Name: 444221516 OUYANG SHIYU
Email: oyshiyu@toki.waseda.jp
Code: https://github.com/hanpier/MEME.git

Motif Finding with MEME

1 Introduction

Motif finding is a central problem in computational biology, with applications in the study of gene function and regulation, protein-DNA interactions, and evolutionary relationships. A motif is a short, recurring pattern of nucleotides or amino acids that has some biological significance. Identifying motifs in biological sequences can provide valuable insights into the underlying mechanisms of cellular processes.[1]

However, motif finding is a challenging task due to the large number of possible motifs, the high degree of variability within motifs, and the noise present in biological sequences. To address these challenges, a variety of methods have been developed for motif finding, including algorithmic approaches such as Gibbs sampling and expectation maximization, and statistical methods such as information theory and machine learning. This project describes a deterministic method called MEME to discover several different motifs of differing and fixed width in protein dataset.

2 Introduction

Motif finding is a central problem in computational biology, with applications in the study of gene function and regulation, protein-DNA interactions, and evolutionary relationships. A motif is a short, recurring pattern of nucleotides or amino acids that has some biological significance. Identifying motifs in biological sequences can provide valuable insights into the underlying mechanisms of cellular processes.[1]

However, motif finding is a challenging task due to the large number of possible motifs, the high degree of variability within motifs, and the noise present in biological sequences. To address these challenges, a variety of methods have been developed for motif finding, including algorithmic approaches such as Gibbs sampling and expectation maximization, and statistical methods such as information theory and machine learning. This project describes a deterministic method called MEME to discover several different motifs of differing and fixed width in protein dataset.

2.1 Overview of MEME

MEME (Multiple Expectation Maximization for Motif Elicitation) is a popular algorithm for motif finding that uses an expectation maximization (EM) approach. The basic idea behind MEME is to search for motifs by iteratively estimating the probability of each position in the motif and re-estimating the motif based on these probabilities.

In MEME, the user specifies the length of the motif and the number of occurrences of the motif in the input sequences. The algorithm begins by randomly selecting initial positions for the motif instances and using these positions to estimate the probability of each position in the motif. The motif is then re-estimated based on these probabilities, and the process is repeated until convergence. In this project, a modified MEME algorithm proposed in[2], which is the third version of MEME, will be implemented to overcome those limitations in the early versions of MEME. Specifically, the innovations will include:

- Use of Dirichlet mixture priors to initialize PSSM.
- A global search algorithm 'TEST' based on approximated EM heuristics for choosing optimum starts points for MEME.

In general, the MEME algorithms find a different motif with optimum width at each pass and by specifying the number of pass for running MEME, multiple motifs could be found based on EM algorithm. It determines good start points and avoids being stuck at local optima running

EM without repeatedly re-running from different random starting points. Above all, same as its early versions, this MEME algorithm eliminates the best-founded motif 'probabilistic-ally' at a time and avoids rediscovery of the same motif in the process of find succeeding optimum motifs.

2.2 Restatement of the Problem

Considering the background information and restricted conditions identified in the problem statement, we need to solve the following problems:

- Find a statistical model like PSSM to score the probability of each position in the motif.
- Find the start position of motif.

2.3 My Work

Considering One Occurrence Per Sequence model. This model assumes there is exactly one occurrence per sequence of motif in the dataset. Here is a flow chart illustrating the steps of this MEME algorithm:

- Initialization: The length of the motif and the number of occurrences (instances) of the motif in the input sequences are specified. Initial positions for the motif instances are randomly selected.
- Expectation step: The probability of each position in the motif is estimated based on the observed frequencies of each residue (amino acid or nucleotide) at that position. This probability is stored in a Position Specific Scoring Matrix (PSSM).
- Maximization step: The motif is re-estimated based on the PSSM and the initial positions of the motif instances are updated.
- Convergence: The EM process is repeated until convergence, at which point the final motif and PSSM are output.

3 Model

There are three types of models supported by MEME specifically the OOPS, the ZOOPS and the TCM models.

OOPS: One Occurrence Per Sequence model. This model assumes there is exactly one occurrence per sequence of motif in the dataset.

ZOOPS: Zero or One Per Sequence model. This model assumes zero or one motif occurrences per sequence.

TCM: Two Component Mixture model. This model assumes that there are zero or more nonoverlapping occurrences of the same motif in the sequence.

In the project, only the OOPS model is implemented and discussed. The parameters associated with this model is as equation(1)

$$\theta = [\theta_0 \quad \theta_1] = [p_0 \quad p_1 \quad p_2 \quad \cdots \quad p_W] = \begin{bmatrix} P_{A,0} & P_{A,1} & P_{A,2} & \dots & P_{A,W} \\ P_{C,0} & P_{C,1} & P_{C,2} & \dots & P_{C,W} \\ P_{D,0} & P_{D,1} & P_{D,2} & \dots & P_{D,W} \\ \dots & \dots & \dots & \dots & \dots \\ P_{Y,0} & P_{Y,1} & P_{Y,2} & \dots & P_{Y,W} \end{bmatrix} \tag{1}$$

The alphabet for the sequences in this project is A, C, D, E, F…, Y. Hence, $P_{x,j}$ is the

probability of letter x occurring at a background if $j = 0$ or at at motif position j if $1 \le j \le W$.

Additionally, the OOPS model incorporates a new parameter γ - the prior the prior probability of a sequence model containing a motif occurrence, where $\gamma = 1$. Thus, assume a sequence of length L and motif length of W, the number of possible motif start position is m, where $m = L - W + 1$, thereby the prior probability of a sequence model contain a motif starting at either one of the m possible position could be expressed as λ, where $\lambda = \frac{1}{\mathrm{m}}$.

4 Training

The dataset contains n sequences of different length as $X = X_1, X_2, \ldots X_n$. The model includes an indicator variable called the 'missing information' denoted by Z, where $Z_{ij} = 0$ meaning a motif starts at position j of X_j.

To compute the optimum starting point of motif, the model runs an EM algorithm. On a high level, the E step is to variable Z and M step is to update each column of PSSW matrix $p_k (1 \le k \le W)$ and p_0. The detailed update rule is shown below.

4.1 Conditional Probability

Here lists the conditional probabilities used in the EM steps.

$$\log \Pr(X, Z | \ \theta, \lambda) = \sum_{k=0}^{W-1} \mathbf{I}(i, j+k)^T \log p_k + \sum_{k \in \Delta_{i,j}} \mathbf{I}(i,k)^T \log p_0$$
$$\Delta_{i,j} = 1, 2, \ldots, j-1, j+w, \ldots L \tag{2}$$

Where $I(i,j)$ is an indicator vector to indicate the corresponding letter among A, C, D, E, F…, Y at position j of sequence X_i. $\Delta_{i,j}$ is the set of non-motifs positions if the motif starts at position $X_{i,j}$.

The conditional probability of a length-W sub sequence being the background component or the motif component could be defined as

$$\log \Pr\left(X_{ij} | \ \theta_c\right) = \sum_{k=0}^{W-1} \mathbf{I}(i, j+k)^T \log p_{k'} \tag{3}$$

Where $k' = 0$ of $c = 0$ otherwise $k' = k + 1$ with $c = 1$.

The conditional probability of a sequence that does not contain motif under the ZOOPS model could be defined as

$$\log \Pr\left(X_i | \ Q_i = 0, \theta\right) = \sum_{k=0}^{L-1} \mathbf{I}(i, j+k)^T \log p_0 \tag{4}$$

4.2 E-step

The E-step is to calculate the expected value of Z.

$$Z_{i,j}^{(t)} = \frac{\Pr\left(X|\ Z_{i,j}=1,\theta^{(t)}\right)}{\sum_{j=1}^{m}\Pr\left(X|\ Z_{i,j}=1,\theta^{(t)}\right)} \tag{5}$$

4.3 M-step

The M-step of EM will re-estimate the PSSM matrix and background probability following the formula below.

$$p_k^{(t+1)} = \frac{c_k + d_k}{c_k + d_k}, 0 \le k \le W \tag{6}$$

$$c_k = \begin{cases} t - \sum_{j=1}^{W} c_j \\ \sum_{i=1}^{n}\sum_{j=1}^{m} Z_{i,j}^{(t)} \mathbf{I}(i, j+k-1)^T \end{cases} \tag{7}$$

Here, d_k could be seen as pseudo-counts to avoid zero probability in the updating process. It could also be viewed as a prior value that incorporates knowledge about motif columns, which will be discussed later.

4.4 Scoring

We could use the change of PSSM matrix P_k $(1 \le k \le W)$ as stop criterion, which is the practical criterion being implemented in MEME suite. The absolute value of difference of PWM with the last iteration should be zero upon convergence.

4.5 Motif Eraser

The section above summaries the steps used in a single pass of MEME, which is for detecting a single type of motif from the sequences. The most important algorithm is what I called the 'motif eraser', which will be applied after each motif for a particular width is found, this is called a pass of MEME. It will 'erase' the motif 'probabilistic-ally' as if it is not in the sequence. In theory, the approach used by MEME to find multiple motifs at different passes is the greedy search algorithm. Initially, MEME assumes $P_r(Z_{ij} = 1) = \lambda$ for all Z_{ij}. In the subsequent passes of MEME, a new prior on each Z_{ij} will be computed in the E step and takes into account a new motif of width W starting at each position might overlap the occurrences of the motifs found previously. Mathematically, to update the Z variable to include previous founded motif position, a variable V is introduced. For $i = 1, \ldots n$ and $j = 1, \ldots m$.

$$V_{ij} = \begin{cases} 1 & \text{if no old motifs in } \left[X_j, \ldots, X_{j+w-1}\right] \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

To compute V , a binary variable U - an indicator of which positions are not contained in occurrences of previously found motifs. For $i = 1, \ldots n$ and $j = 1, \ldots m$.

$$U_{ij} = \begin{cases} 1 & \text{if } X_{i,j} \notin \text{ previously founded motifs} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

This variable U is updated after each pass of MEME with the formula:

$$U_{ij}^{p} = U_{ij}^{p-1}\left(1 - \max_{k=j-W+1,\ldots,j} Z_{i,k}^{(t)}\right) \tag{10}$$

The maximum of Z_{ij} is used because Z_{ij} is not independent with each other since current motif cannot overlap themselves. The value of will be used as the value for $P_r(U_{i,j}=1)$ in the next pass $p+1$. Then we can estimate the $P_r(V_{i,j}=1)$ as

$$\tilde{Z}_{i,j}^{(t)} = Z_{i,j}^{(t)} \Pr\left(V_{i,j}=1\right) \tag{11}$$

Note that this new $\tilde{Z}$ will be used in place of $Z^{(t)}$ in the M step of EM process in equation (7) as well as updating U variable in equation (10).

4.6 Avoid local minima using Prior Information

4.6.1 'TEST' algorithm

The paper also proposes an algorithm to generate optimum starting θ_1 for running MEME. This algorithms will iterate all the possible subsequences for a given motif width and λ and estimate the soundness of θ_1 after one pass of EM using likelihood of the model using the conditional probability $P_r(X_{i,j}|\theta)$ defined in equation (2). Note that this single iteration of EM is different with the EM in the MEME. It is an approximated EM process and uses $P_r(X_{i,j}|\theta)$ to approximate the expected value of Z and uses the maximum likelihood (looking at the one start position for a sequence) of being motif occurrences to compute letter count to calculate θ_1, rather than 'softly' computing the joint likelihood and considering every possible staring position in the sequence. Thus, this approximated EM process is much faster. Furthermore, I utilized dynamic programming to implement the calculation of $P_r(X_{i,j}|\theta)$ in making it even faster and saves a huge amount of computation power. The detailed recursion rule could be found in the technique report from Bailey and Elkan, 1995[2]. Also, this algorithm is implemented to test several starting λ for a given W at once. The procedure of the algorithm is as follows. And I apply it into the condition of fixed width.

Algorithm 1: The process of TEST (W, list of λ values, dataset)

set $\theta_0 = \mu$, average letter frequencies in dataset
for each sequence X_k starting at position l in dataset **do**
  map subsequence $X_{k,l}$ to θ_1
  for each sequence X_i starting at position j in dataset **do**
    compute $P_r(X_{i,j}|\theta)$
  endfor
endfor
make list of single subsequence in each sequence
with maximum value of $P_r(X_{i,j}|\theta)$
for each value of λ in list **do**
  calculate c_k, $1 \le k \le W$, given top $nm\lambda$ subsequences in (sorted) list
  estimate θ_1 after one pass of EM as $\hat{\theta}_1 = [c_1, c_2, \ldots, c_W]$
  compute likelihood of model $(\theta_0, \hat{\theta}_1, \lambda)$
  save θ_1 if likelihood of $(\theta_0, \hat{\theta}_1, \lambda)$ best for this value of λ
endfor

5 Inference and Results

5.1 Inference

In this project, the inference stage of MEME of motif finding process is entirely based on the missing information Z. The missing information during the process is an expected variable so it requires setting a threshold to turn it into a 'hit' only if the normalized Z variable exceeds 0.5 rather than a simple 'argmax' operation. This will ensure more confidence to the predicted motif sequence. The original paper uses a slightly different approach which is worthy of mentioning here. Rather than using the intermediate result of Z information at each pass. It uses the PSSM matrix θ_1 to evaluate all the subsequences using a score function and a threshold based on λ and finding the start position with the best score among the positions that exceeds the threshold. This is quite interesting and it is more robust than entirely based on the Z variable. I will implement it if I have more time.

Algorithm 2: MEME

Input sequence X, $pass_{max}$
Running $TEST(X, a\ list\ of\ \lambda, W)$ and store θ
for $pass = 1$ **to** $\boldsymbol{pass_{max}}$ **do**
  Initialize θ by the stored value from 'TEST'
  Run EM till convergence form chosen $\phi = (\theta, \lambda, W)$
  Update the prior probabilities $U_{i,j}$ to approximate multiple motif model
endfor

5.2 Results

- Motif
  "ASCPVLTCGP"
- Position
  [40 23 0 57 8 73 49 85 29 46 82 4 49 79 71 23 78 41 0 43 69 82 76 3 25 5 36 82 34 74 30 89 15 54 42 49 84 66 71 22 23 11 51 31 77 45 30 83 48 7 80 45 60 33 84 82 86 23 79 22 21 12 68 69 81 38 68 64 89 65 47 58 53 12 56 30 12 72 16 85 57 42 86 81 30 8 6 65 32 82 24 53 11 26 15 75 44 41 56 7 27 47 34 63 46 9 32 35 13 27 88 78 86 76 26 26 74 44 51 7 29 87 46 64 64 72 13 5 57 19 80 10 68 1 28 64 33 82 56 88 63 41 57 31 30 29 8 51 26 29 31 58 69 84 47 1 55 88 34 31 65 66 7 7 89 76 60 58 37 73 84 87 18 6 18 14 83 1 60 16 68 42 31 10 47 8 57 32 52 51 32 59 64 2 72 26 66 58 7 5]
  As for the multiple motif, this is the other result which is the second match.
- Motif
  "IWFFMEEARD"
- Position
  [83 83 72 84 50 35 66 60 76 19 62 65 82 21 20 46 57 70 30 4 36 47 45 81 67 61 70 26 16 33 15 21 5 28 62 37 68 25 86 47 42 59 25 80 38 17 4 13 37 23 25 16 86 12 11 37 21 61 9 86 59 72 22 15 59 75 11 76 8 6 35 87 43 57 76 13 22 86 45 7 45 86 33 40 76 58 84 82 5 18 80 41 44 68 2 43 90 54 78 31 12 36 7 8 8 84 71 4 2 65 58 21 53 1 78 60 54 86 21 60 10 77 60 7 40 82 32 35 19 87 50 32 58 19 84 44 58 25 3 65 84 70 37 65 54 62 37 23 59 89 45 41 54 22 8 85 13 68 55 4 20 82 44 72 38 13 5 76 14 53 69 17 41 17 49 62 40 18 87 4 52 81 7 68 78 44 35 85 39 70 53 69 18 44 42 11 22 1 66 50]

References

[1] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in bipolymers," 1994.
[2] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine learning,* vol. 21, no. 1, pp. 51-80, 1995.