GSMC: Combining Parallel Gibbs Sampling with Maximal Cliques for Hunting DNA Motif

CHAO PEI,¹ SHU-LIN WANG,¹ JIANWEN FANG,² and WEI ZHANG¹

ABSTRACT

Regulatory elements are responsible for regulating gene transcription. Therefore, identification of these elements is a tremendous challenge in the field of gene expression. Transcription factors (TFs) play a key role in gene regulation by binding to target promoter sequences. A set of conserved sequence patterns with a highly similar structure that is bound by a TF is called a motif. Motif discovery has been a difficult problem over the past decades. Meanwhile, it is a foundation stone in meeting this challenge. Recent advances in obtaining genomic sequences and high-throughput gene expression analysis techniques have enabled the rapid development of computational methods for motif discovery. As a result, a large number of motif-finding algorithms aiming at various motif models have sprung up in the past few years. However, most of them are not suitable for analysis of the large data sets generated by next-generation sequencing.

To better handle large-scale ChIP-Seq data and achieve better performance in computational time and motif detection accuracy, we propose an excellent motif-finding algorithm known as GSMC (Combining Parallel Gibbs Sampling with Maximal Cliques for hunting DNA Motif). The GSMC algorithm consists of two steps. First, we employ the commonly used Gibbs sampling to generating initial motifs. Second, we utilize maximal cliques to cluster motifs according to Similarity with Position Information Contents (SPIC). Consequently, we raise the detection accuracy in a great degree, in the meantime holding comparative computation efficiency. In addition, we can find much more credible cofactor interacting motifs.

Keywords: DNA motif, maximal cliques, Gibbs sampling, SPIC, cofactor motif.

1. INTRODUCTION

AN IMPORTANT GOAL OF BIOLOGICAL RESEARCH to understand the mechanisms that regulate gene expression (Stegmaier et al., 2013). Identification of regulatory elements, especially the binding sites in DNA for transcription factors (TFs), is a vital task in this goal (Das and Dai, 2007). TFs are proteins that bind to DNA, typically upstream from and close to the transcription start site of a gene. The expression of that gene is regulated by activating or inhibiting the transcription mechanism (Tompa et al., 2005). Moreover,

¹College of Computer Science and Electronics Engineering, Hunan University, Changsha, China.

²Biometric Research Program, Division of Cancer Treatment and Diagnosis, National Cancer Institute, Rockville, MD 20850.

transcription regulation is usually triggered by the binding of TFs to specific DNA segments that are known as TF-binding sites (TFBSs) (Zhang and Chen, 2016). A motif is defined as a set of binding sites recognized by one TF. These binding sites are mutually similar, which happens to explain the general characteristics of the motif. Therefore, motif discovery is a significant step in the process of unraveling the mechanism of gene expression. At the most basic level, the motif-finding problem is the task of identifying recurring patterns of conserved short strings that appear in a set of DNA sequences (promoter region upstream from and close to the transcription start site of a gene). Usually, these patterns are fairly short (5 to 20 bps long).

A genome-wide ChIP study produces thousands or more DNA fragments consisting of several hundred base pairs, which cover the binding sites for a TF (Ikebata and Yoshida, 2015). Providing a set of DNA fragment sequences associated with a TF, by comparing the motifs discovered from these sequences with the known TF-binding motifs in a TFBSs database, for example, JASPAR (Sandelin et al., 2004), TRANSFAC (Wingender et al., 2000), we can not only recognize the binding sites for the ChIPed TF but also recognize the cofactor motifs that regulate the TF activity (Smith et al., 2005; Bailey, 2011; Goi et al., 2013).

The majority of early motif-finding algorithms can be divided into two categories. One is word-based (based on the string) methods that mostly adopt exhaustive enumeration, such as Weeder (Pavesi et al., 2001); the other is probabilistic-based (model-based) algorithms, for instance, MEME [multiple EM (expectation maximization) for motif elicitation] (Bailey and Elkan, 1994), AlignACE (Hughes et al., 2000), and ANN-Spec (Workman and Stormo, 2000). The word-based enumerative methods ensure global optimality, and they are appropriate for short motifs finding. Therefore, they are frequently used for motif finding in eukaryotic genomes where motifs are usually shorter than prokaryotes (Das and Dai, 2007). However, the methods cited earlier are not suitable for handling ChIP-seq data. Some of them have undergone reconstruction, arising from many ChIP-tailored algorithms. STEME (suffix free EM for motif elicitation) (Reid and Wernisch, 2011), a ChIP-tailored version of MEME, utilizes a branch-and-bound technology to remove negligible oligomers with significantly low probabilities effectively. For the sake of reducing the computational load in the counting operation, DREME (motif discovery in transcription factor—ChIP-seq data) (Bailey, 2011) and CisFinder (Sharov and Ko, 2009) adopt similar strategies, taking the risk of missing vital motifs in earlier steps of the recursion. Although Hegma (Ichinose et al., 2012) is competitive among current algorithms, the degradation of its detection accuracy is nonignorable.

The model-based approaches are mostly based on the EM (expectation maximization) algorithm (Bailey and Elkan, 1994) or Gibbs sampling (Lawrence et al., 1993). The RPMCMC (repulsive parallel MCMC algorithm for discovering diverse motifs) algorithm is a parallel version of the commonly used Gibbs sampling, running on a parallel-interacting Gibbs sampler. When the routes of different sampling chains are close to each other, a repulsive force (defined as a function of position probability matrices [PPMs]) will drag them forward in different directions. As a result, different sampling chains are urged to seek various regions so that the RPMCMC is more likely than old methods to discover much more diverse motifs. For old and other recent new motif discovery methods, we make a summary that they commonly exhibit low detection accuracy, are instable, and are time consuming. Thus, we are devoted to overcoming these drawbacks and improving them. As with the RPMCMC algorithm, we use parallel mutually exclusive Gibbs sampling to obtain the initial motif. When another motif similarity measure method and motif clustering method are applied to motif screening, we achieve the same or even better performance as the RPMCMC algorithm. These two methods will be demonstrated later.

The main purpose of this study is to derive a novel motif discovery algorithm, which acquires higher detection accuracy while holding competitive computational efficiency. Moreover, the novel method can detect much more believable diverse motifs. To cater to these needs, we propose a new motif-finding algorithm known as GSMC, based on Parallel Gibbs sampling (Lawrence et al., 1993) and Maximal Cliques (Zhang and Chen, 2016) clustering. We adopt parallel-interacting Gibbs sampling for generating initial motifs and then we utilize Maximal Cliques to group initial motifs into different clusters according to Similarity with Position Information Contents (SPIC) (Zhang et al., 2013) among motifs. Finally, we consider the first motif in each cluster as an output motif. On the basis of the ZOOPS (Zero or one motif occurrences per dataset sequence) sequence model, we design the GSMC algorithm without the OOPS constraint and motif length limit (Yu et al., 2015). We implement the GSMC algorithm with C++. When a set of TF ChIP-seq datasets of the ENCODE project (Dunham et al., 2012) is used to compare the performance of GSMC and two other high-performance algorithms, the results show that GSMC happens to make up for the other two algorithms on 20 ChIP-seq data sets for motif discovering; therefore, GSMC is a practical and alternative method in the field of motif finding.

2. MATERIALS AND METHODS

2.1. Gibbs sampling model

The parallel Gibbs sampling algorithm applies several parallel-interacting Gibbs samplers to produce PPMs. Here, we provide an overview of the Gibbs sampling model.

Let $S = (S_1, \dots, S_N)$ denote a dataset of sequences, where N is equal to the number of sequences in the dataset. Each sequence is over the alphabet set $\{A, C, G, T\}$. We look within each sequence for mutually similar segments of specified width W . The algorithm maintains two evolving data structures. The first is the pattern description, in the form of a probabilistic model of residue frequencies for each position i from 1 to W . Residue frequencies consist of the variables $p_{i,a}, p_{i,c}, p_{i,g}, p_{i,t}$. This pattern description is accompanied by an analogous probabilistic description of the “background frequencies” q_a, q_c, q_g, q_t and they denote the occurrence frequency of each residue in background sequences. The second data structure is a set of position u_k ($k=1, \dots, N$), for the common pattern within the sequences. Given a set of input sequences, the objective is to identify the “best,” defined as the most probable, common pattern. This pattern is achieved by locating the alignment that maximizes the ratio of the corresponding pattern probability to background probability (Lawrence et al., 1993).

Initialization of the algorithm is to choose random starting positions in all sequences and to write them into starting positions set $SS = \{ss_1, \dots, ss_N\}$. It then proceeds via the iterative execution of the following two steps:

Step one: predictive update step. One of the N sequences, S_i is selected randomly or sequentially. The pattern description $p_{i,j}$ and background frequencies q_j are then calculated, as described in Equation (1), from the current position u_k in all sequences excluding S_i .

Let $c_{i,j}$ be the count of nucleotide j in the position i . For the i th position of the pattern, we have $N-1$ observed amino acids, because S_i has been excluded. Bayesian statistical analysis suggests that, for the purpose of pattern estimation, these $c_{i,j}$ should be supplemented with residue-dependent “pseudocounts” b_j to yield pattern probabilities

$$p_{i,j} = \frac{c_{i,j} + b_j}{N - 1 + B} \quad (1),$$

where B is the sum of the b_j . The q_j is calculated analogously, with the corresponding counts taken over all nonpattern positions.

Step two: sampling step. In the sequence S_i , each segment of width W is considered a possible instance of the pattern. $Score_e(i, j)$ shows the possibility of the segment x from position j to $j+W-1$ in sequence S_i to be a pattern instance. $Score_b(i, j)$ represents the possibility of the segment x from position j to $j+W-1$ in sequence S_i to be a background sequence. The weight $r_{i,j}$ is assigned to the segment x , and with each segment so weighted, a random one is selected. Its position then becomes the new ss_i . All the $r_{i,j}$, $Score_e(i, j)$, and $Score_b(i, j)$ of these segments are calculated according to the pattern description $p_{i,j}$ and background frequencies q_j . The calculation formulas of $r_{i,j}$, $Score_e(i, j)$, and $Score_b(i, j)$ are described by Equations (2), (3), and (4).

$$Score_e(i, j) = Score_e(S_i[j, j+W-1]) = \prod_{t=0}^{W-1} p_{t, S_{i,t+j}} \quad (2)$$

$$Score_b(i, j) = Score_b(S_i[j, j+W-1]) = \prod_{t=0}^{W-1} q_{S_{i,t+j}} \quad (3)$$

$$r_{i,j} = \frac{Score_e(i, j)}{Score_b(i, j)} \quad (4)$$

After normalization, $r_{i,j}$ gives the probability that the pattern instance in sequence S_i belongs at position j . The algorithm finds the most probable alignment by selecting a set of u_i that maximizes the product of these ratios. Likewise, the algorithm finds the one that may maximize F , the sum of the logarithms of these ratios. In the notation developed earlier, F is indicated by Equation (5).

$$F = \sum_{i=1}^W \sum_{j \in \{a, c, g, t\}} c_{i,j} \log \frac{p_{i,j}}{q_j} \quad (5)$$

2.2. GSMC algorithm

The GSMC algorithm is composed of two parts, which include motif generating and postprocessing. GSMC shares some similarities with RPMCMC, especially in the phase of motif generating. First, we

utilize parallel-interacting Gibbs sampling to produce various motifs. Second, based on SPIC and information content (IC), we employ maximal clique clustering to group motifs into different categories. Finally, we select the optimal motif in each category as the final output motif.

2.2.1. Motif generation. We use the ZOOPS model (Bailey and Elkan, 1994) in which there is zero or one motif occurring per sequence. As the input sequences, $S^+ = \{s_1^+, \dots, s_n^+\}$ and the length of them is L_i ($i = 1, \dots, n$). $S^- = \{s_1^-, \dots, s_n^-\}$, which is the reverse complement of S^+ . Our model uses the set of n concatenated sequences, $S = \{s_1, \dots, s_n\}$, where $s_i = (s_i^+, s_i^-)$ ($i = 1, \dots, n$). Whether there is a motif instance in sequence s_i or not, it depends on the value of z_i (one or zero). If $z_i = 1$, a K -mer motif exists in sequence s_i , starting at the site $u_i \in \{1, \dots, L_i - K + 1, L_i + 1, \dots, 2L_i - K + 1\}$. We utilize PPM to represent the motif. The size of PPM is $4 \times K$ because the length of $\{a, c, g, t\}$ is four, where K denotes the length of motif. Let $\theta_k = (\theta_{k,a}, \theta_{k,c}, \theta_{k,g}, \theta_{k,t})^T$ indicate the kst position of motif and $\Theta = (\theta_1, \dots, \theta_k)$ represent the motif. Let $\theta_0 = (\theta_{0,a}, \theta_{0,c}, \theta_{0,g}, \theta_{0,t})^T$ be probability vector of the background sequences.

Provided the input sequences S , the objective is to detect the PPM Θ without the knowledge of the motif length K and the background probability vector θ_0 where the latent variables contain $U = \{u_1, \dots, u_n\}$ and $Z = \{z_1, \dots, z_n\}$.

As shown in Table 1 and Table 2, motif generation comprised two parts, which include initialization and updating. The stage of updating relies on Gibbs sampling.

The number of replicas, n_motif set by the user is equal to the number of Gibbs samplers. When the number of iterations comes to the value of *burnin*, a number of exclusive Gibbs samplers start to sample and save data. Once on the procedure running for *iter* times, the procedure of motif generation terminates.

2.2.2. Postprocessing: clustering motifs. After accomplishing the section of motif generation, we can attain massive motif PPMs. Nevertheless, most of them are either false or redundant motifs. It is essential to filter and cluster PPMs for picking out true motifs.

First of all, we get rid of many false motifs according to their *IC* value less than a cutoff value, which is given by Formula (6):

$$IC = \sum_{i=1}^K \sum_{j \in \{a, c, g, t\}} p_{i,j} \log \frac{p_{i,j}}{q_j} \quad (6)$$

Then, we use the SPIC metric to compute the similarity score between each pair of rest motifs. As is shown later, the SPIC metric is given by Formulas (7), (8), (9), and (10). For a motif M_x containing n_x sequences with length L_x , let $F_x = (f_x(b, X))_{4 \times L_x}$ be its position frequency matrix (PFM) and P_x be its position-specific scoring matrix (PSSM) that is defined as

$$P_x = (P_x(b, X))_{4 \times L_x} = \left(\log \frac{p_x(b, X)}{q_x(b)} \right)_{4 \times L_x} \quad (7),$$

TABLE 1. INITIALIZATION FUNCTION
FOR INITIALIZING $K, \theta, \theta_0, U, Z$

Algorithm 1: Initialization function

Input: a set of sequences $S, k_{max}, k_{min}, markov, n_motif$

Output: $K, \theta, \theta_0, U, Z$

1: $K \leftarrow rand(\) \% (k_{max} - k_{min} + 1) + k_{min}$

2: **for** $j \leftarrow 0$ to 3 **do**

3: $tmp[j] = gamma_rand(1, 1)$

4: $\theta \leftarrow tmp[j] / (tmp[0] + tmp[1] + tmp[2] + tmp[3])$

5: $\theta_0 \leftarrow pow(0.25, markov)$

6: **for** $p \leftarrow 0$ to the size of S **do**

7: **for** $i \leftarrow 0$ to n_motif **do**

8: $U \leftarrow rand(\) \% (S[p].length(\) - K[i] - 1) + 1$

9: $Z \leftarrow 0$

10: **return** $K, \theta, \theta_0, U, Z$

TABLE 2. UPDATING FUNCTION
FOR SAMPLING AND SAVING MOTIFS

Algorithm 2: Updating function

Input: $K, \theta, \theta_0, U, Z, n_motif, iter, burnin$

Output: the set of PPMs Θ

1: **for** $j \leftarrow 0$ to $iter$ **do**

2: **for** $i \leftarrow 0$ to n_motif **do**

3: $update(\theta_i)$

4: $update(\theta_{0i})$

5: $update(K_i)$

6: $update(U)$

7: $update(Z)$

8: **if** $j \geq burnin$ **then**

9: add θ to Θ

10: **return** Θ

PPMs, position probability matrices.

where $f_x(b, X)$ and $P_x(b, X)$ are the count and probability of base $b \in \{A, C, G, T\}$ appearing at position X of M_x (i.e., column X of P_x), respectively, and $q_x(b)$ is the probability of base b appearing in the background sequences. A pseudo-count is added when computing these probabilities. The IC of column X of the PSSM P_x is defined as

$$I(X, P_x) = \sum_{b \in \{A, C, G, T\}} p_x(b, X) P_x(b, X) = \sum_{b \in \{A, C, G, T\}} p_x(b, X) \log \frac{p_x(b, X)}{q_x(b)} \quad (8).$$

For two motifs M_1 and M_2 with PSSMs P_1 and P_2 , and PFMs F_1 and F_2 , respectively, the similarity score between position X of M_1 and position Y of M_2 is defined as

$$Sim(M_1(X), M_2(Y)) = \min \left\{ 1, \frac{\max\{score(P_1(X), F_2(Y)), score(P_2(Y), F_1(X))\}}{\max\{score(P_1(X), F_1(X)), score(P_2(Y), F_2(Y))\}} \right\} \quad (9),$$

where

$$score(P_x(A), F_y(B)) = I(A, P_x) \sum_b \left(f_y(b, B) \cdot \log \frac{p_x(b, A)}{q_x(b)} \right), x, y \in \{1, 2\}, A, B \in \{X, Y\} \quad (10).$$

Further, we construct a motif similarity graph on the basis of similarity score between each pair of motifs. In the graph, each node stands for a motif. Two nodes will be linked by an edge if and only if the weight value is greater than a preset threshold. The weight value of two connected motifs is the similarity score between them. More specifically, binding site motifs that belong to the same TF are more likely to form highly connected sub-graphs with high edge weights in the motif similarity graph than are those from different TFs or spurious motifs. Therefore, if one motif has a low similarity score with any additional motifs, there is a strong possibility that this motif is spurious. These motifs will be deleted from the graph.

FIG. 1. An example of finding maximal clique associated with node v .

Finally, we utilize Maximal Cliques to cluster motifs on the basis of the motif similarity graph. Clustering motifs via Maximal Cliques consists of four steps. A brief introduction of the four steps is given next.

Step 1: For each node, find a maximal clique associated with it, as depicted in Figure 1. First, sort the neighbor nodes of v in ascending order by the weights of their edges incident to v to get an array $\{E, D, C, B, A\}$. Second, successively delete the nodes in the array as well as their incident edges until v and the remaining nodes have the same degree. Finally, for each deleted node in the reverse order, if the node connects with all the remaining nodes in the initial graph, add the node and incident edges in the initial graph.

Step 2: Merge cliques into clusters. Initially, sort all the unassigned cliques in descending order by the sum of edge weights. Then, compare the first clique with any other unassigned clique (C_i and C_j), if the overlap ratio of the nodes in C_i appearing in C_j divided by C_i is no less than a preset threshold α or if the overlap ratio of the nodes in C_j appearing in C_i divided by C_j is no less than a preset threshold β , merge C_j into C_i , and repeat the previous steps until all of the cliques in the queue are assigned.

Step 3: Delete redundant nodes. Compare the latter cliques with the former, and remove the nodes in the latter cliques appearing in the fore.

Step 4: Sort clusters. Sort all the clusters in a decreasing order of edge-weight sum. Choose the first motif in each cluster as the final output motif.

2.3. Performance assessment

We report the performance of three motif discovery algorithms on the dataset: 20 TF ChIP-seq datasets of the ENCODE project. ChIP-seq experiments produce a great amount of DNA segments, which contain many TFBSs bound to a certain TF. In addition to finding these TFBSs, we can discover some cofactor motifs that are involved in the regulatory module of the primary TF. The performance evaluation of the algorithm is outlined next.

First of all, to compare motif detection accuracy among three algorithms, each predicted motif (PPM) by them is matched to JASPAR CORE motifs by using the online TOMTOM tool (Gupta et al., 2007). For a given predicted PPM, TOMTOM outputs the matching scores to all annotated TFBSs (the name of TFBSs) in JASPAR with the statistical significance (E-values: the expected number of false positives in the matches). For each algorithm, a diversity of the discovered motifs is evaluated with the number of known motifs in JASPAR CORE that are matched significantly to the produced PPMs with the acceptable level of significance at E-value less than

TABLE 3. DEFAULT PARAMETERS OF GSMC AND RPMCMC THAT WERE USED IN ALL EXPERIMENTS, DREME WAS EXECUTED USING THE DEFAULT SETTINGS

<i>GSMC</i>	
<i>Parameter</i>	<i>Value</i>
Max/min motif length	$L_{min} = 6, L_{max} = 14$
No. of replicas	$M = 20$
No. of iterations	$N = 70$
Burn-in period	Burning = 20
IC threshold of motif	$IC_{cutoff} = 7$
Similarity threshold of motif	$Similarity_{cutoff} = 0.6$
<i>RPMCMC</i>	
<i>Parameter</i>	<i>Value</i>
Max/min motif length	$L_{min} = 6, L_{max} = 14$
No. of replicas	$M = 20$
No. of iterations	$N = 70$
Burn-in period	Burning = 20
Motif clustering	$\lambda = 0.3$

DREME; GSMC, Combining Parallel Gibbs Sampling with Maximal Cliques for hunting DNA Motif; IC, information content; RPMCMC.

0.05 (Ikebata and Yoshida, 2015). The less the E-value, the higher the detection accuracy. For advanced options of TOMTOM program, we choose Pearson correlation coefficient as the motif column comparison function and set the significance threshold at 0.05. Then, we compare the total running time between the RPMCMC algorithm and our GSMC algorithm. Finally, we make a comparison of the number of cofactor motifs among three algorithms.

2.4. Programs and parameter selection

The RPMCMC program used in the article was released on January 6, 2015 (<http://dawe.ism.ac.jp/yoshidalab/motif>). The DREME program is available online (<http://meme-suite.org/tools/dreme>). Two motif discovery algorithms (RPMCMC and GSMC) were compiled and installed on 64-bit Ubuntu 14.04 in VMware. The DREME program is directly utilized online. To ensure a comparison that is as fair as possible among the three motif discovery algorithms, the values of the adjustable parameters in RPMCMC and GSMC were selected so as to obtain the optimal performance respectively. For the RPMCMC algorithm, we keep a default value of some parameters and change the others. We modify the number of iterations to 70 and the number of replicas to 20. For the DREME algorithm, we keep the value of all parameters unchanged because it is available online from the site <http://meme-suite.org/tools/dreme>. For the GSMC algorithm, the number of iterations and replicas are the same as the RPMCMC's. As a section of the GSMC program, clustering motifs are involved in several parameters, such as motif similarity cutoff, IC threshold, α , and β . In agreement with Zhang and Chen (2016), the values of parameters α and β are set to (0.5 0.5) and the value of the similarity cutoff parameter is set to 0.6. As the third section describes, the IC threshold parameter is sampled from 6 to 8.5 in steps of 0.5 and we choose 7 as its best value. All parameters of GSMC and RPMCMC are given in Table 3.

FIG. 2. Optimal value selection of the parameter IC threshold. (A) The IC thresholds are selected from 6 to 8.5 in steps of 0.5. For each IC threshold value, a ChIP dataset (wgEncodeSydhTfbsHepg2Nrf1IggrabPk) is tested 16 times. Every time, we choose the minimal E-value of all meeting matches between the predicted motifs and the motif NRF1. Then, we draw them in broken lines. (B) For each IC threshold value, we calculate the average E-value of 16 minimal E-values and draw these average E-values in broken lines. (C) For each IC threshold value, we compute the mean-square deviation of the 16 minimum E-values and plot them in the same way. IC, information content.

FIG. 3. Comparison of motifs' detection accuracy among RPMCMC, GSMC, and DREME on datasets (wgEncodeSydhTfbsHepg2Nrf1IggrabPk). DREME; GSMC, Combining Parallel Gibbs Sampling with Maximal Cliques for hunting DNA Motif; RPMCMC.

3. RESULTS AND DISCUSSION

3.1. Parameter optimization and performance comparison

Empirically, we set the value of the similarity cutoff parameter to 0.6 and the values of parameters α and β to (0.5 0.5). For the sake of optimizing the value of the IC threshold parameter, we choose one TF ChIP-seq dataset as the parameter test dataset and make 16 times experiments of the dataset for each IC threshold value from 6 to 8.5 in steps of 0.5 whereas the remaining values of other parameters are unchanged. Motif E-values of different IC thresholds are presented in Figure 2A. Average E-values of discrete IC thresholds are given in Figure 2B. Figure 2C describes the E-values' mean-square deviation of different IC thresholds. The three figures are indicated next.

According to Figure 2B, we can refer to the relationship between the average detection accuracy and the IC threshold value. The less the average E-value, the higher the average detection accuracy. From Figure 2C, we know the fluctuations of the 16 minimal E-values. The less the E-values' mean-square deviation, the higher the stability. Above all, 7 is chosen as the optimal value of the parameter IC threshold in that there is the average least E-value and the second least mean-square deviation in this case.

3.2. Performance on discovering motifs for ENCODE ChIP-Seq datasets

We use the same ENCODE datasets as the paper (Ikebata and Yoshida, 2015). They are SYDH TFBS narrowPeak files (available from NCBI's Gene Expression Omnibus using the accession number GSE31477), and we can download them from <http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeSydhTfbs/>. We choose 20 datasets with different sizes from the 228 ENCODE ChIP-seq datasets as our experimental data. GSMC is implemented in C++, RPMCMC is available on the authors' websites, and DREME can be used online directly. All the tests are conducted on Intel® Core™ i3 processor with 4-core CPUs and 4 GB of main memory. On the virtual platform of Ubuntu 14.04, we run GSMC and RPMCMC for 20 datasets, record running time, and save all the produced PPMs. With the help of the TOMTOM, we compare the produced PPMs with all the known motifs in JASPAR and leave the least E-value of matching between the ChIPed motif and the query motifs. Then, we make comparisons among GSMC, RPMCMC,

FIG. 4. Comparison of motifs' detection accuracy among RPMCMC, GSMC, and DREME on 20 chosen datasets.

FIG. 5. Comparison of computational efficiency between GSMC and RPMCMC. Running time is the CPU execution time.

and DREME in terms of detection accuracy, computational speeds, and the number of cofactor motifs discovered. From a performance perspective, our GSMC algorithm makes up for rather than takes the place of existing DNA motif-finding methods for analyzing the large ChIP-seq data.

3.2.1. Detection accuracy. Each dataset corresponds to a ChIPed TF, which has a relevant motif. We run each dataset for 50 times and each time, we make a note of the least E-value of all meeting matches (E-value < 0.05) that are aimed at the ChIPed motif. Figure 3 illustrates the comparison of detection accuracy among three algorithms on a ChIP dataset (wgEncodeSydhTfbsHepg2Nrf1IggrabPk) in which the binding sites of NRF1 were studied in cell HepG2. When it comes to detection accuracy, DREME surpasses the others and GSMC is nearly equal to RPMCMC.

In Figure 3, we have made a comparison of detection accuracy among three algorithms on one dataset. Next, we make comparisons on all chosen datasets. Figure 4 provides the comparison of GSMC with RPMCMC and DREME on 20 chosen datasets. For most datasets, GSMC and RPMCMC are superior in detection accuracy to DREME.

3.2.2. Computational speeds. Figure 5 displays the computational time for GSMC and RPMCMC on 20 datasets. Each dataset is tested for 16 times, and we take the arithmetic mean as the running time. In terms of computation efficiency, GSMC is comparable to RPMCMC. Motif producing of GSMC basically conforms to RPMCMC. Therefore, the bottleneck calculating the posterior probabilities of the motif start sites u_i in RPMCMC still exists in GSMC. However, the clustering section of GSMC is superior to RPMCMC in terms of flexibility and scalability. With these performances, GSMC is more likely to be a high-efficient method in future studies. Though GSMC cannot surpass RPMCMC in computational efficiency, it is a competitive algorithm in the case of computing time.

3.2.3. Comparison on the number of cofactor motifs discovered. A recent study by Chin Lui Goi and Peter Little called cell-type and TF-specific enrichment of transcriptional cofactor motifs in ENCODE ChIP-seq data claims that TF-specific interactions between TFs and cofactors are essential for

FIG. 6. Comparison on the number of cofactor motifs. Each dataset is tested 16 times, and each data point in the figure is the mean value of 16 tests.

transcriptional regulation through recruitment of general transcription machinery to gene promoter regions (Goi et al., 2013). It is also mentioned in the study that some of the cofactor motifs are experimentally verified cofactors and others are potentially novel cofactors. The same can be said of the cofactors discovered in our study. Even with all that, they have evident reference value and directive significance to further studies. Therefore, the ability of detecting cofactor motifs is crucial to a motif-discovering algorithm. Just as shown in Figure 6, the cofactor motifs found by GSMC outnumber those found by RPMCMC on almost all datasets.

Obviously, GSMC is superior in the number of cofactor motifs to RPMCMC. In other words, the ability of GSMC to detect cofactor motifs is better than RPMCMC and basically equal to DREME.

4. CONCLUSION

Many older popular motif finders cannot be applied to handling large datasets generated by ChIP-seq (Reid and Wernisch, 2014). Though a vast amount of novel algorithms spring up for handling huge datasets, most of them attend to computation speed and lose sight of the accuracy of motif detection. To overcome the drawbacks of them, we develop a ChIP-tailored motif discovery tool called GSMC, which hunts for DNA motifs by combining Parallel Gibbs Sampling with Maximal Cliques. In terms of computation time and detection accuracy, GSMC can rival the most recent motif discoverer RPMCMC, which was specifically designed to handle large datasets. Moreover, GSMC is capable of detecting much more known and potential cofactor motifs where GSMC far exceeds RPMCMC. Besides, GSMC is superior in accuracy of motif detection to DREME. On the whole, we present a novel motif finder reconciling computation speed with detection ability. Locating these motifs plays a critical role in identifying transcriptional regulation. We expect that GSMC will have a place in the process of exploring gene expression.

ACKNOWLEDGMENTS

This work was supported by the grants of the National Science Foundation of China (Grant Nos. 61472467, 61672011, and 61471169) and the Collaboration and Innovation Center for Digital Chinese Medicine of 2011 Project of Colleges and Universities in Hunan Province.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

- Bailey, T.L. 2011. DREME: Motif discovery in transcription factor ChIP-seq data. *Bioinformatics* 27, 1653–1659.
- Bailey, T.L., and Elkan, C. 1994. Fitting a mixture model by expectation maximization to discover motifs in bipolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28–36.
- Das, M.K., and Dai, H.K. 2007. A survey of DNA motif finding algorithms. *BMC Bioinformatics* 8, S21.
- Dunham, I., Kundaje, A., Aldred, S.F., et al. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57–74.
- Goi, C.L., Little, P., and Xie, C. 2013. Cell-type and transcription factor specific enrichment of transcriptional cofactor motifs in ENCODE ChIP-seq data. *BMC Genomics* 14, 1–11.
- Gupta, S., Stamatoyannopoulos, J.A., Bailey, T.L., et al. 2007. Quantifying similarity between motifs. *Genome Biol.* 8, R24.
- Hughes, J.D., Estep, P.W., Tavazoie, S., et al. 2000. Computational identification of cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. *J. Mol. Biol.* 296, 1205–1214.
- Ichinose, N., Yada, T., and Gotoh, O. 2012. Large-scale motif discovery using DNA Gray code and equiprobable oligomers. *Bioinformatics* 28, 25–31.

- Ikebata, H., and Yoshida, R. 2015. Repulsive parallel MCMC algorithm for discovering diverse motifs from large sequence sets. *Bioinformatics* 31, 1561–1568.
- Lawrence, C.E., Altschul, S.F., Boguski, M.S., et al. 1993. Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 262, 208–214.
- Pavesi, G., Mauri, G., and Pesole, G. 2001. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* 17, S207–S214.
- Reid, J.E., and Wernisch, L. 2011. STEME: Efficient EM to find motifs in large data sets. *Nucleic Acids Res.* 39, e126.
- Reid, J.E., and Wernisch, L. 2014. STEME: A robust, accurate motif finder for large data sets. *PLoS One* 9, e90735.
- Sandelin, A., Alkema, W., Engstrom, P., et al. 2004. JASPAR: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32, D91–D94.
- Sharov, A.A., and Ko, M.S. 2009. Exhaustive search for over-represented DNA sequence motifs with CisFinder. *DNA Res.* 16, 261–273.
- Smith, A.D., Sumazin, P., Das, D., et al. 2005. Mining ChIP-chip data for transcription factor and cofactor binding sites. *Bioinformatics* 21, I403–I412.
- Stegmaier, P., Kel, A., Wingender, E., et al. 2013. A discriminative approach for unsupervised clustering of DNA sequence motifs. *PLoS Comput.Biol.* 9, e1002958.
- Tompa, M., Li, N., Bailey, T.L., et al. 2005. Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* 23, 137–144.
- Wingender, E., Chen, X., Hehl, R., et al. 2000. TRANSFAC: An integrated system for gene expression regulation. *Nucleic Acids Res.* 28, 316–319.
- Workman, C., and Stormo, G. 2000. ANN-Spec: A method for discovering transcription factor binding sites with improved specificity. *Pac. Symp. Biocomput.* 5, 467–478.
- Yu, Q., Huo, H.W., Chen, X.Y., et al. 2015. An efficient algorithm for discovering motifs in large DNA data sets. *IEEE Trans. Nanobiosci.* 14, 535–544.
- Zhang, S.Q., and Chen, Y. 2016. CLIMP: Clustering motifs via maximal cliques with parallel computing design. *PLoS One* 11, e0160435.
- Zhang, S.Q., Zhou, X.G., Du, C.B., et al. 2013. SPIC: A novel similarity metric for comparing transcription factor binding site motifs based on information contents. *BMC Syst. Biol.* 7 Suppl 2, S14.

Address correspondence to:

Prof. Shu-Lin Wang
College of Computer Science and Electronic Engineering
Hunan University
Changsha 410082
Hunan
China

E-mail: smartforesting@gmail.com