Supplementary: Simultaneously Learning DNA Motif along with Its Position and Sequence Rank Preferences through EM Algorithm

ZhiZhuo Zhang,¹ Cheng Wei Chang,² Willy Hugo,¹ Edwin Cheung,² Wing-Kin Sung,¹,²

¹National University of Singapore & ²Genome Institute of Singapore
¹ {zhizh, hugowill, ksung}@comp.nus.edu.sg & ²{changcw99, cheungcwe}@gis.a-star.edu.sg
1 Supplementary Result

1.1 EEM estimates the correct motif length
One of the strong point of SEME is that user need not establish any motif length (which is, in most cases, hard to
estimate). As shown in Figure 1, for most of the cases, the EEM procedure estimated motif lengths that are very
close to the planted motif length. We further observed that, when the estimated PWM length differs, EEM tends to
underestimate the length because most motifs in JASPAR contains very weak signal in their flanking positions and
EEM will exclude them to avoid over-fitting the training data.

1.2 REM recalls the majority of the actual motif sites
We want to study the sampling efficiency of re-sampling EM procedure of SEME. For each dataset and for each
sampling ratio $\mu$, we trace back the subsampled sites in the re-sampling EM procedure, and check how many true
planted sites were included in the subsampled set. Under the same sampling ratio, we define a term “efficiency ratio”
to be the ratio between the number of true planted sites in the biased subsampled set and that in the uniform subsampled
set. In Figure 2, the red line shows the efficiency ratio changes from 600 to 2 when the sampling ratio $\mu$ changes to
$2^{-10}$ to $2^{-1}$. It shows great efficiency ratio when the sampling ratio is low, because SEME do subsampling using
the output PWM from Extending EM procedure which has much higher chance to sample a true sites than naive
uniform sampling. On the other hand, we observe that most true sites can be sampled even in the low sampling
ratio, so increasing the sampling ratio is almost the same as increasing the background ratio, which makes efficiency
ratio drop dramatically. To illustrate this, we can check the average recall rate (blue line in Figure 2) across different
sampling ratio, and it shows near 60% recall rate at sampling ratio $2^{-10} \approx 0.001$ and 90% recall rate at sampling ratio
$2^{-5} \approx 0.031$. The error bar in Figure 2 presents the interval +/- one standard deviation from the average recall rate,
and we can see that higher sampling ratio can bring smaller variances of recall rate. To balance trade-off between the
efficiency ratio and coverage, we fix default sampling ratio=0.01 in the later experiments of this paper.
Fig. 1: **Comparison of EEM estimated motif length and actual motif length** The estimation of PWM length by the EEM procedure closely matches the actual planted motif. When the planted PWM have degenerate positions on its flanking positions, EEM will predict a shorter PWM which excludes the latter.

Fig. 2: **Efficiency Ratio and Recall Rate across different sampling ratios.** We apply different sampling ratios to run SEME on 75 simulation datasets. The left y axis is value of the average efficiency ratio of SEME biased sampling against the uniform sampling. The right y axis is the average recall rate of true planted sites. The error bar presents the region +/- one standard deviation of recall rate.
1.3 The performances of EEM and REM

We summarize the SEmE performance by comparing the PWMs output in each EM phase and planted JASPAR PWMs. As shown in Figure 3, EEM’s output PWMs have significantly better discriminative capability and similarity to the real planted PWM than the random PWM generated by random shuffled the columns in the EEM’s PWMs. That is, EEM not only estimates correct motif length but also recover the motif signals, although the signal is still imperfect. Figure 3 (left side) shows the PWMs after REM refinement are much more similar (less PWM divergence) to the planted PWMs comparing to the PWMs outputted by EEM PWMs. And Figure 3 (right side) further shows the REM outputted PWMs indeed already have very close discriminative capability as the planted PWM and also are better than the EEM outputted PWMs.

![Recovered Motif Similarity](image1)

**Fig. 3:** The PWMs outputted by EEM and REM compared to actual PWM. (Left) The similarity between recovered PWM and planted PWM. The EEM outputted PWMs (blue line) with mode around PWM divergence=0.1 are significantly more similar to planted PWM than the random generated PWMs (green line) which have mode around 0.23. Also, the REM outputted PWMs (red line) have much better similarity (less divergence) to the planted PWMs comparing to EEM outputted PWMs. (Right) Comparison of Discriminative Capability. The EEM outputted PWMs (blue line) with mode around AUC score=0.9 are significantly better than the random generated PWMs (green line) which have mode around AUC score=0.5. The REM outputted PWMs (red line) have very close AUC score distribution as the planted PWMs (black line) and better performance than EEM outputted PWMs.

1.4 Detailed Result on Metazoan Compendium Dataset

The detail result of this comparison is shown in Figure 4. Using the same criteria of success detection (PWM divergence < 0.18)[7], we find that SEmE successfully detected the correct primary motifs in 21 datasets whereas Amadeus succeeded in 18 datasets, Weeder and Trawler succeeds in 11 and 12 datasets respectively. Moreover, the result showed SEmE can find more accurate motifs (12 motifs with PWM divergence less than 0.12, comparing to 9 motifs for the second best program Amadeus). SEmE further detected a significant position preference for the correct motif for many datasets in this benchmark: most of them tend to bind nearer to the TSS position (Figure 5).

2 Data Preparation

2.1 Known Motif Databases

Two known motif databases are used in our paper: JASPAR vertebrate core database with total 123 PWM motifs[8] and Transfac database with total 830 PWM motifs (including non-vertebrate species)[9].

2.2 Simulation Data

Seventy-five PWMs with length longer than 9 were extracted from JASPAR vertebrate core database. For each PWM motif, we generated a training data for the algorithms to learn the motif and a testing data to test its performance. The
Fig. 4: Comparison of de-novo motif discovery tools on the metazoan compendium. Each column of the table presents the results for one motif discovery tool, and each row corresponds to one data set of the metazoan compendium. The color of the checkmarks represents the accuracy of the motif discovered as measured by the normalized euclidean distance, and we use the thresholds on the PWM divergence as proposed by Linhart et al[7]. The symbol $\infty$ marks long execution times (h) that were aborted in[7]. In the last row of the table, we report the total number of motifs discovered by each of the tools.
Fig. 5: **SEME detected TF motifs with significant position preference to TSS**

Seven examples of SEME’s output of metazoan compendium dataset. The result indicates these TF binding sites are enriched near the transcription start sites. The TSS position is located around 200bp from the rightmost position. The original 1200bp promoter sequences may be shortened after removing “N”-masked regions, so the TSS position may be shifted in those cases.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>De novo Motif Logo</th>
<th>SEME estimated Motif Position Distribution in Promoter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human.ESF4_Com.203</td>
<td><img src="image1.png" alt="Image" /></td>
<td><a href="graph1.png">Graph</a></td>
</tr>
<tr>
<td>Fly.Myc_Chrx.723.mapped</td>
<td><img src="image2.png" alt="Image" /></td>
<td><a href="graph2.png">Graph</a></td>
</tr>
<tr>
<td>Human.CREB_Phang.2354</td>
<td><img src="image3.png" alt="Image" /></td>
<td><a href="graph3.png">Graph</a></td>
</tr>
<tr>
<td>Human.ETS1_Hollenhorst.1191</td>
<td><img src="image4.png" alt="Image" /></td>
<td><a href="graph4.png">Graph</a></td>
</tr>
<tr>
<td>Human.INSa_Odom.1485</td>
<td><img src="image5.png" alt="Image" /></td>
<td><a href="graph5.png">Graph</a></td>
</tr>
<tr>
<td>Human.NF1_Com.679</td>
<td><img src="image6.png" alt="Image" /></td>
<td><a href="graph6.png">Graph</a></td>
</tr>
<tr>
<td>Human.YY1_XiRen.721</td>
<td><img src="image7.png" alt="Image" /></td>
<td><a href="graph7.png">Graph</a></td>
</tr>
</tbody>
</table>
training dataset consists of 1000 random sequences of length 400bp where 500 of them were planted with binding sites of the PWM motif. The binding site sequences were generated randomly according the PWM probability and planted uniformly across different positions and sequence rank which ensures no additional information provided. For testing, we generated one positive and one negative dataset. The positive one consists of 1000 random sequences (length=400bp) and each of them was planted with one binding site sequence generated from the selected JASPAR PWM, which is similar to the training data except that all the positive sequences contains one planted binding site. For negative data, it consists of 1000 random sequences (length=400bp) without any planted binding site.

2.3 Metazoan Compendium Data

A metazoan compendium was published by Linhart et.al[7], which consisting of 32 experimental datasets based on microarray, ChIP-chip, ChIP-DSL, and DamID as well as Gene Ontology data[1]. A list of the promoter sequences of many target genes (1000bp upstream and 200bp downstream the Transcription Start Site (TSS)) are used as the positive input for each motif-finding program and promoter sequences of other non-target genes are used as background sequences. The dataset was downloaded from http://acgt.cs.tau.ac.il/amadeus/download.html.

2.4 ChIP-seq Benchmark Data

We collected 164 published ChIP-seq libraries from ENCODE project and our lab over different cell-lines and transcription factors. The ChIP-seq narrow peak files (provided by Yale group) for Encode project can be downloaded from http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeYaleChIPseq/.

The ChIP-seq data published by our lab can be accessed by the NCBI accession numbers as follows:

<table>
<thead>
<tr>
<th>ChIP-seq Dataset</th>
<th>Peak Number</th>
<th>Accession number</th>
<th>reference genome</th>
</tr>
</thead>
<tbody>
<tr>
<td>LncapAR_DHT</td>
<td>10000</td>
<td>GSM699631</td>
<td>hg18</td>
</tr>
<tr>
<td>LncapFoa1_DHT</td>
<td>56304</td>
<td>GSM699635</td>
<td>hg18</td>
</tr>
<tr>
<td>Mcf7ER_e2</td>
<td>17442</td>
<td>GSM589237</td>
<td>hg18</td>
</tr>
<tr>
<td>Mcf7Foa1_e2</td>
<td>18436</td>
<td>GSM588930</td>
<td>hg18</td>
</tr>
<tr>
<td>mm8ESloci_cmyc</td>
<td>3422</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_ctcf</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_e2f1</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_esrbbredo</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_klf4</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_nanog</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_nmyc</td>
<td>7182</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_oct4</td>
<td>3761</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_p300</td>
<td>524</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_smad1</td>
<td>1126</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_sox2</td>
<td>4526</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_stat3</td>
<td>2546</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_suz12</td>
<td>4215</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_tcfcp211</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
<tr>
<td>mm8ESloci_zfx</td>
<td>10000</td>
<td>GSE11431</td>
<td>mm8</td>
</tr>
</tbody>
</table>

2.5 Known Co-TFs List

For each ChIPed TF, the table below contains a list of its co-TFs’ motifs. The list of co-TFs for ES cell dataset, can be found in [10]; The list of co-TFs for LNCaP AR dataset can be found in [5, 3]; The list of co-TFs for MCF7 ER dataset can be found in [3].
3 Supplementary Method

3.1 Review of Mixture Model for Motif Finding

Applying mixture model to learn motifs in a set of sequences is first proposed by MEME[2]. It assumes the observed sequences are generated by two independent components: motif model and background model. Let the alphabet be \{a_1, \ldots, a_L\}. The background model is a length-L probability vector \( \theta_0 = (\theta_{0,1}, \ldots, \theta_{0,L}) \) where \( \theta_{0,k} \) is the probability of observing \( a_k \). The motif model describes a length-W sequence as \( \Theta \), which is a \( L \times W \) matrix where \( \Theta_{j,k} \) is the probability that the nucleotide \( a_k \) occurs at position \( j \). For any length-W sequence \( X_i \), the probability that \( X_i \) is generated from the motif model and the background model can be computed as follows.

\[
Pr(X_i|Z_i = 1) = Pr(X_i|\Theta) = \prod_{j=1}^{W} \prod_{k=1}^{L} \Theta_{j,k}^{I(a_k,X_{i,j})} \tag{1}
\]

\[
Pr(X_i|Z_i = 0) = Pr(X_i|\theta_0) = \prod_{j=1}^{W} \prod_{k=1}^{L} \theta_{0,k}^{I(a_k,X_{i,j})} \tag{2}
\]

where \( X_{i,j} \) is the letter in \( j \)-th position of sample \( X_i \) and \( I(x,y) \) is an indicator function which is 1 if and only if \( x = y \).

Any set of sequences can be conceptually split into a set \( X = \{X_1, X_2, \ldots, X_n\} \) of \( n \) overlapping subsequences of length W. MEME assumes those length-W subsequences in \( X \) are extracted from a mixture of motif model \( \Theta \) and a background model \( \theta_0 \), where \( \lambda (0 < \lambda < 1) \) is the parameter which defines the prior probability of \( X_i \) generated by motif model. The probability framework of the mixture model is defined as follows:

\[
Pr(X) = \prod_{i=1}^{n} (\lambda Pr(X_i|Z_i = 1) + (1 - \lambda) Pr(X_i|Z_i = 0)) \tag{3}
\]

Then MEME formulated the motif finding problem as an optimization problem, which finds a set of parameters \((\lambda, \theta_0, \Theta)\) which maximizes the likelihood of data \( Pr(X) \). This optimization problem is NP-hard. EM algorithm is a state of the art method to solve this maximum likelihood problem. The EM algorithm makes use of the concept of missing data. In this case, the missing data \( Z_i \) is the knowledge of which component each sample \( X_i \) come from, and \( Z_i = 1 \) if \( X_i \) is from motif model; and \( Z_i = 0 \) otherwise. Also by definition, \( Pr(Z_i = 1) = \lambda \). The objective function of EM can be revised as a “complete log likelihood function”:

\[
\log Pr(X, Z|\lambda, \theta_0, \Theta) = \sum_{i=1}^{n} (Z_i \log(\lambda Pr(X_i|\Theta)) + (1 - Z_i) \log((1 - \lambda) Pr(X_i|\theta_0))) \tag{4}
\]
The EM algorithm iteratively maximizes the expected log likelihood over the conditional distribution of missing data \( \{Z_i\} \) given the current estimation of parameters \( \{\lambda, \theta_0, \Theta\} \). In the E-step, the expected value of \( Z_i \) in the iteration \( t \) can be computed as:

\[
Z_i^{(t)} = \frac{\eta_i^{(t)}}{1 + \eta_i^{(t)}},
\]

where \( \eta_i \) is the likelihood ratio between the motif model and the background model.

\[
\eta_i^{(t)} = \frac{\lambda(\theta^{(t-1)})}{(1 - \lambda(\theta^{(t-1)}))} \frac{Pr(X_i | \theta^{(t-1)})}{Pr(X_i | \theta_0^{(t-1)})}
\]

In the M-step, the parameters are estimated to maximize the expected log likelihood function given the expected value \( \{Z_i\} \) in the last iteration:

\[
\{\lambda^{(t)}, \theta_0^{(t)}, \Theta^{(t)}\} = \underset{\lambda, \theta_0, \Theta}{\text{arg max}} \mathbb{E}_{Z^{(t)}} \left[ \log Pr(X, Z^{(t)} | \lambda, \theta_0, \Theta) \right]
\]

and we can compute the explicit formulas for each parameter.

\[
\lambda^{(t)} = \frac{\sum_{i=1}^{n} Z_i^{(t)}}{n}
\]

For \( k = 1, \ldots, L \) and \( j = 1, \ldots, W \), we have

\[
\theta_0^{(t), k} = \frac{\sum_{i=1}^{n} \sum_{w=1}^{W} (1 - Z_i^{(t)}) I(k, X_{i,w})}{\sum_{i=1}^{n} \sum_{w=1}^{W} (1 - Z_i^{(t)})}
\]

\[
\theta_j^{(t), k} = \frac{\sum_{i=1}^{n} Z_i^{(t)} I(a_k, X_{i,j})}{\sum_{i=1}^{n} Z_i^{(t)}}
\]

In SEME implementation, we consider two more binding preferences: position and sequence rank in addition to DNA sequence preference information in the traditional EM algorithm. The position preference tries to model if the binding site prefers certain positions. We discretize the positions into \( K \) bins. The probability a binding site occurs in the \( k \)-th position bin is denoted as \( \alpha_k \), for \( k = 1, \ldots, K \), while the background distribution is assumed to be uniform. Precisely, for every \( X_i \), we have:

\[
Pr(X_i^{(pos)} = k | Z_i = 1) = \alpha_k; Pr(X_i^{(pos)} = k | Z_i = 0) = \frac{1}{K}
\]

Similarly, the sequence rank preference tries to model if the binding site prefers the sequences with certain range of ranks assuming input sequences are sorted by some measurement. We discretize the ranks into \( K \) bins as position preference. The probability a binding site occurs in the \( k \)-th rank bin is denoted as \( \beta_k \), for \( k = 1, \ldots, K \), while the background distribution is assumed to be uniform. Precisely, for every \( X_i \), we have:

\[
Pr(X_i^{(rank)} = k | Z_i = 1) = \beta_k; Pr(X_i^{(rank)} = k | Z_i = 0) = \frac{1}{K}
\]

We use naive bayesian approach to model three types of information (sequence, position, rank)

\[
Pr(X_i | Z_i) = Pr(X_i^{(seq)} | Z_i) Pr(X_i^{(pos)} | Z_i) Pr(X_i^{(rank)} | Z_i)
\]

where the probability of sequence information for bound state and unbound state \( Pr(X_i^{(seq)} | Z_i) \) can be referred to Equations 1 and 2.
Similar to Equation 7, the “complete log-likelihood function” with additional binding preferences can be modified as follows:

\[
\log Pr(X, Z | \Phi) = \sum_{i=1}^{n} \left( Z_i \log(\lambda \prod_{j=1}^{L} \prod_{k=1}^{W} \theta_{j,k}^{I(a_k,X^{(pos)}_i)}) \alpha_{X^{(pos)}_i} \beta_{X^{(rank)}_i} + (1 - Z_i) \log((1 - \lambda) \prod_{j=1}^{L} \prod_{k=1}^{W} \theta_{0,k}^{I(a_k,X^{(pos)}_i)})/K^2) \right)
\]

(12)

where \( \Phi = (\lambda, \Theta, \theta_0, \alpha_1, ..., \alpha_K, \beta_1, ..., \beta_K) \) are the parameters of mixture model in SEME.

Similarly, EM algorithm can be applied to optimize Equation 14.

In the E-step, the likelihood ratio between the model model and the background model is:

\[
\eta_i^{(t)} = \frac{\lambda \prod_{j=1}^{L} \prod_{k=1}^{W} \theta_{j,k}^{I(a_k,X^{(pos)}_i)}) \alpha_{X^{(pos)}_i} \beta_{X^{(rank)}_i} / K}{1 - \lambda \prod_{j=1}^{L} \prod_{k=1}^{W} \theta_{0,k}^{I(a_k,X^{(pos)}_i)}) / K}
\]

(13)

and the expected value of \( Z_i^{(t)} \) can be computed using Equation 5.

In the M-step, the parameters \( \Phi = (\lambda, \Theta, \theta_0, \{\alpha_1, ..., \alpha_K\}, \{\beta_1, ..., \beta_K\}) \) can be estimated by maximizing the expected log likelihood function given the expected value \( Z_i \) in the last iteration.

\[
\Phi^{(t)} = \arg \max_{\Phi} E_{Z^{(t)}} \left[ \log Pr(X, Z^{(t)} | \Phi) \right]
\]

(14)

The parameters \( (\lambda, \Theta, \theta_0) \) can be updated using Equations 8, 9 and 10, respectively. The additional parameters for position preference and sequence rank preferences can be updated as follows:

\[
\forall k \in \{1, ..., K\}, \alpha_k^{(t)} = \frac{\sum_{X_i \in X} Z_i^{(t)} I(k, X_i^{(pos)})}{\sum_{X_i \in X} Z_i^{(t)}}
\]

(15)

\[
\forall k \in \{1, ..., K\}, \beta_k^{(t)} = \frac{\sum_{X_i \in X} Z_i^{(t)} I(k, X_i^{(rank)})}{\sum_{X_i \in X} Z_i^{(t)}}
\]

(16)

Above is the general probabilistic framework of SEME, however, it cannot achieve good efficiency and accuracy for practical use if we directly apply classic EM algorithm to solve this problem. Hence, we developed four phases in the SEME pipeline: 1. Identify over-represented short l-mer; 2. Extending short l-mer to full length PWM motif (extending EM); 3. Optimizing extended motif and other modeling parameters (resampling EM); 4. Sorting discovered motifs by pre-defined scoring and filtering the redundant ones. Below, we describe each phase in SEME pipeline.

### 3.2 Identifying Over-represented l-mers

In the first phase, SEME computes the frequencies of all the short l-mers \((l = 5\) by default\) in the input sequences, and also their frequencies in background if control sequences or background model are provided. Then, all the short l-mers have higher frequencies in the input sequences than the background will be outputted to the next phase for further processing. If no background or control sequences are provided, 1st-order markov model will be estimated from the input sequences as the background model.

### 3.3 Extending EM Procedure

The original EM algorithm does not allow varying the length of PWM within the EM iteration. Assume we know the motif contains a conserved short l-mer seed \( q \) (obtained from the first phase), this section developed the extending EM (EEM) which can extend the length-l seed while maximizes the likelihood of the observed data. Assume the maximum length of the motif is \( W_{max} \). From the set of input sequences, we extract a set of length-(2\( W_{max} - |q|\)) sequences \( X_q = \{X_1, X_2, ..., X_n\} \) whose middle part is \( q \), i.e., \( X_i|W_{max} - |q|, W_{max} - 1| = q \). For example,
if the l-mer is "GGTCA" and the longest possible motif length is 10, \( X_q \) are all the sites matching string pattern "NNNNNGGTCAANNNNN". By the definition of \( X_q \), we can consider all potential binding sites which contain the short conserved l-mer \( q \) with the length less than the predefined length using threshold \( W_{max} \).

Similar to the original EM method, we first define a wide PWM model (may contain non-binding site positions, but is wide enough to cover all the potential binding site positions), \( \Theta \) is \( L \times 2W_{max} - |q| \) matrix and background model \( \overline{\theta}_0 \), and two variables \( l_1, l_2 \) for indexing the real binding site start and end positions in \( \Theta \). In each EM iteration, a subset of columns in the wide PWM model \( \Theta \) will be used to compute the expectation, and the column is included only if it can increase the likelihood in the M-step and show significant difference to background distribution. Let \( \Theta_{[l_1,l_2]} = \{ \Theta_{l_1}, \Theta_{l_1+1}, \ldots, \Theta_{l_2} \} \). The computation for modeling sequence information will be carried on a subset of position in the sites, that is, the positions outside of \([l_1,l_2]\) will not be used and the positions for the given l-mer also will not be used because that l-mer positions are same across \( X_q \).

Here, we have, \( \forall X_{iu} \in X_q \)

\[
Pr(X_{iu}^{(seq)}|\Theta_{[l_1,l_2]}) = \prod_{j=l_1\ldots l_2\land j \notin [W_{max}-|q|,W_{max}-1]}^{L} \prod_{k=1}^{L} (\Theta_{j,k})^{I(a_k,X_{iu,j})} \tag{17}
\]

\[
Pr(X_{iu}^{(seq)}|\overline{\theta}_0) = \prod_{j=l_1\ldots l_2\land j \notin [W_{max}-|q|,W_{max}-1]}^{L} \prod_{k=1}^{L} (\theta_{0,k})^{I(a_k,X_{iu,j})} \tag{18}
\]

And, the position model and sequence rank model remains the same as Equations 15 and 16. Then, we define one iteration of the extending EM procedure as follow:

In the E-step, similar to Equation 13, we compute the likelihood ratio \( \eta_{iu}^{(t)} \) as,

\[
\eta_{iu}^{(t)} = \frac{\lambda^{(t-1)} Pr(X_{iu}^{(seq)}|\Theta_{[l_1,l_2]}) \alpha^{(t-1)} X_{iu}^{(seq)} \beta^{(t-1)} X_{iu}^{(rank)}}{(1 - \lambda^{(t-1)}) Pr(X_{iu}^{(seq)}|\overline{\theta}_0^{(t-1)}) / K^2} \tag{19}
\]

and then the expectation value of \( Z_{iu}^{(t)} \) as,

\[
Z_{iu}^{(t)} = \frac{\eta_{iu}^{(t)}}{1 + \eta_{iu}^{(t)}} \tag{20}
\]

In the M-step, the modeling parameters \((\lambda^{(t-1)}, \overline{\theta}_0^{(t-1)}, \Theta^{(t)}, \{\alpha_1, \alpha_2, \ldots, \alpha_K\}, \{\beta_1, \beta_2, \ldots, \beta_K\})\) are updated using Equations 8, 9, 10, 15 and 16, respectively, which are exactly the same as the original EM algorithm except considering \( X_q \) instead of \( X \). Moreover, the two index variables \( l_1, l_2 \) will also be updated in this step by trying to select a column outside \([l_1,l_2]\) to maximize the log likelihood objective function. In another point of view, there is no change for position and sequence rank preferences by including one more column, and the effect for sequence preference of including one more column is the same as replacing a background column to a new column value. Because we can assume the new added column already exists but its value is the same as background, so it brings no effect in the computation before. The increase of log likelihood by considering the \( j \)-th column is defined as:

\[
\Delta_j^{(t)} = \sum_{k=1}^{L} \sum_{X_{iu} \in X_q} I(a_k, X_{iu}^{(seq)}, j) Z_{iu}^{(t)} (\log(\Theta_{j,k}) - \log(\theta_{0,k}^{(t)})), j \notin [l_1,l_2] \tag{21}
\]

In a greedy manner, the extending EM procedure chooses the column \( j \) with the largest \( \Delta_j^{(t)} \), and tests whether the alphabet distribution of that column is significantly different from the background model. Let \( p = \arg \max_j \Delta_j^{(t)} \), the Chi-square statistic \( \chi \) is defined as:

\[
\chi = \frac{\sum_{k=1}^{L} \sum_{X_{iu} \in X_q} I(a_k, X_{iu}^{(seq)} Z_{iu}^{(t)} (\Theta_{p,k}^{(t)} - \theta_{0,k}^{(t)})^2}{\sum_{X_{iu} \in X_q} I(a_k, X_{iu}^{(seq)} Z_{iu}^{(t)}(\theta_{0,k}^{(t)})^{-2}} \tag{22}
\]
Procedure 1 Extending EM

Input: l-mer \( q \), longest motif length \( W_{\text{max}} \), the original data \( X \)
Output: final extended PWM \( \Theta^{(t)}_{[l_1,l_2]} \)

1: \( X_q := \) the set of sites in \( X \) of length \( 2W_{\text{max}} - |q| \) and containing \( q \) in the middle ;
2: Initialize \( (\lambda^{(0)}, l_1^{(0)}, l_2^{(0)}, \Theta^{(0)}, \delta_0^{(0)}, \{\alpha_1^{(0)}, \ldots, \alpha_K^{(0)}\}, \{\beta_1^{(0)}, \ldots, \beta_K^{(0)}\}) \) according to Equations 23–28
3: \( t := 1 \)
4: repeat
5: \( \text{E-step: } \forall X_{tu} \in X_q, \) compute \( \varphi^{(t)}_{tu} \)
6: \( t := t + 1 \);
7: \( \text{M-step: update } (\lambda^{(t)}, \Theta^{(t)}, \delta_0^{(t)}, \{\alpha_1^{(t)}, \ldots, \alpha_K^{(t)}\}, \{\beta_1^{(t)}, \ldots, \beta_K^{(t)}\}) \) according to Equations 8, 9, 10, 15 and 16
8: if \( |l_1^{(t-1)} - l_2^{(t-1)}| < W_{\text{max}} \) then
9: \( \text{select the best column } p = \arg \max_{\Delta_j^{(t)}} \Delta_j^{(t)} \) with Equation 21
10: \( \text{if column position } p \text{ distribution is significantly different with } \delta_0^{(t)} \text{ using Chi-square test then} \)
11: \( l_1^{(t)} = \min(l_1^{(t-1)}, p), l_2^{(t)} = \max(l_2^{(t-1)}, p); \)
12: \( \text{end if} \)
13: \( \text{end if} \)
14: until no change for \( l_1, l_2 \) and \( \Theta^{(t)} \) converge;
15: The columns in \( \Theta^{(t)}_{[W_{\text{max}}-q], W_{\text{max}}-1} \) are further diluted.
16: return \( (\Theta^{(t)}_{[l_1,l_2]}, \{\alpha_1^{(t)}, \ldots, \alpha_K^{(t)}\}, \{\beta_1^{(t)}, \ldots, \beta_K^{(t)}\}); \)

\[
\lambda^{(0)} = 1 - \frac{|X|Pr(q|B)}{|X_q|} \tag{23}
\]

where \( B \) is the background model provided by the user.

\[
l_1^{(0)} = W_{\text{max}} - |q|, l_2^{(0)} = W_{\text{max}} - 1 \tag{24}
\]

\[
\forall j = 1, \ldots, W; k = 1, \ldots, L, \Theta^{(0)}_{j,k} = \sum_{X_{tu} \in X_q} \frac{I(a_k, X_{tu,j}^{(\text{seq})})}{|X_q|} \tag{25}
\]

\[
\forall k = 1, \ldots, L, \delta_0^{(0)}_{a,k} = \sum_{X_{tu} \in X_q} \frac{I(a_k, X_{tu,j}^{(\text{seq})})}{|X|} \tag{26}
\]

\[
\forall k = 1, \ldots, K, \alpha^{(0)}_k = \sum_{X_{tu} \in X_q} \frac{I(k, X_{tu,j}^{(\text{pos})})}{|X_q|} \tag{27}
\]

\[
\forall k = 1, \ldots, K, \beta^{(0)}_k = \sum_{X_{tu} \in X_q} \frac{I(k, X_{tu,j}^{(\text{rank})})}{|X_q|} \tag{28}
\]

As a final step, the middle of the motif which contains the l-mer \( q \) is further diluted. We perform this step since it is not unrealistic that these \( l \) positions are absolutely conserved. Let \( l(j) \) denote the position in the \( \Theta^{(t)}_{[l_1,l_2]} \) occupied by the l-mer \( q \) position \( j \). This final step will set, \( \Theta_{l(j),a} = 0.5 \) if \( I(a, q_j) = 1 \); \( 0.5/(L-1) \), otherwise for \( j = W_{\text{max}} - l, \ldots, W_{\text{max}} - 1 \).
3.4 Re-sampling EM Procedure

In EEM, SEME finds a rough motif model with proper motif length. The motif can be further refined using classic EM algorithm[2] to improve the accuracy. However, when the input data $X$ is big, this step is slow. With the idea of importance sampling, we proposed the re-sampling EM (REM) procedure which reduces the running time by running EM algorithm on a subsample of the original data $X$.

Let $Q(\cdot)$ be the sampler function, where $Q(x) = 1$ if $x$ is sampled; and 0 otherwise. When $Q(\cdot)$ is a uniform random sampler, this approximation is trivial and we can directly use the original EM method and formulas in the sampled dataset in this case. Here, we generalize the formulas of EM to an arbitrary sampler $Q(\cdot)$, which satisfies $Pr(Q(x) = 1) > 0, \forall x \in X$.

**Theorem 1.** Let $X_Q = \{X_{i_1}, \ldots, X_{i_N}\}$ be a subset sampled from the original dataset $X$ using the sampler function $Q(\cdot)$, then,

\[
E_{X_Q} \left[ \sum_{X_{i_n} \in X_Q} \frac{\log Pr(X_{i_n}, Z_{i_n}|\Phi)}{Pr(Q(X_{i_n}) = 1)} \right] = \sum_{X_i \in X} \log Pr(X_i, Z_i|\Phi)
\]

where $E_{X_Q} \left[ \sum_{X_{i_n} \in X_Q} \frac{\log Pr(X_{i_n}, Z_{i_n}|\Phi)}{Pr(Q(X_{i_n}) = 1)} \right]$ is the expected value of $\sum_{X_{i_n} \in X_Q} \frac{\log Pr(X_{i_n}, Z_{i_n}|\Phi)}{Pr(Q(X_{i_n}) = 1)}$ over all possible subset $X_Q$.

**Proof.** According to sampling property:

\[
E_{X_Q}[Q(x)] = 1 \cdot Pr(Q(x) = 1) + 0 \cdot Pr(Q(x) = 0) = Pr(Q(x) = 1)
\]

Then the proof is straightforward. For each site, the sampling process is independent. Hence, the expectation of the summation value of a subsampled set $X_Q$ can be broken down to the expectation of contribution of each site $X_i \in X$ to the summation value.

\[
E_{X_Q} \left[ \sum_{X_{i_n} \in X_Q} \frac{\log Pr(X_{i_n}, Z_{i_n}|\Phi)}{Pr(Q(X_{i_n}) = 1)} \right] = \sum_{X_i \in X} E_{X_Q}[Q(X_i)] \cdot \frac{\log Pr(X_i, Z_i|\Phi)}{Pr(Q(X_i) = 1)} = \sum_{X_i \in X} \log Pr(X_i, Z_i|\Phi)
\]

where $\Phi = (\lambda, \Theta, \beta_0, \alpha_1, \ldots, \alpha_K, \beta_1, \ldots, \beta_K)$ are all the modeling parameters in our mixture model. □

According to Theorem 1 and the large sample theory[6], we can expect to get the same log likelihood value as Equation 14 by weighting each subsequence $X_{i_n}$ in the sampled dataset $X_Q$ with $\frac{1}{Pr(Q(X_{i_n}) = 1)}$, when the sample size $|X_Q|$ is large enough. Therefore, Equation 14 can be approximated as:

\[
\Phi(t) = \arg \max_{\Phi} E_{Z^{(t)}} \left[ \sum_{X_{i_n} \in X_Q} \frac{\log Pr(X_{i_n}, Z_{i_n}^{(t)}|\Phi)}{Pr(Q(X_{i_n}) = 1)} \right]
\]

Interestingly, no matter how we choose the sampler function $Q(\cdot)$, the maximum likelihood estimation always converges to the original one, when the sample size is large enough. However, noted that the number of parameters in the motif model is much larger than that in the background model, which means if we want to learn a motif model as good as the background model, it needs more samples from binding sites than from background sites. In the real dataset, the prior probability of binding site $\lambda$ is usually very small(less than 0.01). This motivates us to perform biased sampling, i.e., we want to use a sampler function which samples more binding sites. Here, we define our sampler to sample subsequences according to the PWM model outputted by extending EM $\Theta^{(EEM)}$, that is:

\[
Pr(Q(x) = 1) = \min(L^w \cdot \mu \cdot Pr(x|\Theta^{(EEM)}), 1), \forall x \in X
\]

where $\mu$ is the sub-sampling ratio defined by the user, and $w$ is the motif length.

Here is the rationale behind Equation 33. We want to control the final sample size roughly $\mu \cdot n$, where $n$ is the total number of sites. For length $w$ sequences, there are $L^w$ possibilities, and if we use the $Q(\cdot)$ above to sample all these $L^w$ w-mers, the expected number of sampled sites is

\[
\sum_{x \in \{a_1, \ldots, a_L\}^w} L^w \cdot \mu \cdot Pr(x|\Theta^{(EEM)}) = \mu \cdot L^w \cdot \sum_{x \in \{a_1, \ldots, a_L\}^w} Pr(x|\Theta^{(EEM)}) = \mu \cdot L^w
\]
Therefore, if the original dataset \(X\) with size \(n\) is formed by a uniform subset of those unique \(L^w\) w-mers, we can expect the size of \(X_Q\) is \(\mu \cdot n\).

Below, we describe the implementation detail for re-sampling EM (REM) procedure. First, for the E-step, it is almost the same as the original EM except that we adding two boolean parameters (\(\tau_{\text{pos}}\) and \(\tau_{\text{rank}}\)) to indicate whether the computation should consider the position model and the sequence model or not.

\[
\eta_u^{(t)} = \frac{(\lambda \prod_{j=1}^{L} \prod_{k=1}^{L} ( \Theta_{j,k}^{(t-1)} I(ak, X_i^{(seq)}_u)) \prod_{k=1}^{L} \theta_0^{(t-1)} I(ak, X_i^{(seq)}_u)))}{(1-\lambda) \prod_{j=1}^{L} \prod_{k=1}^{L} ( \Theta_{j,k}^{(t-1)} I(ak, X_i^{(seq)}_u)))}
\]

The motivation of introducing the indicator variables (\(\tau_{\text{pos}}\) and \(\tau_{\text{rank}}\)) is to avoid over-fitting the data in the final model by assuming the position bias and the sequence rank bias must exist. These two indicator variables will be updated in the M-step in each iteration and will be set to 1 only if the expected binding sites distribution is significantly different to uniform distribution (i.e., background distribution).

Next, we describe the M-step. Using the new objective function, \((\lambda, \Theta, \theta_0)\) in the \(t\)-th iteration of the M-step can be estimated by Equations 36-38.

\[
\lambda^{(t)} = \frac{\sum_{X_u \in X_Q} \sum_{Z^{(t)}} X_{iu} P r(Q(X_{iu}) = 1)}{\sum_{X_u \in X_Q} \sum_{Z^{(t)}} X_{iu} P r(Q(X_{iu}) = 1)}
\]

For \(k = 1, \ldots, L\) and \(j = 1, \ldots, W\), we have

\[
\theta_{0,k}^{(t)} = \frac{\sum_{X_u \in X_Q} \sum_{Z^{(t)}} X_{iu} \sum_{w=1}^{W} (1-Z^{t}) I(a_k, X_i,j)}{\sum_{X_u \in X_Q} \sum_{w=1}^{W} P r(Q(X_{iu}) = 1)}
\]

\[
\theta_{j,k}^{(t)} = \frac{\sum_{X_u \in X_Q} \sum_{Z^{(t)}} X_{iu} \sum_{w=1}^{W} Z^{t} I(a_k, X_i,j)}{\sum_{X_u \in X_Q} \sum_{w=1}^{W} P r(Q(X_{iu}) = 1)}
\]

As the position and sequence rank modeling parameters are independent to our sampler function \(Q(\cdot)\), so we do not have to re-weight each site in \(X_Q\). (\(\alpha_1, \ldots, \alpha_K, \beta_1, \ldots, \beta_K\) are updated using Equations 15 and 16, except that we replace \(X\) with \(X_Q\).

For the value of \(\tau_{\text{pos}}\) and \(\tau_{\text{rank}}\), they will be updated depending on the result of two Chi-square tests. Precisely, \(\tau_{\text{pos}} = 1\) if the positional distribution of binding sites \(X_i^{(\text{pos})} \cdot P r(X_i^{(\text{pos})}|Z_{iu} = 1)\) is significantly different from the uniform distribution (Chi-square test); and \(\tau_{\text{pos}} = 0\) otherwise. The Chi-square statistic \(\chi\) is defined as: (Recall \(I(\cdot, \cdot)\) is the indicator function.)

\[
\chi_{\text{pos}} = \sum_{k=1}^{K} \left( \frac{(\sum_{X_u \in X_i} I(k, X_i^{(\text{pos})}) P r(X_i^{(\text{pos})}|Z_{iu} = 1) - \frac{1}{K} \sum_{X_u \in X_i} P r(X_i^{(\text{pos})}|Z_{iu} = 1))}{\sum_{X_u \in X_i} P r(X_i^{(\text{pos})}|Z_{iu} = 1)} \right)^2
\]

Similar significance test can be applied for \(\tau_{\text{rank}}\).

\[
\chi_{\text{rank}} = \sum_{k=1}^{K} \left( \frac{(\sum_{X_u \in X_i} I(k, X_i^{(\text{rank})}) P r(X_i^{(\text{rank})}|Z_{iu} = 1) - \frac{1}{K} \sum_{X_u \in X_i} P r(X_i^{(\text{rank})}|Z_{iu} = 1))}{\sum_{X_u \in X_i} P r(X_i^{(\text{rank})}|Z_{iu} = 1)} \right)^2
\]

As the default setting, if the p-value for the Chi-square test is less than 0.05, the indicator variable \(\tau_{\text{pos}}\) or \(\tau_{\text{rank}}\) will be updated to 1, respectively; 0. Otherwise.

For the starting conditions, the sampling rate \(\mu\) is set to 0.01 in this paper, which is a trade-off between efficiency and accuracy. The initial \((\lambda^{(0)}, \theta_1, \theta_0, \Theta^{(0)})\) can be set as:

\[
\lambda^{(0)} = \frac{2(\lambda^{(\text{EMM})} \cdot |X_i^{(\text{EMM})}|)}{|X|}
\]
Procedure 2 Re-sampling EM

Input: EEM outputted PWM $\Theta^{(EEM)}$, sampling rate $\mu$, expected number of binding sites in EEM ($\lambda^{(EEM)}$, $|X_q^{(EEM)}|$), the original data $X$

Output: Final recovered PWM $\Theta^{(t)}$

1: Initialize ($\lambda^{(0)}$, $\tau_{pos}^{(0)}$, $\tau_{rank}^{(0)}$, $\Theta^{(0)}$), $\{\alpha_1^{(0)}$, $\alpha_2^{(0)}$, ..., $\alpha_K^{(0)}\}$, $\{\beta_1^{(0)}$, $\beta_2^{(0)}$, ..., $\beta_K^{(0)}\}$) according to Equations 41-46
2: $X_Q :=$ sampling a subset of $X$ using $\Theta^{(EEM)}$ and Equation 33;
3: $t := 1$
4: repeat
5: E-step: $\forall X_{iu} \in X_Q$, compute $Z_{iu}^{(t)}$ according to Equations 5 and 6
6: M-step: compute ($\lambda_t^{(t)}$, $\tau_{pos}^{(t)}$, $\tau_{rank}^{(t)}$, $\Theta^{(t)}$, $\theta_0^{(t)}$, $\{\alpha_k^{(t)}\}_{k=1}^K$, $\{\beta_k^{(t)}\}_{k=1}^K$)
7: $t := t + 1$
8: until $\Theta^{(t)}$ converge;
9: return $\Theta^{(t)}$;

where ($\lambda^{(EEM)}$, $|X_q^{(EEM)}|$) is the expected number of binding sites estimated from the EEM procedure.

\[ \tau_{pos}^{(0)} = 0, \tau_{rank}^{(0)} = 0 \] (42)

\[ \Theta^{(0)} = \Theta^{(EEM)} \] (43)

\[ \forall k = 1, \ldots, L, \theta_{0,k}^{(0)} = \sum_{X_{iu} \in X_Q} I(a_k, X_{iu,j}^{(seq)}) / |X_Q| \] (44)

\[ \forall k = 1, \ldots, K, \alpha_k^{(0)} = \alpha_k^{(EEM)} \] (45)

\[ \forall k = 1, \ldots, K, \beta_k^{(0)} = \beta_k^{(EEM)} \] (46)

In fact, the initialization for the position model (Equation 45) and sequence rank model (Equation 46) is not necessary, because in the first EM iteration, they are always excluded in the E-step computation and updated in the M-step. The pseudo code of re-sampling EM procedure is described in Procedure 2.

3.5 Sorting and Redundancy Filtering

The PWMs outputed by REM are evaluated and sorted by empirical AUC value or Z-score with the input data (defined in section 4.1 and 4.2). Finally, we try to eliminate redundant PWMs as follows. If a PWM overlap with another PWM by more than 10%, we will treat the PWM with lower score as redundant and remove it.

Procedure 3 AUC

Input: PWM $\Theta$, positive data $X^{pos}$, negative data $X^{neg}$

Output: AUC value

1: for all sequence $X_i \in X^{pos}$, $X^{neg}$ do
2: $\text{Score}_i :=$ compute maximum PWM score $\log Pr(X_{i,j} | \Theta)$, for all subsequence $X_{i,j} \in X_i$
3: end for
4: Sort $\{\text{Score}_i\}$ in descendent order, and $\{i_1, \ldots, i_n\}$ are the sorted indices
5: for $k = 1, \ldots, n$ do
6: $TPR_k := |\{X_{1} \in X^{pos}, \theta_{i_{k1},i_{k2}} \in \text{Score}_{i_{k1}} \} / |X^{pos}||$ [true positive rate]
7: $FPR_k := k - |\{X_{1} \in X^{pos}, \theta_{i_{k1},i_{k2}} \in \text{Score}_{i_{k1}} \} / |X^{pos}||$ [false positive rate]
8: end for
9: Compute ROC curve based on $\{TPR_k, FPR_k\}$
10: return the area under the ROC curve
3.6 Implementation Details

Avoid insufficient samples  In the default setting, REM procedure will at least subsample 1000 sites from the original input data. This is to avoid learning from a subsample with insufficient data. As an implementation detail, we further consider terrible edge effect when the sample size is not enough. Some sites are not expected to be sampled in the given sampling rate. If these exception sites are considered, the motif model will quickly converge to the single site with heaviest weight. Hence, in each REM iterations, we further ignore the sites with ultra low sampling probability (i.e., \( Pr(Q(X_{iu}) = 1) < \frac{1}{1+2|X|Pr(X_{iu}|Z_{iu}=0)} \)) in the M-step in the implementation.

Masking ChIPed TF motif in coTF motif finding  If option "-mask" is used, SEME treats the motif have both significant position preference and sequence rank preference as ChIPed TF motif. Prior to mining the motifs of co-TFs, the sites in the input sequences occupied by the ChIPed TF motif (under FDR 0.0001) will be masked.

Binning for position and sequence rank  We assume all input sequences have the same length, and each sequence is divided into \( K \) equal position bins. The choice of \( K \) is defined by \( \max(L/w; 20) \), where \( L \) is the length of each sequence and \( w \) is the length of motif. Once the \( K \) is determined, the sequence rank follows the same \( K \), which divides the list of ordered sequences into \( K \) rank bins, and each rank bin contains equal number of sequences.

Handling “N”-mask regions and unequal length sequences  In the current implementation, if the input DNA sequences contain “N”, those characters will be removed from the original sequences (e.g., AAACCNNNTTTT is converted to AAACCTTTT). So it is possible that the original input sequences are equal length, and they become unequal after removing the “N”-mask regions. The unequal length sequences will bring problem in the position preference detection. Once the number of position bins \( K \) is determined using average sequence length \( L \), all the unequal-length sequences will be equally divided into \( K \) bins, although the bin size for different sequences are different. For each site \( X_i \), SEME only considers which position bin it occurs instead of the absolute position. This strategy works well when all sequences have approximately the same length.

4 Measurement used in the evaluation

4.1 AUC (Measure Discriminative Capability)

Given a set of positive sequences and negative sequences, and a PWM motif. We compute the best match score of the PWM motif in every sequence. Then using different PWM score cut-off, we can compute the "True Positive Rate" and "False Positive Rate" of the PWM and generate the receiver operating characteristic (ROC) curve. Finally, the AUC score of the given PWM can be calculated as the area under the ROC curve. The procedure can be referred to Procedure 3.

4.2 Z-score (Measure Over-representation)

Given a set of positive sequences and negative sequences, and a PWM motif. Under the PWM score cutoff with FDR=0.001, we get all the sites matching the PWM motif in both positive sequences and negative sequences. Then, we define \( TP \) be the number of matched sites in the positive sequences and \( FP \) be the number of matched sites in the negative sequences. And \( T \) and \( F \) denote the total number of sites in the positive and negative sequences, respectively. Then, we define the binomial Z-score as follow.

\[
Z_{score} = \frac{TP - FP \cdot \frac{T}{P}}{\sqrt{\frac{T}{(1 - \frac{FP}{P}) \frac{FP}{P}}}}
\]  \hspace{1cm} (47)

4.3 Other Statistical Measurements

Given a set of positive sequences and negative sequences, and a PWM motif. We get all the sites matching the PWM motif higher than the PWM score cut-off (under FDR=0.001) in both positive sequences and negative sequences. Then, we define \( TP \) as the number of matched sites in the positive sequences and \( FP \) as the number of matched sites in the negative sequences. And \( TN \) and \( FN \) denote the number of unmatched sites in the positive sequences and the negative sequences, respectively. Then, positive predictive value is defined as follow.
PPV (Positive Predictive Value)

$$PPV = \frac{TP}{TP + FP} \tag{48}$$

SPC (Specificity)

$$SPC = \frac{TN}{TN + FP} \tag{49}$$

ASP (Average site performance)

$$ASP = \frac{TP}{TP + FP} + \frac{TP}{TP + FN} \tag{50}$$

4.4 PWM Divergence (Measure Similarity)

Harbison et al [4] introduced a normalized Euclidean distance to measure the divergence between two PWMs $M^1$ and $M^2$ as follows:

$$PWM\text{Divergence} = Euclidean\text{Distance}(M^1, M^2) = \frac{1}{\sqrt{2 \cdot l}} \sum_{i=1}^{l} \sqrt{\sum_{b \in \{A,C,G,T\}} (M^1_{i,b} - M^2_{i,b})^2} \tag{51}$$

where $l$ is the overlap length of two aligned PWMs. The divergence score is between 0 (perfect identity) and 1 (complete dis-similarity). The above computation is repeated for each possible alignment of the PWMs which contains at least seven overlapping columns (only the overlapping columns are included in the first sum); if one of the motifs is shorter than 8, the minimal allowed overlap is the shorter motif length minus 2. All computations are repeated with the reverse complement of one of the PWMs. Finally, the smallest distance is chosen as the PWMs divergence score.

5 Programs used and their parameter settings

All tools were tuned to their best performances by our best effort in all experimental settings. However, due to running time issue, the input sequences will be clipped to 1000 or 2000 for some programs like MEME, Weeder, since those programs cannot handle big datasets. All programs tested in this paper have standalone version, and they were tested in our linux server with 48GB RAM, two Intel X5670 CPUs and two Fermi M2050 GPUs.

SEME
http://biogpu.ddns.comp.nus.edu.sg/chipseq/SEME/
command for Simulation:
java -jar SEME.jar -i $inputseq.fa -bgmodel uniform.bg -maxlen 30 -n 5
command for Amadeus Benchmark:
java -jar SEME.jar -i $inputseq.fa -n 4 -maxlen 12
command for ChIP-seq ChIPed TF motif:
java -jar SEME.jar -i $inputseq.fa -c $bgseq.fa -n 5
command for ChIP-seq co-TF motif:
java -jar SEME.jar -i $inputseq.fa -c $bgseq.fa -n 20 -mask

We set maximum motif length to 30bp in the simulation data as we know the maximum length of planted motif is 30bp (we also set MEME maximum motif length to 30bp in simulation) and for Amadeus benchmark, we limit the maximum length to 12bp, as the maximum motif length supported by Amadeus is also 12bp, and other motif finders’ motif lengths are also limited to less 12bp in the original paper[7]. For other experiments, we use the common setting (default maximum motif length 15bp) to ensure the output result comparable to other programs in different measurement
(like PWM divergence, which is known to have bias in motif length). Background sequences for ChIP-seq experiment are extracted from 1000bp-away flanking region from the ChIP-seq peak summit locations.

**MEME**

http://meme.nbcr.net/meme4_3_0/

command for Simulation:

meme $inputseq.fa -dna -nmotifs 5 -maxw 30 -mod zoops -revcomp -maxsize 50000000 -bfile uniform.bg

command for ChIP-seq ChIPed TF motif:

meme $top2kseq.fa -dna -nmotifs 2 -maxw 20 -mod zoops -revcomp -maxsize 50000000 -bfile $bgfile

command for ChIP-seq co-TF motif:

meme $top2kseq.fa -dna -nmotifs 20 -maxw 20 -mod zoops -revcomp -maxsize 50000000 -bfile $bgfile

**CUDA-MEME**

https://sites.google.com/site/yongchaosoftware/Home/cuda-meme

command for Simulation:

cuda-meme $inputseq.fa -dna -nmotifs 5 -maxw 30 -mod zoops -revcomp -maxsize 50000000 -gpu_num 0 -num_threads 6 -bfile uniform.bg

command for ChIP-seq ChIPed TF motif:

cuda-meme $top2kseq.fa -dna -nmotifs 2 -maxw 20 -mod zoops -revcomp -maxsize 50000000 -gpu_num 0 -num_threads 6 -bfile $bgfile

command for ChIP-seq co-TF motif:

cuda-meme $top2kseq.fa -dna -nmotifs 20 -maxw 20 -mod zoops -revcomp -maxsize 50000000 -gpu_num 0 -num_threads 6 -bfile $bgfile

**HMS**

http://www.sph.umich.edu/csg/qin/HMS/

command for ChIP-seq ChIPed TF motif:

hms -i $inputseq.fa -w 20 -dna 4 -iteration 100 -chain 10 -strand 2 -seqprop 0.1 -t_dof 3 -dep 1 -peaklocation $peaksummitfile

command for ChIP-seq co-TF motif (try $width=6-26, output 20 different PWMs):

hms -i $inputseq.fa -w $width -dna 4 -iteration 100 -chain 10 -strand 2 -seqprop 0.1 -t_dof 3 -dep 1 -peaklocation $peaksummitfile

**ChIPMunk**

http://line.imb.ac.ru/ChIPMunk/

command for ChIP-seq ChIPed TF motif:

java -cp ru_genetika.ChIPHorde 25:25,20:20,15:15,10:10,8:8 mask yes 1.0 s: $inputseq.fa 100 10 1 4 random $CGbias

command for ChIP-seq co-TF motif (try 4 sets of motif width=8,10,15,20,25, output 20 different PWMs):

java -cp ru_genetika.ChIPHorde 25:25,20:20,15:15,10:10,8:8,25:25,20:20,15:15,10:10,8:8,25:25,20:20,15:15,10:10,8:8,25:25,20:20,15:15,10:10,8:8 mask yes 1.0 s: $inputseq.fa 100 10 1 4 random $CGbias
comment:
CGratio in the input sequences is provided to ChIPMunk

Weeder
http://159.149.109.9/modtools/
command for ChIP-seq ChIPed TF motif:
weederlauncher.out Stop1kseq.fa HS medium S
command for ChIP-seq co-TF motif:
weederlauncher.out Stop2kseq.fa HS large S T20
comment:
Weeder is a very time consuming program, so we only run medium mode for top 1000 sequences with highest ChIP
Intensity in ChIPed TF motif finding and large mode for top 2000 sequences with highest ChIP Intensity in co-TF
motif finding.

Cisfinder
http://lgsun.grc.nia.nih.gov/CisFinder/
command for ChIP-seq ChIPed TF motif:
get_motifs.pl $outputDir $inputseq.fa 0.05 1.5 0.75 1000 -POS 1 -ONE -C $bgseq.fa
command for ChIP-seq co-TF motif :
get_motifs.pl $outputDir $inputseq.fa 0.05 1.5 20 0.75 1000 -POS 1 -ONE -C $bgseq.fa
comment:
perl script for running cisfinder program can be downloaded in:
http://lgsun.grc.nia.nih.gov/CisFinder/download/get_motifs.pl

Trawler
http://ani.embl.de/trawler/
command for ChIP-seq ChIPed TF motif:
trawler.pl -nb_of_cluster 5 -directory $outputDir -sample $inputseq.fa -background $bgseq.fa -dir_id trawler -xtralen 2
command for ChIP-seq co-TF motif :
trawler.pl -directory $outputDir -sample $inputseq.fa -background $bgseq.fa -dir_id trawler -xtralen 2
comment:
for co-TF motif, we don’t use option nb_of_cluster (using K-mean clustering), as we think the default clustering method
can report more motifs. And we tested -nb_of_cluster 20 option, in many case, these option wont output any motif by
reporting no significant clusters

Amadeus
http://acgt.cs.tau.ac.il/amadeus/
command for ChIP-seq ChIPed TF motif:
java -jar Allegro_v1.0.jar file parafile_len12_normal
command for ChIP-seq co-TF motif:
java -jar Allegro_v1.0.jar file parafile_len12_large
comment:
In order to run Amadeus in command line, we need prepare a parameters-configure file, which contains the target set
(input sequences) and background set (input sequences + background sequences). For normal mode, the Amadeus will
at most output 5 non-redundant motifs and for large mode, it will output at most 20 non-redundant motifs. We set the
motif length as 12bp, which is the maximum support motif length for Amadeus.

References


