initial_PWM_all **do**:
 for position k **do**:
 for character c **do**:

$$\Pr(X_i | Z_{ij} = 1, p) = \prod_{k=j}^{j+W-1} p_{c_k, k-j+1}$$



$$Z_{ij}^{(t)} = \frac{\Pr(X_i | Z_{ij} = 1, p^{(t)}) \Pr(Z_{ij} = 1)}{\sum_{k=1}^{L-W+1} \Pr(X_i | Z_{ik} = 1, p^{(t)}) \Pr(Z_{ik} = 1)}$$

Maintain the Z_{ij} with the largest product

for sequence i **do**:

$$MS_i = MS_i = \arg \max_j Z_{ij}$$

#####

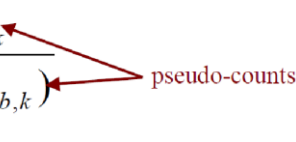
Calculate PWM according to the above-found motif sites

#####

for position k **do**:

for character c **do**:

$$p_{c,k}^{(t+1)} = \frac{n_{c,k} + d_{c,k}}{\sum_b (n_{b,k} + d_{b,k})}$$



Where $n_{c,k}$ is the number of character c in position k in all sequences.

$$PWM = [[p_{c,k}]]$$

2. Performance evaluation

3. Observation from the results

By comparing the performance of E-Brute algorithm upon different parameters, we can observe that the case with lower ICPC is more difficult for motif discovery. The PWM

of ICPC = 2 can be completely mined out, while the site prediction achieved more than 90%(Needs to be specified) where the site error is due to the duplication of motif pattern in the original sequences. There are two explanations for these observations as follows:

In the case with ICPC = 1, the motif-finder algorithm is likely to make mistakes because the randomly generated gene sequences could coincidentally contain the sequence pattern with even higher ICPC. For example, in our dataset of “ICPC1,ML8,SC10.run6”. The predicted motif founded from gene sequences are shown as figure 1, whose ICPC is even higher than the actual latent PWM we used when generating these sequences (presented as figure 2)

```
>PREDICTEDMOTIF 8
0.0 0.2 0.1 0.7
0.1 0.0 0.8 0.1
0.9 0.1 0.0 0.0
0.0 0.2 0.0 0.8
0.0 0.0 0.9 0.1
0.1 0.0 0.1 0.8
0.0 0.2 0.0 0.8
0.2 0.0 0.8 0.0
<
```

Figure 1 predicted PWM

```
>MOTIF1 8
0.028 0.799 0.11499999999999988 0.058
0.81 0.056999999999999994 0.078 0.055
0.774 0.16 0.007 0.05899999999999994
0.779 0.163 0.032 0.026
0.20299999999999996 0.034 0.754 0.009
0.001 0.019 0.28700000000000003 0.693
0.002 0.051 0.755 0.192
0.006 0.755 0.04 0.199
<
```

Figure 2 actual PWM

In the case with ICPC=2, there is seldom a mistake in predicted site, while the predicted PWM is exactly the same with the actual one. This is because there are two sites in the certain sequence where motif is delicately appearing. For example, in our dataset of “ICPC2,ML6,SC10”, motif CCAAAC is appeared in both site 9 and site 235 in the 5th sequence. In this case, there is no way to figure out the definite site between 9 and 235, but this brings no obstacle in finding out the exact motif.