Hindawi Publishing Corporation
BioMed Research International
Volume 2015, Article ID 218068, 10 pages
http://dx.doi.org/10.1155/2015/218068

Research Article

A Fast Cluster Motif Finding Algorithm for ChIP-Seq Data Sets

Yipu Zhang and Ping Wang

Department of Automation, School of Electronics and Control Engineering, Chang'An University, Xi'an 710064, China

Correspondence should be addressed to Yipu Zhang; zephyr26026@163.com

Received 8 April 2015; Accepted 4 June 2015

Academic Editor: Andre Van Wijnen

New high-throughput technique ChIP-seq, coupling chromatin immunoprecipitation experiment with high-throughput sequencing technologies, has extended the identification of binding locations of a transcription factor to the genome-wide regions. However, the most existing motif discovery algorithms are time-consuming and limited to identify binding motifs in ChIP-seq data which normally has the significant characteristics of large scale data. In order to improve the efficiency, we propose a fast cluster motif finding algorithm, named as FCmotif, to identify the (l, d) motifs in large scale ChIP-seq data set. It is inspired by the emerging substrings mining strategy to find the enriched substrings and then searching the neighborhood instances to construct PWM and cluster motifs in different length. FCmotif is not following the OOPS model constraint and can find long motifs. The effectiveness of proposed algorithm has been proved by experiments on the ChIP-seq data sets from mouse ES cells. The whole detection of the real binding motifs and processing of the full size data of several megabytes finished in a few minutes. The experimental results show that FCmotif has advantageous to deal with the (l, d) motif finding in the ChIP-seq data; meanwhile it also demonstrates better performance than other current widely-used algorithms such as MEME, Weeder, ChIPMunk, and DREME.

1. Introduction

A Transcription Factor (TF) binds to the specific DNA sequences, which carries the information of the transcription and gene expressions. Hence locating the Transcription Factor Binding Sites (TFBSs) is crucial for uncovering the underlying relationship of regulating transcription and comprehending evolutionary mechanism of living organisms. The identification of TFBSs, or socalled motif discovery, is an issue of discovering short similar nucleotide segments with a common biological function. The planted (l, d) motif discovery is the famous version for motif discovery [1], which can be formulated as follows: given a set of n-length DNA sequences $\mathbf{S} = \{s_i \mid i = 1, \ldots, t\}$ over the alphabet $\Sigma = \{\text{A, C, G, T}\}$, two nonnegative integers l and d $(d < l < n)$, where l is the length of a motif and d is the maximum number of mutations between the motif and a predicted binding site. The task is to find a l-length motif m occurring in most of the sequences including up to d mutations. m is called an (l, d) motif and each occurrence of m is called a motif instance. Various motif discovery algorithms have been developed to locate motifs in promoter sequences from coregulated or homologous genes based on either Consensus or Position Weight Matrix (PWM) [2].

In recent years, high-throughput technique ChIP-seq [3, 4], which couples chromatin immunoprecipitation experiment [5] with high-throughput sequencing technologies, has extended the identification of binding locations of a given TF to that of the genome-wide regions. The genome-wide ChIP experiment generally produces thousands of sequences of a few hundred bps (ChIP-seq peaks), which provides data set of one or two magnitudes larger than a typical motif discovery data set and sequences with a high resolution. The novel ChIP technique ChIP-exo can locate binding sites at a higher resolution, but its binding regions identified by ChIP-seq or ChIP-exo experiments may be dozens of bps away from the true binding sites [6]. Computational motif discovery methods are still needed to identify the binding locations of a TF in ChIP-seq or ChIP-exo data sets [7] in the high accuracy.

In order to detect motifs in large-scale ChIP-seq data, some traditional motifs discovery algorithms have been proposed in their ChIP-tailored versions, such as MDscan [8]

and MEME-ChIP [9]. These algorithms normally find motifs by using a limited part of the sequences, while ignoring the remaining unselected sequences. That decreases the chance of discovering motifs related to infrequent cofactors. Meanwhile, PWM-based methods also have been developed. For instance, STEME [10] applies suffix trees to accelerate EM steps. This strategy acts well in case of finding short motifs. However, it executes much slower when the width of motif increases in the large data set. HMS [11] is an improved version of Gibbs that combines sampling algorithms with greedy search steps. ChIPMunk [12] introduces EM algorithms with a greedy approach and applies a more complex statistic model. These algorithms aim to optimize a PWM of ChIP-enriched region. They still have an unsolved problems of local optimum and the iteratively training also costs too much. Additionally, consensus-based algorithms are designed based on word-enumeration methods, such as RAST [13] and CisFinder [14], which can process whole ChIP-seq data set by two contrastive data sets. Both RAST and CisFinder are limited to find short motifs and may miss the useful information contained in the sequences.

To overcome these shortcomings, in this paper, we propose a fast cluster motif finding algorithm, named FCmotif, to solve the (l, d) motif identification problem in large scale ChIP data set. FCmotif utilizes the emerging substrings mining strategy to find the enriched substrings at first and makes each emerging substring as a reference core to construct PWM. Then our algorithm uses the constructed PWMs to cluster the motifs in different length, and we consider intramotif dependency in statistics model to calculate information content (IC) and false discovery rate (FDR) to optimize the outputs. FCmotif achieves to deal with the whole data set that does not limit to the OOPS (one occurrence of the motif instance per sequence) constraint. The experimental results show that FCmotif is advantageous to deal with the (l, d) motif finding in the ChIP-seq data, and it also demonstrates better performance than other current widely-used algorithms such as MEME, Weeder, ChIPMunk, and DREME.

2. Materials and Methods

We know that the characteristic of a ChIP-seq data set is a large scale set of relative shorter sequences. That is, the amount and quality of ChIP-seq data have been dramatically increased. Each sequence of ChIP-seq data set contains less "the background information," and several instances of the motifs could be expected to exist in thousands of sequences. From this point of view, our main objective is to handle the whole data set and distinguish the motif instance from the relative "cleaner" background sequence.

2.1. Motif Representation. Generally, a motif can be represented by a PWM Θ, of which each column stores the occurring frequency of the four types of nucleotides ($\Sigma = \{A, C, G, T\}$). Let $\Theta = (\theta_1, \ldots, \theta_l)$, where θ_i represents the probability of nucleotide preference at the ith position of the motif, and let θ_0 be the probability of nucleotide observing at the nonmotif positions in the sequences. For each substring of l length (we also call l-mer) $s = s_1, \ldots, s_l$, the log-likelihood of letter s_i at position i is given by

$$p(s_i) = \log \frac{\theta_{ik}}{\theta_{0k}}, \tag{1}$$

where θ_{ik} is the probability of observing letter $s_i = k$ at position i and θ_{0k} is the background probability of letter k. This classical product-multinomial model proposed by Liu et al. [15] has been widely used in *de novo* statistic algorithms such as EM and Gibbs algorithm. It assumed that the positions within the motif are independent of each other [16]. However, recent researches imply that the commonly used product-multinomial model may be too simplistic in identifying the binding motifs, while some positions of TF binding motif exert an interdependent effect on binding affinities of TFs [17–19].

To provide a better fit model to increase the quality of motifs identified by ChIP-Seq, a more sophisticated model that involves the intramotif dependency should be considered. Here, "intramotif dependency" means that the frequency of nucleotide combinations spanning several positions deviates from the expected frequency under the independent motif distribution [11]. For instance, if the frequency of two nucleotides, "GT," in a pair of positions is much higher or lower than the product of frequency of "G" in the first position and frequency of "T" in the second position, we infer that these two positions are dependent. Here, we implement a 16-component dependent multinomial model to scan each pair of positions within the motif to determine the intramotif dependency. Let $\Phi_{i,i+1}$ represent the probability of observing nucleotide pair at ith and $(i + 1)$th position of the motif. For each pair of positions, there are l-1 dependent multinomial distributions to be estimated. The log-likelihood of letters s_i, s_{i+1} at position i and $i + 1$ is

$$p(s_i, s_{i+1}) = \log \frac{\Phi_{i,i+1}(s_i, s_{i+1})}{\Phi_0(s_i, s_{i+1})}, \tag{2}$$

where Φ_0 represents the background probability of the nucleotide pair. The Log-Likelihood Ratio (LLR) of l-mer s is then

$$\mathrm{LLR}(s) = \log \frac{U(s) \cdot V(s)}{p_0(s)}, \tag{3}$$

$$U(s) = \prod_{i=1}^{l} \prod_{k \in \Sigma} \theta_{ik}, \tag{4}$$

$$V(s) = \prod_{i=1}^{l-1} \prod_{k_1, k_2 \in \Sigma} \Phi_{i,i+1}(k_1, k_2). \tag{5}$$

Here, formula (4) represents the joint probability of the independent nucleotides in motif, and formula (5) represents the joint probability of the nucleotide pair in motif. Formula (3) is the LLR of s under the corresponding background distribution p_0. For the background (nonmotif) regions, we employ a high-order Markov model to obtain the weak

dependency in background DNA sequences. Compared with the uniform distribution or random distribution background, the high-order Markov model can improve the sensitivity and specificity of identifying motifs. In this study, we use a third-order Markov model to characterize the background sequence. As an example, the probability of an l-mer s $(s_i, s_{i+1}, \ldots, s_{i+l-1})$ in the background under a third-order Markov model can be represented by

$$p_0(s) = p(s_i)\, p(s_{i+1} \mid s_i)\, p(s_{i+2} \mid s_i, s_{i+1}) \cdots p(s_{i+l-1} \mid s_{i+l-2}, s_{i+l-3}, s_{i+l-4}). \quad (6)$$

Thereby, the Information Content of motif can be represented as

$$\mathrm{IC} = \sum_{s \in \Sigma^l} p(s \mid \Theta) \log\left(\frac{p(s \mid \Theta)}{p_0(s \mid \theta_0)}\right). \quad (7)$$

2.2. Emerging Substrings Mining. For the large-scale data set, calculating the likelihood score of each substring costs too much, which makes probabilistic training methods unpractical. Pattern-driven strategy can use shorter time to count the substrings that have higher occurrence frequencies. Since each instance differs from motif at most d positions, we expect to find some instances occurring multiple times in thousands of sequences and reduce the disturbance of random overrepresented substrings. With the above considerations, we utilize both a test set and a control set of DNA sequences to search the possible motif instances. Generally, the test set consists of the sequences with motifs, while the control set contains the background sequences. The interested substrings are the ones that present in the test set and absent in the control set, and we call such substrings emerging substrings. The task converts to solve emerging substrings mining problem [20] and then identifies motif instances from the emerging substrings. The emerging substrings mining problem is defined as follows.

Given a test set $\mathbf{S}_t$ and a control set $\mathbf{S}_c$ of sequences over the alphabet $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, frequency threshold λ_f $(1/|\mathbf{S}_t| \le \lambda_f \le 1)$, and growth rate threshold λ_g $(\lambda_g > 1)$, the task is to find all substrings u $(l_{\min} \le u \le l_{\max})$ satisfying the conditions $f(u, \mathbf{S}_t) \ge \lambda_f$ and $g(u, \mathbf{S}_t, \mathbf{S}_c) \ge \lambda_g$ at the meantime. Such substrings are called emerging substrings. Here, $f(u, \mathbf{S})$ represents the frequency of substring u occurring in set $\mathbf{S}$, and $g(u, \mathbf{S}_t, \mathbf{S}_c) = f(u, \mathbf{S}_t)/f(u, \mathbf{S}_c)$, that is, the growth rate of substring u from set $\mathbf{S}_t$ to set $\mathbf{S}_c$. Large value $g(u, \mathbf{S}_t, \mathbf{S}_c)$ means that substring u is highly discriminative for two input data sets.

With the above material, our algorithm can be summarized as the following main procedures. First, we compare the substrings in both test set and control to obtain the emerging substrings. Second, calculate measure score of the emerging substrings to find the true motif instances. Nevertheless, there are still some key problems needed to be solved: (i) As the exact motif length is unknown, we need to select a range of emerging substring length to find motif. (ii) The interested emerging substrings contain true motifs, the instances of both mutation and random disturbance, how to reduce the influence of the unreal instances. (iii) We need to choose one model from OOPS, ZOOPS (zero- or one-motif occurrences per sequence), and TCM (two-component mixture) to find motif instances in each sequences. Therefore, our algorithm is designed in detail to further process the emerging substrings and handle these problems.

2.3. FCmotif Algorithm

Step 1 (searching emerging substrings). An essential assumption is that the evidence for binding motif is large in test set and small in the control set. To streamline the predicting sites algorithm and handle the ChIP-Seq data, our algorithm utilizes pattern-driven word enumeration strategy to search the emerging substrings. Assume motif length is l; we first count the amount of all possible 4^l l-mers in both test set and control set; then we select the rich ones. The threshold frequency λ_f and growth rate λ_g are two important parameters employed in this step.

As previous studies [21, 22], we knew the probability of the occurrence of a random mutated instance m' of a reference motif m at random i positions is

$$P_d(i) = \binom{d}{i} p_{\mathrm{con}}^i (1-p)^{d-i}, \quad 0 \le i \le d, \quad (8)$$

where p_{con} is the mutating probability, and it can also represent the conservation of motif. We set p_{con} as 0.2, 0.5, and 0.8 to represent high conservation, intermediate conservation, and low conservation, respectively.

Then, according to the definition of (l, d) motif, the probability of a random (l, d) instance m' of motif m occurring in a sequence can be calculated by

$$P_{\mathrm{occ}} = \sum_{i=0}^{d} P_d(i) \frac{1}{\binom{l}{i} \times 3^i}. \quad (9)$$

Moreover, for the different models, each sequence contains different amount of motif instances, so the value of λ_f can be set by different models and P_{occ}. We set $\lambda_f = 0.8P_{\mathrm{occ}}$ when model is OOPS, $\lambda_f = 0.6P_{\mathrm{occ}}$ when model is ZOOPS, and $\lambda_f = 1.2P_{\mathrm{occ}}$ for TCM. Meanwhile, the default value of λ_g that we set is 2. Table 1 shows an example of searching the emerging substrings of length 6 in 600 sequences for ZOOPS model; $(1, d) = (6, 1)$ and $p_{\mathrm{con}} = 0.8$. From the example, we can find that the emerging substring "CAGCGA" satisfies both $f(u, \mathbf{S}_t) > \lambda_f$ and $g(u, \mathbf{S}_t, \mathbf{S}_c) > \lambda_g$. However, only the emerging substring cannot indicate motif; it may miss the mutated instances especially for larger value of l and d.

Step 2 (constructing the corresponding PWM). The emerging substrings can represent a part of the enriched (overrepresented) motifs. However, it still contains the fake instances made up by the background sequences and cannot reflect the true mutated (l, d) motif instances. It is necessary to measure the statistics scores of all possible instances and find the (l, d) instances among the emerging substring. The PWM indicates the distributions of each character at each position

Table 1: An example of searching the emerging substrings of all possible l-mers.

l-mer	Number in test set	Number in control set	$f(u, \mathbf{S}_t)$	$g(u, \mathbf{S}_t, \mathbf{S}_c)$	$\lambda_f = 0.02667$ $\lambda_g = 2$
AACTGC	5	16	0.0083	0.3125	N
AAGTGG	8	6	0.0133	1.3333	N
CAGCGA	19	3	0.0317	6.3333	Y
TGACTT	15	7	0.025	2.1429	N
GCTTCA	2	5	0.0033	0.4	N
⋮	⋮	⋮	⋮	⋮	⋮

Table 2: An example of searching the neighbourhood instances of a reference l-mer.

Reference l-mer: ACCACGTG $(l, d) = (8, 1)$ $z = 19.36$ $C_1 = 119$ $C_2 = 9$									
Position	Instance	z	C_1	C_2	Position	Instance	z	C_1	C_2
$i = 1$	***C***CCACGTG	**21.78**	146	11	$i = 2$	A***A***CACGTG	**11.83**	51	5
	GCCACGTG	**26.14**	177	7		A***G***CACGTG	**22.66**	149	9
	TCCACGTG	**15.93**	91	9		A***T***CACGTG	**3.28**	16	6
$i = 3$	AC***A***ACGTG	**2.38**	9	4	$i = 4$	ACC***C***CGTG	**2.88**	15	6
	AC***G***ACGTG	**1.82**	7	3		ACC***G***CGTG	**8.23**	25	3
	AC***T***ACGTG	1.31	5	3		ACC***T***CGTG	**6.16**	23	5
$i = 5$	ACCA***A***GTG	−0.73	7	9	$i = 6$	ACCAC***A***TG	0.14	27	26
	ACCA***G***GTG	−0.12	14	14		ACCAC***C***TG	−0.30	17	18
	ACCA***T***GTG	**2.88**	48	30		ACCAC***T***TG	−2.08	2	8
$i = 7$	ACCACG***A***G	**5.92**	24	5	$i = 8$	ACCACGT***A***	−0.07	3	3
	ACCACG***C***G	**8.63**	30	4		ACCACGT***C***	1.03	7	5
	ACCACG***G***G	**3.53**	14	5		ACCACGT***T***	−0.05	4	4

of the motif, and it is the core of measuring the statistical significance, so we construct the corresponding PWM of each emerging substring by the (l, d) mutating.

Assume each emerging substring is a reference motif; the motif instances should exist in the mutated l-mers at most d positions from the reference motif (called the "neighborhood" instances), which have a larger amount in the test set than that in the control set. When we find out the mutated instances from the reference motif, we can use them to construct the core PWM and measure the statistical scores. In this way, see each emerging substring as a reference; we first search its "neighborhood" instances and evaluate the z-score of each one. z-score is a statistical measurement of a score's relationship to the mean in a group of scores which is estimated based on the hypergeometric probability distribution [14]:

$$z(u) = \frac{q_1 - q_2}{\sqrt{q(1-q)(N_1 + N_2)/(N_1 N_2)}}, \tag{10}$$

where $q_1 = C_1/N_1$, $q_2 = C_2/N_2$, and $q = (C_1 + C_2)/(N_1 + N_2)$. C_1 and C_2 represent the number of occurrences of l-mer u in $\mathbf{S}_t$ and $\mathbf{S}_c$, while N_1 and N_2 are the total number of l-mers in $\mathbf{S}_t$ and $\mathbf{S}_c$, respectively.

For convenience of description, here we give an example to explain the searching process. Consider a specific reference l-mer "ACCACGTG," which has 119 matches in the test set and 46 matches (9 matches after adjusting test set and control set with the same size) in the control set. As the previous study [21], for the length $l = 8$, we use $(l, d) = (8, 1)$ to search the "neighborhood" instances. Therefore, we find the l-mers that have mutated to the other three characters at one position from the reference l-mer. Note that using (8, 1) model, we only need to search 24 l-mers to find the "neighborhood" ones but not the whole searching space of 4^l (65536, $l = 8$) l-mers. Table 2 shows each neighborhood instance of the reference l-mer "ACCACGTG," z-score, and the number of occurrences in the test and the control sets.

Use each emerging substring as a reference center and incorporate its neighborhood instances; two Position Count Matrices (PCMs) of size $4 \times l$, M_1 and M_2 can be formed. M_1 is composed of the qualified neighborhood instances in $\mathbf{S}_t$, and M_2 is similarly composed of the qualified neighborhood instances in $\mathbf{S}_c$ adjusted for length (rescaled by N_1/N_2). Here, the qualified neighborhood instances refer to the instances with $z > 1.643$ or the instances with the maximum positive z-score if there is no instances with $z > 1.643$ [23]. While the PCMs M_1 and M_2 are constructed by adding up the counts of "A, C, G, T" at each position of the qualified neighborhood instances.

It is worthy to note that the test set is a mix of motif instances and random disturbances of background, and the control set is full of background noise. Hence, we can use the background distribution found in the control set to recorrect the contribution and estimate the PWM in the test set. That is,

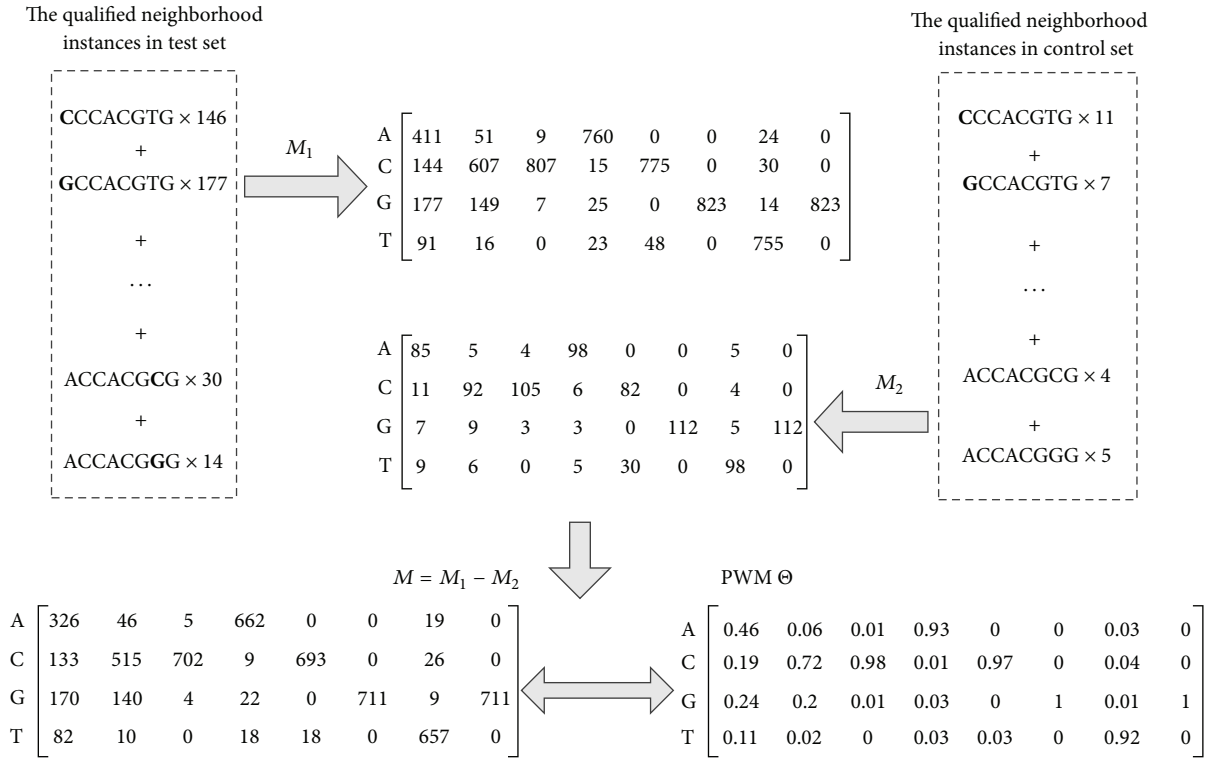

Figure 1: An example of constructing the PWM of the corresponding substrings.

M_2 can be regarded as the expected count matrix constructed from false positive motifs in $\mathbf{S}_t$. In this way, let the PCM $M = \max(M_1 - M_2, 0)$; we can get PWM Θ of the reference emerging substring by normalizing each row into probability distribution. In order to avoid zero frequency, 5% pseudocounts to each position are added. As we also concerned the intramotif distribution in the probabilistic model, Φ can be estimated in the same way. Figure 1 shows an example of constructing PWM of the corresponding emerging substring in Table 2.

Step 3 (clustering longer motifs). See each emerging substring as a seed; its PWM can be obtained by the steps above, while the corresponding motif with high IC score can also be computed. However, the PWMs may represent many similar motifs with a few letters varying as previous studies [24, 25]. In order to eliminate redundant motif information and expand the short motif to form longer motif, we cluster the similar motifs and combine the motifs having the long common-overlap segments by utilizing a metric of computing the Euclidean distance between two l-mers as described below:

$$D(a,b) = \frac{1}{\sqrt{2}l}\sum_{i=1}^{l}\sqrt{\sum_{k\in\Sigma}\left(a_{ik} - b_{ik}\right)^2}, \tag{11}$$

where l is motif length and a_{ik} and b_{ik} are the estimate probabilities of observing letter k at position i of l-mers a and b, respectively. Since the length of predicted motifs may be different, we actually use the minimum distance between motifs among all possible overlaps of motifs a and b induced by shifts that the minimum overlap is 7 bases or two bases fewer when the motifs are even shorter. Hence, we use the Harbison similarity score [26]:

$$\text{sim}(a,b) = \max_{a',b'}\left[1 - D\left(a',b'\right)\right], \tag{12}$$

where a' and b' correspond to all possible overlaps of l-mers a and b. In this way, two l-mers a and b are considered similar if the PWMs of a and b have the Harbison similarity score $\geq$ 0.75. In practice, as the motif length is unknown, we use l ($l_{\min} \leq l \leq l_{\max}$) in a proper range and cluster the PWMs of different length which satisfy the Harbison similarity score constraint. So in this step, a longer motif can be obtained by the corresponding PWM that is combined by clustering the PWMs of different l.

Step 4 (output). With the combined PWMs, we employ two measures to optimize the motifs; first we compute IC and then utilize the False Discovery Rate (FDR) to control the final outputs. The False Discovery Rate as a function of the threshold μ can be intuitively defined as

$$\text{FDR}(\mu) = \frac{I_2 \cdot N_1/N_2}{I_1}, \tag{13}$$

where $I_1 = \sum_{s\in\mathbf{S}_t} I(\text{LLR}(s) > \mu)$ is the number of l-mers found in $\mathbf{S}_t$ and $I_2 = \sum_{s\in\mathbf{S}_c} I(\text{LLR}(s) > \mu)$ is the number of l-mers found in $\mathbf{S}_c$. LLR(s) can be calculated by formula (3). Here, we define μ as an integer satisfying $0 \leq \mu \leq \max[\text{LLR}_{s\in\mathbf{S}_t}(s)]$

Input: a test set $\mathbf{S}_t$ and a control set $\mathbf{S}_c$.
Output: the set of motifs C
(1) $C \leftarrow \emptyset$ // the set of motifs
(2) $X_s \leftarrow \emptyset$ // the set of emerging substrings
(3) $X_q \leftarrow \emptyset$ // the set of the qualified neighborhood instances
(4) $A \leftarrow \emptyset$ // the set of PWMs
(5) $B \leftarrow \emptyset$ // the set of intra-motif distributions
(6) **For** $l \leftarrow l_{\min}$ to $l_{\max}$ **do**
(7) **For** each l-mer of 4^l substrings: u **do**
(8) **if** $f(u, \mathbf{S}_t) \geq \lambda_f$ && $g(u, \mathbf{S}_t, \mathbf{S}_c) \geq \lambda_g$ **then**
(9) add u to X_s
(10) **For** each l-mer of X_s: x **do**
(11) **For** each $d \leftarrow 1$ to $d_{\max}$ **do**
(12) calculate z-score of each neighborhood instance x'
(13) **if** $z(x') > 1.643$ **then**
(14) Add x' to X_q
(15) use x and X_q to construct Θ and Φ
(16) add Θ to set A and add Φ to set B
(17) **For** each Θ of A **do**
(18) **if** $\text{sim}(\Theta, \Theta') \geq 0.75$ $(\Theta' \in A)$ **then**
(19) $\Theta \leftarrow \Theta$ cluster with Θ' and delete Θ' from A.
(20) **if** $\text{FDR}(\mu) > 0.2$ then
(21) delete Θ from A
(22) use Θ and corresponding Φ to compute IC.
(23) add x_{motif} formed by Θ of top 50 IC score to C
(24) **return** C

Algorithm 1

which leads to $\text{FDR}(\mu) < 0.2$ (FDR value changes with different data sets). Once μ is determined, the l-mers in $\mathbf{S}_t$ with $\text{LLR}(s) > \mu$ are the predicted motif instances. In practice, we finally generate at least 50 top IC score motifs by formula (7) satisfying the FDR constraint.

Main algorithm of FCmotif is shown in Algorithm 1.

In Step 1, lines (6) to (9), we find the emerging substrings enriched in the test set; then lines (10) to (15) are the step to construct the PWM and the intramotif distribution for each emerging substring. Lines (16) to (19) are the step to cluster PWM with the similar Harbison similarity score. Lines (20) to (24) are the last step to compute IC and FDR and finally output the result.

3. Results and Discussion

We use the ChIP-seq data sets of 12 TFs profiled in mouse ES cells [27] to test the validity of our algorithm. These 12 data sets are key to the maintenance of pluripotency, in which Nanog, Oct4, Sox2, Esrrb, and Zfx are regulators of self-renewal; Klf4, cMyc, and mMyc are the crucial reprogramming factors [28, 29]; Tcfcp2l1 is preferentially upregulated in ES cells [30]; Smad1 and STAT3 have the significant meaning to the signalling pathways, and CTCF is a key component for transcriptional insulation [31]. For the test set, we extract 200 bps sequence segments centered at a peak of TF location. For the control set, we extract 500 bps sequence segments starting from nucleotide positions 400 bps away from both ends of 200 bps positive sequence segments. The total sizes range from 1 Mb to 50 Mb. Table 3 is the statistics information about the 12 mES ChIP-seq data sets.

Table 3: The information of 12 mES ChIP-seq data sets.

TF	Peaks	Total size (Mb)	Running time (sec)
CTCF	39601	48.98	20570 s
cMYC	3422	4.23	360 s
Esrrb	21644	26.83	6101 s
Klf4	10872	13.45	1871 s
Nanog	10342	12.82	1489 s
nMyc	7181	8.88	1018 s
Oct4	3761	4.65	426 s
STAT3	2546	3.14	295 s
Smad1	1126	1.40	98 s
Sox2	4526	5.60	536 s
Tcfcp2l1	26907	33.59	11075 s
Zfx	10336	12.76	1310 s

Our algorithm runs by using the following parameters: word enumeration analysis is performed with the length l ranging from 6 to 12. $\lambda_f = 0.8P_{\text{occ}}$ when model is OOPS, $\lambda_f = 0.6P_{\text{occ}}$ when model is ZOOPS, and $\lambda_f = 1.2P_{\text{occ}}$ for TCM. $\lambda_g = 2$ while $P_{\text{occ}} = 0.2$, 0.5, and 0.8 to represent high conservation, intermediate conservation, and low conservation, respectively. The (l, d) settings we used include $(6, 1)$, $(7, 1)$, $(8, 1)$, $(9, 2)$, $(10, 2)$, $(11, 2)$, and $(12, 3)$. The default value of threshold z for selecting qualified neighbourhood

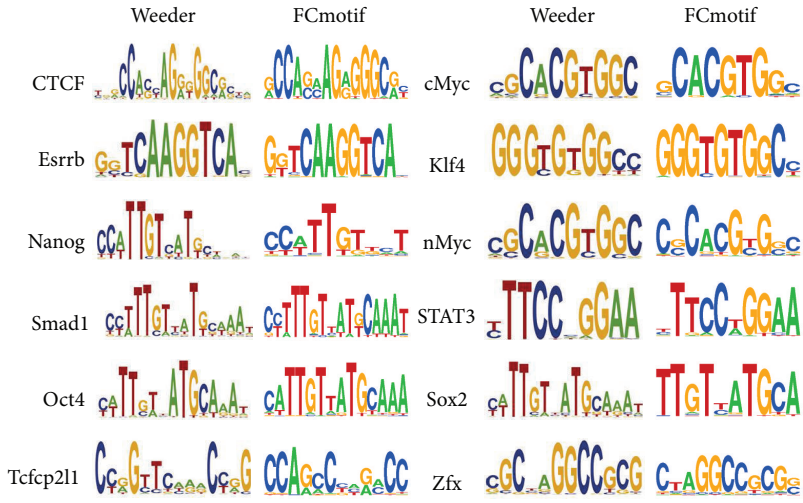

Figure 2: The logos of primary motifs predicted by Weeder and FCmotif.

instances is 1.643, and the FDR constraint is 0.2. FCmotif is implemented in Matlab under the experiment environment: 2.67 Hz CPU and 4 G memory.

3.1. Results on 12 TF Binding Sites in mES Cells. To evaluate the performance of our algorithm, we compare the primary motifs of 12 TFs in mES ChIP-seq data sets discovered by our algorithm with the motifs found by Chen et al. with Weeder. The motif comparison is performed by comparing matrices [32], which supports various scoring metrics and shows the results as the logo of aligned words, in order to grasp the similarities between a predicted motif and the known motifs. Figure 2 shows all 12 motifs identified by FCmotif and motifs found by Chen et al., indicating that the quality of results is comparable. Moreover, we also compare the running time on our algorithm with that of the popular motif discovery algorithms, MEME [33], Weeder [34], ChIPMunk [12], and DREME [35]. Figure 3 shows the running time of above algorithms of 12 mES ChIP-seq data sets. Note that both MEME and Weeder are too slow to deal with ChIP-seq data sets containing thousands of sequences and often fail after running for many days. For this reason, the results in Figure 3 for MEME and Weeder are obtained on the reduced-size data sets. ChIPMunk is an iterative algorithm which can process up to tens of thousands of sequences but with enormous computation at the same time. DREME can predict more accurate motifs than the traditional motif discovery algorithm but was restricted to 500 top-scoring peaks; it cannot analyze the full size data sets. For our algorithm, we found that the computing time scales up efficiently with sequences size and our algorithm outperforms all the compared motif discovery algorithms. Data of several megabytes can be processed in a few minutes. In addition, it is worth to point out that the word length l is another factor to influence the computational efficiency. Because the number of possible l-mers and the number of neighborhood instances are increasing dramatically with l increasing. For example, for Sox2 data set, running time when

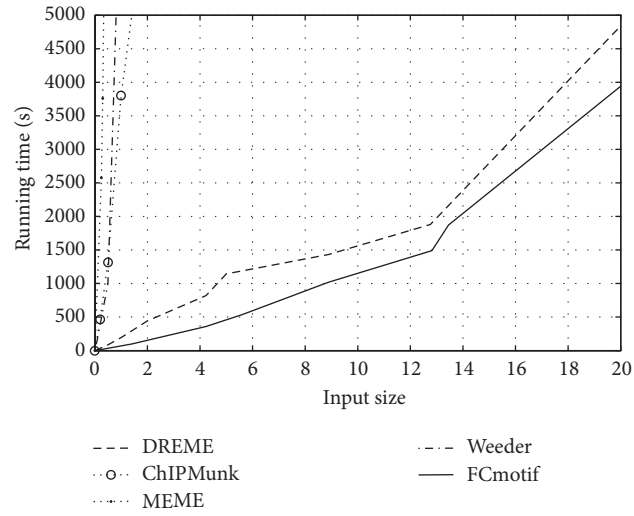

Figure 3: The running time for FCmotif, DREME, ChIPMunk, MEME, and Weeder on the full-size mESCChIP-seq data sets.

$l = 8$ is 536 s but 620 s when $l = 10$. CisFinder [14] is another algorithm that uses the idea of word enumeration and compares the word enrichment of two input sets; it works extremely fast. However, CisFinder outputs the motifs using p-value as a measure, which cannot reflect the significance of the motif but only a single word matching the motif [35].

Although the analysis from 50 to 200 top-scoring binding sites is sufficient to extract the primary motif, yet this data size usually used by Weeder or MEME is not enough to examine alternative motifs. For instance, in Sox2 and Oct4 data sets, Chen et al. report only a single motif with Weeder, respectively. In contrast, FCmotif can find multiple motifs for each TF using the same data sets. As shown in Figure 4(a), our algorithm predict not only the Oct-Sox composite motif bound by Sox2 and Oct4 complex [27] but also the characteristic motifs Sox2 (CCATTGTT) and Oct4

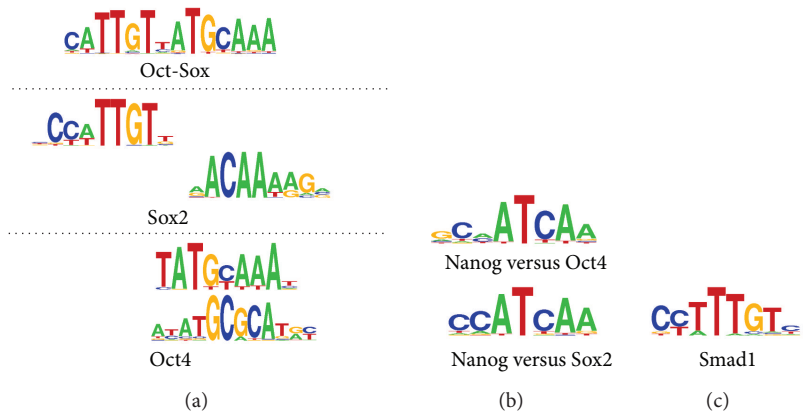

Figure 4: Multiple motifs discovered by FCmotif. (a) The Oct-Sox composite motif, alternative motifs of Sox2, and alternative motifs of Oct4 found by FCmotif. (b) The discriminative motifs found by FCmotif using Nanog data set as the test set, the Oct4 data set, or the Sox2 data set as the control set. (c) The extra motif found by FCmotif in Smad1 data set.

(TATGCAAAT). Meanwhile, some predicted motifs, with no similarity with the prevalent consensus of the data set, often have a high significance and may reveal alternative consensus. Such as the motif (ATATGCGCATGC) in the Oct4 data set, it corresponds to an alternative Oct4 motif reported in other studies [13, 35].

Nanog and Smad1 data sets also have nearly the same binding regions like Sox2 and Oct4 data sets as discussed by Chen et al. As shown in Figure 2, the significant motif found in Nanog is Oct-Sox, which means these similarity binding regions may cause Nanog motif to bind indirectly via one or both of Sox2 and Oct4 TFs and raise the difficulty to identify motifs of the TFs. Therefore, the approach to find the enriched motifs in Nanog data set relative to Sox2 or Oct4 data sets is needed. Here, we use Nanog ChIP-seq data set as the test input and either of Sox2 and Oct4 data sets as the control input. From the results shown in Figure 4(b), the significant predicted motifs of these two compared data sets are similar, and both of them are also similar to the previously reported motifs "CCATCA" by [23, 35] as an alternative Nanog motif.

In addition, for Smad1 data set, our algorithm not only discovers a motif "AACAAAGC" matching the published Smad1 motif "AAACAAAG" but also finds other motifs, like "CCTTTGTC", which matches a Sox2 motif (Figure 4(c)). And these discovered motifs demonstrate the frequent cobinding relationship of Smad1 and Sox2 TFs.

In contrast with the traditional analysis of transcriptional regulation that motifs commonly bind to a DNA-binding TF, genome-wide locations for a specific TF usually do not carry the primary or alternative binding motif but the binding motifs for other TFs. To explore this issue, we employ the ChIP-seq approach to characterize these TFs that bind to DNA indirectly through binding to a cofactor. We use a histone acetyltransferase generally found at enhancer regions [27], P300, to reveal the interaction of cofactors and DNA-binding TFs, and hope to infer the potential tissues of transcription regulation. From the results, we found that P300 does not only associate with the notable Oct4, Sox2, Nanog, and Smad1 TFs but also cooccurs with Oct-Sox complex and other abundant TFs including a TEF motif "AGGATTGCT" and the core of AP4-L motif "CAGCAGG." In addition, there are still several motifs found by our algorithm in relative low probabilities, such as Esrrb motif "GAgGGTgA," Klf4 motif "GGTGTGGg," and Tcfcp2l1 motif "CCAGTTgcA."

The results of experiment show that our algorithm makes a good trade-off between accuracy and efficiency. It shows better performance than the other compared algorithms. Data of several megabytes can be handled in a few minutes; when data is up to 50 Mb, it can be handled in several hours. We can see in the word enumeration that FCmotif counts all the l-mers in both test and control sets. Suppose the both sets have the same size: the sequence length n and the number of the sequences t, and the motif length is l, so the computational complexity of counting all the l-mers is $O(ntl)$ which is completely acceptable. The step of searching emerging substrings dramatically reduce the number of potential motif instances; generally, the order of magnitude of emerging substrings is $O(10^2)$. Moreover, note that the range of l is from 6 to 12 bps, the (l, d) values include $(6, 1)$, $(7, 1)$, $(8, 1)$, $(9, 2)$, $(10, 2)$, $(11, 2)$, and $(12, 3)$, and the number of the neighborhood instances E is

$$E = C_l^1 3 + C_l^2 3^2 + \cdots + C_l^d 3^d. \tag{14}$$

So the maximum E is 6570 for $(l, d) = (12, 3)$, which means our algorithm does not need to search thousands of possible instances to form PWM and can limit the number of enriched l-mers in several dozens. For a few megabytes data, our algorithm can search out a fixed length motif in a few minutes with the amount of computation in $O(10^8)$ or $O(10^9)$ (for a large l), which is more faster than that of the probability training methods. In addition, recent studies indicate that many regulatory regions are located in transposable elements which are commonly not conserved [36].

4. Conclusions

In this paper, we propose a fast cluster motif finding algorithm, named FCmotif, to solve the (l, d) motif identification problem in large scale ChIP data set. FCmotif overcomes drawbacks of traditional algorithms which are time-consuming and cannot handle the full size data; it guarantees to find all potential (l, d) motif instances. FCmotif utilizes a word enumeration strategy and searches the neighborhood instances to form the PWM of enriched substrings. It breaks up constrain of the OOPS model and can find long motifs by clustering the PWM in different lengths. The experiments of the ChIP-seq data sets from mouse ES cells confirm that FCmotif can find not only the primary motif but also the exceptional motifs, which uses less time compared to popular motif discovery algorithms. Meanwhile, the potential cobinding relationship can be also detected by our algorithm. It is worth noting that our algorithm is easy to parallel because the calculation of motif of each length is independent.

In summary, it can be seen that FCmotif is a competitive algorithm to deal with the (l, d) motif finding in the ChIP-seq data. The functions of some motifs found by our algorithm are still unknown. The functions of some motifs found by our algorithm are still unknown, the further experimental validation is needed to prove that these motifs are indeed functional. The analysis of motifs in these complex transcriptional regions is the key issue for the future study.

Disclosure

Ping Wang is the second author of this paper.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (Grant nos. 310824151037, 310832151088, and 310832151091) and the Natural Science Foundation of Shaanxi (2015JM6280).

References

[1] P. A. Pevzner and S.-H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences," in *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, vol. 8, pp. 269–278, 2000.

[2] F. Zambelli, G. Pesole, and G. Pavesi, "Motif discovery and transcription factor binding sites before and after the next-generation sequencing era," *Briefings in Bioinformatics*, vol. 14, no. 2, pp. 225–237, 2013.

[3] E. R. Mardis, "ChIP-seq: welcome to the new frontier," *Nature Methods*, vol. 4, no. 8, pp. 613–614, 2007.

[4] P. J. Park, "ChIP-seq: advantages and challenges of a maturing technology," *Nature Reviews Genetics*, vol. 10, no. 10, pp. 669–680, 2009.

[5] P. Collas and J. A. Dahl, "Chop it, ChIP it, check it: the current status of chromatin immunoprecipitation," *Frontiers in Bioscience*, vol. 13, no. 17, pp. 929–943, 2008.

[6] H. S. Rhee and B. F. Pugh, "Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution," *Cell*, vol. 147, no. 6, pp. 1408–1419, 2011.

[7] C. Jia, M. B. Carson, Y. Wang, Y. Lin, and H. Lu, "A new exhaustive method and strategy for finding motifs in ChIP-enriched regions," *PLoS ONE*, vol. 9, no. 1, Article ID e86044, 2014.

[8] X. Shirley Liu, D. L. Brutlag, and J. S. Liu, "An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments," *Nature Biotechnology*, vol. 20, no. 8, pp. 835–839, 2002.

[9] P. Machanick and T. L. Bailey, "MEME-ChIP: motif analysis of large DNA datasets," *Bioinformatics*, vol. 27, no. 12, Article ID btr189, pp. 1696–1697, 2011.

[10] J. E. Reid and L. Wernisch, "STEME: efficient em to find motifs in large data sets," *Nucleic Acids Research*, vol. 39, no. 18, article e126, 2011.

[11] M. Hu, J. Yu, J. M. G. Taylor, A. M. Chinnaiyan, and Z. S. Qin, "On the detection and refinement of transcription factor binding sites using ChIP-Seq data," *Nucleic Acids Research*, vol. 38, no. 7, pp. 2154–2167, 2010.

[12] I. V. Kulakovskiy, V. A. Boeva, A. V. Favorov, and V. J. Makeev, "Deep and wide digging for binding motifs in ChIP-Seq data," *Bioinformatics*, vol. 26, no. 20, pp. 2622–2623, 2010.

[13] M. Thomas-Chollier, C. Herrmann, M. Defrance, O. Sand, D. Thieffry, and J. Van Helden, "RSAT peak-motifs: motif analysis in full-size ChIP-seq datasets," *Nucleic Acids Research*, vol. 40, no. 4, article e31, 2012.

[14] A. A. Sharov and M. S. H. Ko, "Exhaustive search for over-represented DNA sequence motifs with cisfinder," *DNA Research*, vol. 16, no. 5, pp. 261–273, 2009.

[15] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, "Bayesian models for multiple local sequence alignment and Gibbs sampling strategies," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1156–1170, 1995.

[16] R. Staden, "Methods to define and locate patterns of motifs in sequences," *Computer Applications in the Biosciences*, vol. 4, no. 1, pp. 53–60, 1988.

[17] M. L. Bulyk, P. L. F. Johnson, and G. M. Church, "Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors," *Nucleic Acids Research*, vol. 30, no. 5, pp. 1255–1261, 2002.

[18] P. V. Benos, M. L. Bulyk, and G. D. Stormo, "Additivity in protein-DNA interactions: how good an approximation is it?" *Nucleic Acids Research*, vol. 30, no. 20, pp. 4442–4451, 2002.

[19] M. T. Lee, M. L. Bulyk, G. A. Whitmore, and G. M. Church, "A statistical model for investigating binding probabilities of DNA nucleotide sequences using microarrays," *Biometrics*, vol. 58, no. 4, pp. 981–988, 2002.

[20] J. Fischer, V. Heun, and S. Kramer, "Optimal string mining under frequency constraints," in *Knowledge Discovery in Databases: PKDD 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds., vol. 4213 of *Lecture Notes in Computer Science*, pp. 139–150, Springer, Berlin, Germany, 2006.

[21] Y. Zhang, H. Huo, and Q. Yu, "A heuristic cluster-based em algorithm for the planted (l, d) problem," *Journal of Bioinformatics and Computational Biology*, vol. 11, no. 4, Article ID 1350009, 19 pages, 2013.

[22] Q. Yu, H. Huo, X. Chen, H. Guo, J. S. Vitter, and J. Huan, "An efficient motif finding algorithm for large DNA data sets," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM '14)*, pp. 397–402, IEEE, Belfast, UK, November 2014.

[23] M. J. Mason, K. Plath, and Q. Zhou, "Identification of context-dependent motifs by contrasting ChIP binding data," *Bioinformatics*, vol. 26, no. 22, pp. 2826–2832, 2010.

[24] T. L. Bailey, M. Bodén, T. Whitington, and P. Machanick, "The value of position-specific priors in motif discovery using MEME," *BMC Bioinformatics*, vol. 11, article 179, 2010.

[25] S. Georgiev, A. P. Boyle, K. Jayasurya, X. Ding, S. Mukherjee, and U. Ohler, "Evidence-ranked motif identification," *Genome Biology*, vol. 11, no. 2, article R19, 2010.

[26] C. T. Harbison, D. B. Gordon, T. I. Lee et al., "Transcriptional regulatory code of a eukaryotic genome," *Nature*, vol. 431, no. 7004, pp. 99–104, 2004.

[27] X. Chen, H. Xu, P. Yuan et al., "Integration of external signaling pathways with the core transcriptional network in embryonic stem cells," *Cell*, vol. 133, no. 6, pp. 1106–1117, 2008.

[28] P. Cartwright, C. McLean, A. Sheppard, D. Rivett, K. Jones, and S. Dalton, "LIF/STAT3 controls ES cell self-renewal and pluripotency by a Myc-dependent mechanism," *Development*, vol. 132, no. 5, pp. 885–896, 2005.

[29] J. Jiang, Y.-S. Chan, Y.-H. Loh et al., "A core Klf circuitry regulates self-renewal of embryonic stem cells," *Nature Cell Biology*, vol. 10, no. 3, pp. 353–360, 2008.

[30] N. Ivanova, R. Dobrin, R. Lu et al., "Dissecting self-renewal in stem cells with RNA interference," *Nature*, vol. 442, no. 7102, pp. 533–538, 2006.

[31] J. Kim, J. Chu, X. Shen, J. Wang, and S. H. Orkin, "An extended transcriptional network for pluripotency of embryonic stem cells," *Cell*, vol. 132, no. 6, pp. 1049–1061, 2008.

[32] M. Thomas-Chollier, M. Defrance, A. Medina-Rivera et al., "RSAT 2011: regulatory sequence analysis tools," *Nucleic Acids Research*, vol. 39, supplement 2, pp. W86–W91, 2011.

[33] T. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," in *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB '94)*, August 1994.

[34] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in DNA sequences," *Bioinformatics*, vol. 17, supplement 1, pp. S207–S214, 2001.

[35] T. L. Bailey, "DREME: motif discovery in transcription factor ChIP-seq data," *Bioinformatics*, vol. 27, no. 12, Article ID btr261, pp. 1653–1659, 2011.

[36] G. Bourque, B. Leong, V. B. Vega et al., "Evolution of the mammalian transcription factor binding repertoire via transposable elements," *Genome Research*, vol. 18, no. 11, pp. 1752–1762, 2008.