An Estimation of Distribution Algorithm for Motif Discovery

Gang Li, Tak-Ming Chan, Kwong-Sak Leung and Kin-Hong Lee

***Abstract*— The problem of Transcription Factor Binding Sites identification or motif discovery is to identify the motif binding sites in the cis-regulatory regions of DNA sequences. The biological experiments are expensive and the problem is NP-hard computationally. We have proposed Estimation of Distribution Algorithm for Motif Discovery (EDAMD). We use Bayesian analysis to derive the fitness function to measure the posterior probability of a set of motif instances, which can be used to handle a variable number of motif instances in the sequences. EDAMD adopts a Gaussian distribution to model the distribution of the sets of motif instances, which is capable of capturing the bivariate correlation among the positions of motif instances. When a new Position Frequency Matrix (PFM) is generated from the Gaussian distribution, a new set of motif instances is identified based on the PFM via the Greedy Refinement operation. At the end of a generation, the Gaussian distribution is updated with the sets of motif instances. Since Greedy Refinement assumes a single motif instance on a sequence, a Post Processing operation based on the fitness function is used to find more motif instances after the evolution. The experiments have verified that EDAMD is comparable to or better than GAME and GALF on the real problems tested in this paper.**

I. Introduction

TRANSCRIPTION Factor Binding Sites (TFBSs) are crucial components in gene regulation by interacting with transcription factors (TFs) and affecting the transcriptional activity. Specifically, TFBSs are small nucleotide fragments (usually $\leq$ 30 bp) in the cis-regulatory regions of genes in DNA sequences. Such cis-regulatory regions are usually hundreds to several thousand base pairs (bp) in length. TFBSs are bound by certain proteins called transcription factors (TFs), which together regulate the transcriptional activity (expression) of genes, and finally affect the phenotypes of the organism.

The biological experiments to identify TFBSs, such as DNA footprinting [1] and gel electrophoresis [2], are expensive and labor intensive. Computational methods, namely *de novo* TFBS identification/motif discovery, have been proposed as an attractive alternative. The basis is that certain conserved pattern, called "motif", exists among the TFBSs in the cis-regulatory regions for a set of similarly expressed genes (co-expressed genes), because those genes are probably regulated by the same TF. Benefitting from the availability of the large amount of sequencing and microarray data, we now can identify co-expressed genes by clustering and thus extract their cis-regulatory regions. *de novo* TFBS identification/motif discovery tries to find out the motif, or equivalently the set of conserved TFBSs of co-expressed genes without prior knowledge about their consensus appearance.

However, many of the TFBSs are only weakly conserved due to the mutation in evolution, i.e., an unknown degree of mismatching exists between those TFBSs and the consensus. The mismatching between TFBSs exhibits the greatest challenge for computational methods. Besides, we have no idea of the consensus a priori during the searching in *de novo* motif discovery. Li et al [3] have given the problem definition of general consensus patterns based on a simplified assumption that each sequence should have exactly one instance and proved it to be NP-hard.

Due to the low conservation and NP hardness, exact string matching methods fail and exhaustive search is also infeasible. Existing combinatorial approaches [4] set the constraint that the maximal hamming distance d between any pair of the TFBS instances is assumed to be known, and try to find out all the strings satisfying the constraints in polynomial time. Typical works include the data structure of suffix trees [5][6] and projections [7][8]. However, such approaches cannot meet the requirements of real world problems well because they can only handle small motif widths and small d within reasonable computational time. Moreover, d is difficult to determine beforehand and it varies case by case. With too small a d, most of the true TFBSs are missed due to the stringent criteria.

Some stochastic learning methods such as Expectation Maximization in MEME [9], Gibbs sampling [10][11] and Hidden Markov Models [12] have been proposed and shown some success in TFBS identification. The disadvantages of these methods include that they are sensitive to initial parameter settings, and are often trapped in local optima since many of these methods perform local search only. In TFBS identification, the local optima problems become more critical because the weakly conserved TFBSs are typical weak signals surrounded by a large amount of noises.

Recently, novel evolutionary algorithms (EAs) [13][14], mostly Genetic Algorithms (GAs) have been proposed to address TFBS identification [15][16][17][18][19][20][21]. EAs have been proven to be effective and robust in difficult search problems and are superior to locally incremental and single-point search methods because EAs perform global search while maintaining a population of different solutions concurrently. Other advantages of EA compared with the conventional motif discovery methods include the flexibility of representations and scoring functions in which advanced model can be easily fitted, and good scaling property which is promising for the large amount of data in DNA sequences.

The current GA methods are either consensus-based

The authors are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong (email: {gli, jfwang, ksleung, khlee}@cse.cuhk.edu.hk).

This work is supported by RGC Research Grant Direct Allocation of The Chinese University of Hong Kong.

978-1-4244-1823-7/08/$25.00©2008 IEEE

or matrix-based. In the consensus-based methods [15][18][22][19], an individual is encoded as a fragment of nucleotides, either extracted from the collected sequences or generated randomly. Hamming distance is widely used but cannot measure weak conservation precisely, where a dominant nucleotide does not exist. Some researchers try to use more accurate functions such as information content [23] to post process the consensus-based results but they run into the synonym problems which even deteriorate the results [22]. Matrix-based approaches [14][16][17][20][21] model the motif as a position-specific weight matrix (PWM) from a candidate set of TFBSs. However, due to the prohibitively large search space, most of these approaches have the assumption of one instance per sequence [14][16][17][21], or zero or one instance per sequence [20]. However, these simplified assumptions may not be true because some sequences may not contain any TFBSs but is just incorrectly collected based on the expression data. On the other hand, the widely used scoring functions by GAs, such as information content, are not capable of handling various number of instances per sequence. As a result, the application of GAs to TFBS identification is limited.

In this paper, we propose a novel Estimation of Distribution Algorithm for Motif Discovery (EDAMD) to handle more general assumptions for TFBS identification/motif discovery. EDAMD relaxes the simplified assumption of one instance per sequence to be any number of instances in the collected sequences. The objective of EDAMD is to search for the optimal Position Frequency Matrix (PFM) and the corresponding motif instances in sequences. EDAMD models the sampled motif instances as a weighted Gaussian Distribution, which is able to capture the possible dependencies between the probabilities in the corresponding PFMs. The motif instances are to be evaluated by a solid scoring function, which is based on the Bayesian analysis. Moreover, a local searching technique inspired by Gibbs sampling [10][11] and local filtering techniques [21] refines individuals efficiently. With the GA output, a Post Processing procedure improves the results to be even more accurate and complete. As a result, EDAMD not only identifies TFBS more accurately and efficiently, but it also handles more general cases for TFBS identification, which should be welcome by practitioners. Experimental results show that the results of EDAMD are comparable to or better than two other GA-based algorithms, namely GAME and GALF.

The rest of the paper is organized as follows. Section II uses Bayesian analysis to derive a scoring function of the motif instances. Section III describes the local operation to search for the motif instances from the sampled PFMs and the post procedure to find more motif instances based on the result of evolution. Section IV presents the overall algorithm of EDAMD and how a Gaussian distribution is employed in EDA. Section V gives the experimental results and compares the results to other algorithms. Section VI concludes the paper.

II. Scoring Function

Biologically, the TFBS identification problem is to find the subsequences in the cis-regulatory regions which are bound by a common protein. Up till now, biologists are still not very clear about the process of binding, let alone the properties of the binding sites. To cope with this problem computationally, we define the problem and its objective mathematically first. In particular, we design a fitness function for the problem to be used in our EDA approach.

A. Problem Formulation

Data Input. We are given a set of sequences $S = \{S_i|i = 1, 2, ..., D\}$ of nucleotides defined on the alphabet $B = \{A, T, G, C\}$. $S_i = (S_i^j|j = 1, 2, ..., l_i)$ is a sequence of nucleotides, where l_i is the length of the sequence. The width of the motif w can be predefined.

Position Output. We are required to find the Position Indicator Matrix (PIM) $A = \{A_i|i = 1, 2, ..., D\}$, where $A_i = \{A_i^j|j = 1, 2, ...l_i\}$ is the indicator vector with respect to a sequence S_i. A_i^j is 1 if A_i^j is the starting position of a binding site, and 0 otherwise. We refer to the number of motif instances as $|A| = \sum_{i=1}^{D}\sum_{j=1}^{l_i} A_i^j$.

Induced by A is a set of $|A|$ motif instances denoted as $S(A) = (S(A)_1, S(A)_2, ..., S(A)_{|A|})$. $S(A)$ can also be expanded as $(S(A)^1, S(A)^2, ..., S(A)^w)$, where $S(A)^j$ is the set of nucleotides in the jth position in the motif instances.

Consensus Output. We are also required to find the consensus which is a string abstraction of the motif instances. In the absence of a common consensus, we should find the Position Count Matrix (PCM) $N(A)$ of the numbers of different nucleotide bases on the individual positions of the motif instances. $N(A) = (N(A)^1, N(A)^2, ..., N(A)^w)$, and $N(A)^w = \{N(A)_b^w|b \in B\}\}$.

$N(A)$ can be further normalized by $|A|$, and thus we get the Position Frequency Matrix (PFM) $\hat{N}(A)$, which can be regarded as a virtual consensus. Given an A, it is trivial to calculate $N(A)$. On the contrary, given an $N(A)$, it is difficult to find the corresponding A.

Fig. 1 illustrates an artificial motif discovery problem. We use $M(C) = \{M(C)_b|b \in B\}$ to denote the numbers of different nucleotides in the set C, where $M(C)$ applies to all the positions in C. In particular, $M(S)$ can be normalized as the relative frequencies of the nucleotides which is denoted as $\theta_0 = \{\theta_{0b} = \frac{M(S)_b}{\Sigma_{b\in B} M(S)_b}|b \in B\}$.

B. Maximum A Posteriori

To solve a motif discovery problem, we need to find the optimal PIM A or PCM $N(A)$ in terms of a certain optimization measure. We adopt the Bayesian analysis by [24] to derive a statistical model of the motif instances. We repeat the major steps of the derivation herein.

Researchers often assume that the nucleotides in a motif instance are generated independently across positions. Therefore, we assume the motif instances $N(A)$ are generated from the multinomial distribution $\prod_{j=1}^{w} p(N(A)^j)$, where $p(N(A)^j)$ is the independent probability of generating the

sequences S	PIM A	instances $S(A)$	PCM $N(A)$	PFM $\hat{N}(A)$
acgtCGATTGCctaag	0000100000000000	CGATTGC		
taTGATCGAtgacgca	0010000000000000	TGATCGA	A:0261107	A: 0.0 0.2 0.6 0.1 0.1 0.0 0.7
cgaCAATTGAgcttac	0001000000000000	CAATTGA	C:8023323	C: 0.8 0.0 0.2 0.3 0.3 0.2 0.3
gCGCTCGAcaagctgt	0100000000000000	CGCTCGA	G:0800080	G: 0.0 0.8 0.0 0.0 0.0 0.8 0.0
cgttTGTCACAgtcta	0000100000000000	TGTCACA	T:2026600	T: 0.2 0.0 0.2 0.6 0.6 0.0 0.0
tcagcCACACCCagct	0000010000000000	CACACCC		
ccagagCGTCTGAttg	0000001000000000	CGTCTGA		
gacttcaCGACTGAgc	0000000100000000	CGACTGA	$M(S)_A$:38 $M(S)_C$:47	$\theta_{0A} = 0.2375$ $\theta_{0C} = 0.2938$
gctgcccatCGATTGA	0000000001000000	CGATTGA	$M(S)_G$:38 $M(S)_T$:37	$\theta_{0G} = 0.2375$ $\theta_{0T} = 0.2313$
ccaggtacCGATTGCa	0000000010000000	CGATTGC		

Fig. 1. An artificial problem of motif discovery. It shows the sequences S, the position indicator matrix A, the motif instances $S(A)$, the position count matrix $N(A)$, the position frequency matrix $\hat{N}(A)$, the count of the background nucleotides $\{M(S)_b | b \in B\}$ and the background relative frequencies $\{\theta_{0b} | b \in B\}$. In the sequences S, the letters in lower case are the background bases, and the letters in upper case are the motif instances

nucleotides on the jth position of the motif instances. We further assume that the probabilities of generating the nucleotides on a position of the different motif instances are independent. We say $N(A)^j$ follows a multinomial distribution $\prod_{b \in B} \theta_{jb}^{N(A)^j_b}$, where θ_{jb} is the latent probability of generating base b in the position j, $N(A)^j_b$ is the number of nucleotide b in the position j. In a more succinct form, $\prod_{b \in B} \theta_{jb}^{N(A)^j_b}$ can be written as $\theta_j^{N(A)^j}$, where θ_j is the vector of the latent probabilities $\{\theta_{jb} | b \in B\}$ on position j in the motif instances. Therefore, the motif instances $N(A)$ follow the multinomial distribution $\prod_{j=1}^{w} \theta_j^{N(A)^j}$. Similarly, we assume that the nucleotides on the sequences excluding the motif instances follow a multinomial distribution $\theta_0^{M(A^C)} = \prod_{b \in B} \theta_{0b}^{M(A^C)_b}$, where θ_0 is the probabilities generating the background nucleotides and A^C is the complement of A with respect to S. In this paper, we assume θ_0 is fixed as the relative frequencies of the bases in S, which is indifferent to the positions of the bases. We also assume an independent binomial distribution of $|A|$, i.e, $p(A|p_0) = p_0^{|A|} \times (1 - p_0)^{L-|A|}$, where $L = \sum_{i=1}^{N} l_i$ is the sum of the lengths of the sequences and p_0 is an abundance ratio to indicate the probability of the position being a motif site.

A can be viewed as the missing label of the data S, θ is the latent parameters of the distribution model of A, and p_0 is also unknown beforehand. The likelihood of S is the product of the probabilities of the background sequences and the motif instances as shown in Eq. 1.

$$p(S|\theta, \theta_0, A, p_0) = p_0^{|A|}(1 - p_0)^{L-|A|} \theta_0^{M(A^C)} \prod_{j=1}^{w} \theta_j^{N(A)^j} \tag{1}$$

For Bayesian analysis, we employ the multinomial Dirichlet distribution as the conjugate prior for θ, i.e., $p(\theta|\alpha) \propto \prod_{j=1}^{w} \prod_{b \in B} \theta_{jb}^{\alpha_b - 1}$. Please note α is a common prior for all the θ_j. Therefore, the posterior distribution of A and θ is shown in Eq. 2. Please note we have used $\theta_0^{M(A^C)} = \theta_0^{M(S)} / \theta_0^{M(A)} \propto 1/\theta_0^{M(A)}$.

$$\begin{aligned} p(\theta, A|S, \theta_0, p_0, \alpha) &\propto p(S|\theta, \theta_0, A) p(A|p_0) p(\theta|\alpha) \\ &= \frac{p_0^{|A|}(1 - p_0)^{L-|A|}}{\theta_0^{M(A)}} \prod_{j=1}^{w} \theta_j^{N(A)^j + \alpha - 1} \end{aligned} \tag{2}$$

Since, we are not interested in θ, we can integrate them out using the beta function $B(a) = \int \prod_i p_i^{a_i - 1} dp$ and convert the result using gamma function $B(a) = \prod_i \Gamma(a_i) / \Gamma(\Sigma_i a_i)$. The posterior conditional distribution of A alone is shown in Eq. 3.

$$\begin{aligned} p(A|S, \theta_0, p_0, \alpha) &\propto \int p(\theta, A|S, \theta_0, p_0, \alpha) d\theta \\ &= \frac{p_0^{|A|}(1 - p_0)^{L-|A|}}{\theta_0^{M(A)}} \prod_{j=1}^{w} \frac{\prod_{b \in B} \Gamma(N(A)^j_b + \alpha_b)}{\Gamma(|A| + |\alpha|)} \end{aligned} \tag{3}$$

where $|\alpha| = \sum_{b \in B} \alpha_b$ is the sum of prior probabilities of θ and $\sum_{b \in B} N(A)^j_b = |A|$. The objective of motif discovery can thus be formulated as maximize the posterior conditional probability of A in Eq. 3. An obvious advantage of this model is that it does not fix the number of motif instance in a sequence to be one. There can be zero, one, two or more motif instances in a sequence. In addition, prior knowledge, such as the abundance ratio of motif instances in the dataset, the background frequencies of the nucleotides and the prior probabilities of nucleotides in the motif instances, can be easily incorporated in the model. However, calculating the Γ function using numerical method is computationally expensive. We can further simplify Eq. 3 using the Burnside approximation $x! \approx (x + 0.5)^{x+0.5} e^{-x-0.5} \sqrt{2\pi}$. After some tedious derivation, the log of the approximated posterior conditional probability of A is shown in Eq. 4.

$$\begin{aligned} \psi(A) &= log(p(A|S, \theta_0, p_0, \alpha)) \\ &\approx K + |A|(log(p_0) - log(1 - p_0)) - M(A) log(\theta_0) \\ &\quad + \sum_{j=1}^{w} (N(A)^j + \alpha - 0.5) log(N(A)^j + \alpha - 0.5) \\ &\quad - w(|A| + |\alpha| - 0.5) log(|A| + |\alpha| - 0.5) \end{aligned} \tag{4}$$

where K is invariant w.r.t all the variables. For vectors v and v', we use $v log(v')$ as a shorthand for $\sum v_i log(v'_i)$. We will use Eq. 4 as the fitness function in our EDAGA. After further simplification, Eq. 4 contains a term of Information Content (IC), namely $\prod_{j=1}^{w} \theta_j \log \frac{\theta_j}{\theta_0}$, as used in other algorithms, such as GAME [20] and GALF [21]. IC models the discriminatory motif since it is an approximation of the difference between the log probability of motif with respect to the latent probabilities θ and the one with respect to the background probabilities θ_0. However, IC does not accommodate variable number of motif instances, and the simplification from Eq. 4 might be too coarse.

III. Searching Method

In ordinary GA, crossover and mutation are the primary searching operator. However, we can take advantage of the properties of the motif discovery problem, and devise more efficient searching operators than the generic GA operators. The first operator Greedy Refinement is a local search mutation. It finds a new set of binding sites based on the initial set of binding sites. The new set of motif instances has a higher posterior probability than the old one. The second operator Post Processing is applied on the best set of motif instances after the evolution. It retains most of the motif instances, and adds some new motif instances to increase the posterior probability.

A. Greedy Refinement

Given a motif PIM A, it is easy to calculate its fitness according to $\psi(A)$. In the other way around, we can maximize $\psi(A)$ to find the optimal A. However, solving for the optimum of $\psi(A)$ according to Eq. 4 analytically is intractable. Instead, we iteratively search for the optimal A_{ij} (0 or 1), while fixing the rest of A. Moreover, as indicated by the Eq. 4, searching for the optimal A is equivalent to searching for the optimal $N(A)$ as long as there actually exists a set of motif instances A whose PCM is $N(A)$.

For the time being, we hold the strong assumption that each sequence S_i contains a single motif instance, meaning only a single element in A_i is 1. In that case A collapses to a vector P where P_i is the index of the binding site on S_i.

Given a set of motif instances, which may not be the optima necessarily, we hope to find better instances. We take an iterative procedure to refine the motif instances one by one. Suppose we have located the binding sites on all the sequences except a single sequence S_i we are working on, we look for the fragment on S_i which matches the other instances best. A measure of similarity of a site A_i^j to the other binding sites is the probability of A_i^j being a binding site conditional on the other binding sites. The dissimilarity of A_i^j can be measured as the conditional probability of A_i^j not being a binding site. Now we can use the Bayes factor $\varphi(A_i^j)$ in Eq. 5 to determine whether A_i^j is a binding site. The derivation is very similar to the one used in Eq. 3. Please note we have used the equation $\Gamma(x+1) = x\Gamma(x)$. $N(P^*)_b^k$ is the number of nucleotide $b = S_i^{j+k-1}$ in $S(P^*)^k$, meaning the same nucleotide at position k in $S(P^*)$ as the one b in $S(A_i^j)$. $|P^*|$ is the number of motif instances already identified.

$$\begin{aligned}\varphi(A_i^j) &= \frac{p(A_i^j = 1|P^*, S)}{p(A_i^j = 0|P^*, S)} \\ &= \frac{\int p(A_i^j = 1|\theta, P^*, S)p(\theta|P^*, S)\, d\theta}{\int p(A_i^j = 0|\theta, P^*, S)p(\theta|P^*, S)\, d\theta} \\ &\propto \frac{1}{\theta_0^{M(A_i^j)}} \prod_{k=1}^{w} \frac{N(P^*)_b^k + \alpha_b}{|P^*| + |\alpha|} \qquad (5)\end{aligned}$$

After calculating the Bayes factors of all the A_i^j on S_i, we need to determine which one is the binding site. Gibbs sampling [25] selects a site randomly in proportion to the $\varphi(A_i^j)$ in Eq. 5 in S_i. It iteratively samples the binding sites in all the sequences one after another (possibly rewinds to S_1 after sampling on S_N), and updates P^* accordingly after each sampling. Gibbs sampling is a Markov Chain Monte Carlo method, and so it may takes a long time before generating samples following the target distribution.

We adopt a different method to select the binding sites. Instead of sampling the binding sites probabilistically, we select the site of the maximal $\varphi(A_i^j)$ directly, i.e., $P_i = \mathrm{argmax}_{j=1,\ldots,l_i} \varphi(A_i^j)$. Similar to Gibbs sampling, we select the binding sites on all the sequences iteratively. After selecting the binding site on S_i, we continue to select the binding site on S_{i+1}. We process all the sequences in a round, and then we return to S_1 and begin a new round. We stop when P remains the same in two consecutive rounds.

B. Post Processing

Post Processing addresses the issue of variable number of motif instances in a sequence. Greedy Refinement finds a binding site on each sequence S_i. However, a sequence may have zero, one, or more than one binding site(s). It is therefore important to allow the program to remove some spurious binding sites, and add more potential binding sites. The position indicator vector P can be easily converted to position indicator matrix A. The conversion is $A_i^j = \delta_{j,P_i}$, where δ_{j,P_i} returns 1 if the two arguments are equal, and 0 otherwise. Adding or removing a binding site depends on whether it contributes to the score of the whole of the binding sites. To check if a binding site A_i^j contributes to the score or not, we calculate the ratio $\xi(A_i^j)$ between the posterior probabilities (scores in Eq. 3) of the motif instances with it and without it as in Eq. 6.

$$\xi(A_i^j) = \frac{p(A'|S)}{p(A|S)} = \frac{p_0}{1-p_0} \frac{1}{\theta_0^{M(A_i^j)}} \prod_{j=1}^{w} \frac{N(A)_b^j + \alpha_b}{|A| + |\alpha|} \qquad (6)$$

where $A' = A \oplus A_i^j$, adding the motif instance A_i^j to A. If $\xi(A_i^j) > 1$, the binding site A_i^j contributes to the overall scoring fitness, otherwise it affects the fitness negatively. Please note Eq. 6 and Eq. 5 are the same except for the additional term $\frac{p_0}{1-p_0}$. This is expected, since Bayes factor also compares the posterior conditional probabilities with or without a certain binding site.

We adopt a two-phase procedure to Post Process the binding sites identified with Greedy Refinement. In the first phase, we calculate all the $\xi(A_i^{P_i})$, $i = 1, 2, ..., N$, and removes the binding sites of ξ values less than a threshold t_1. Although the assumption of every sequence contains at least a motif instance may not be true, Greedy Refinement tries to find a motif instance on each sequence. Therefore, the first phase is important to eliminate the spurious binding sites introduced by Greedy Refinement. In the second phase, for each sequence S_i, we calculate $\xi(A_i^j)$ for all the possible positions on S_i, and add the binding sites of ξ values bigger than another threshold t_2. This thus locates more binding sites on the sequences. The order of the two phases is not reversible. Due to the possible spurious binding sites, some noise may be embedded in the latent probabilities θ. Therefore, the noise must be removed before searching for more motif instances.

It is difficult to choose appropriate thresholds t_1 and t_2. If t_1 is fixed, a large t_1 may cause true motif instances to be removed, while a small t_1 may have no effect on the possible noise. We calculate t_1 automatically based on the current motif instances. A multinomial distribution parameterized by θ can be induced by $N(A)$ via the Maximum Likelihood approach, i.e., $\theta = \hat{N}(A)$. Suppose we generate a set of artificial motif instances according to the induced distribution, we can calculate their contribution to the current motif instances as in Eq. 6, and thus we get the expected contribution $E(\xi)$ of the motif instances under the distribution θ. For a motif instance identified by Greedy Refinement, if its ξ is less than $E(\xi)$, it is rejected. The threshold $t_1 = E(\xi)$ is calculated in Eq. 7, where a_i is one of 4^w possible motif instances (w is the length of the motif, and each position of the motif has four possible nucleotides: A,C,G and T), and b_i^j is the nucleotide on position j in the motif instance a_i. Please note we have used Eq. 8 to calculate the sum of all the $\xi(a_i)p(a_i), i = 1, 2, \cdots, 4^w$ so that Eq. 7 can be solved analytically.

$$
\begin{aligned}
t_1 = E(\xi(a)) &= \sum_{i=1}^{4^w} \xi(a_i) p(a_i|\theta) \\
&= \frac{p_0}{1-p_0} \frac{1}{(|A|+|\alpha|)^w} \sum_{i=1}^{4^w} \prod_{j=1}^{w} \frac{N(A^j)_{b_i^j} + \alpha_{b_i^j}}{\theta_0^{b_i^j}} \frac{N(A^j)_{b_i^j}}{|A|} \\
&= \frac{p_0}{1-p_0} \frac{1}{(|A|+|\alpha|)^w} \prod_{j=1}^{w} \sum_{b \in B} \frac{N(A^j)_b + \alpha_b}{\theta_0^b} \frac{N(A^j)_b}{|A|}
\end{aligned} \tag{7}
$$

$$
\underbrace{\sum_{j_1}^{4} \sum_{j_2}^{4} \cdots \sum_{j_w}^{4}}_{w} \prod_{j=1}^{w} x_{j_i}^j = \prod_{j=1}^{4} \sum_{i=1}^{4} x_{j_i}^j \tag{8}
$$

From the preliminary experiments, we find the value of t_2 should not be fixed beforehand either, since the appropriate t_2 varies case by case. We use a heuristic rule to adjust t_2 adaptively. After removing some motif instances in the first phase, we use the minimum of all the ξ of the remaining instances as the initial t_2 in the second phase. The second phase is then carried out in rounds iteratively. In a round, a set of candidate motif instances $\{a_i^t | \xi(a_i^t) > t_2\}$ are selected, and then we calculate the new t_2 as $min(\{\xi(a_i^t) | \xi(a_i^t) > 1\})$, which is used in the next round. We use a small initial t_2 at the beginning of the second phase to select a sufficient number of candidate motif instances, and afterwards we increase t_2 adaptively so as to select the motif instances of positive contributions only.

IV. Estimation of Distribution Algorithm

Estimation of Distribution Algorithm (EDA) [26] is a variant of Genetic Algorithm (GA). GA maintains a population of individuals, and generates new individuals from existing individuals in the population. On the contrary, EDA learns a distribution model of the individuals, and samples new individuals from the model instead. As GA encodes the patterns of solutions as schemata which are embedded in the population, EDA incorporates the solution features in the distribution model explicitly, and it is thus able to solve problems efficiently and effectively when the prior knowledge of the problem is available.

Estimation of Distribution Algorithm for Motif Discovery (EDAMD) is basically an iterative algorithm. It samples new Position Frequency Matrices PFMs from a Gaussian distribution, searches for the corresponding motif instances based on the PFMs, and updates the distribution model with the motif instances discovered. As pointed out in Section II, the ultimate goal of EDAMA is finding the binding site of the motif, and the solution can also be represented as the motif consensus, i.e., $N(A)$. There may be interdependencies among the positions in the motif instances, which means the nucleotides on one position affect the nucleotides on another position [27][28][29]. In other words, the relative frequencies of the nucleotides on different positions have some correlation with each other. We cannot incorporate the complicated inter-relation of positions in the scoring function 4, however, we can use a Gaussian distribution, i.e., $\mathcal{G}(x|\mu, \Sigma) \propto e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$, to model the motif instances and capture the possible pairwise correlations across the positions. Since the argument x in the Gaussian model $\mathcal{G}(x|\mu, \Sigma)$ is a column vector, we concatenate all the columns of PFM together, and refer to the resulted vector as Position Frequency Vector (PFV). It actually represents the same nucleotide frequencies of a set of motif instances as PFM.

After sampling a PFM Q, Greedy Refinement is used to find a set of the corresponding motif instances. However, Greedy Refinement starts with a set of motif instances $S(A)$, while Q may not correspond to a real set of motif instances, so it cannot be used in Greedy Refinement directly. To provide $S(A)$ for Greedy Refinement, we can make up a set of artificial motif instances $\tilde{A}$, each of which is exactly the same as Q. Consequently, $D \times Q$ is equal to the $N(\tilde{A})$ of the artificial motif instances, and Q is equal to $\hat{N}(\tilde{A})$. $D \times Q$ is then provided as the initial $N(A)$ for Greedy Refinement to find the matching motif instances. Please note the artificial motif instance $\tilde{A}$ may not exist in S, and they usually do not contain the valid nucleotides since the numbers in $N(\tilde{A})$ are fractional.

Afterwards, we use the sets of motif instances to update the Gaussian distribution. Traditionally, the mean and the covariance are updated as $\mu = \frac{1}{T}\sum_{i=1}^{T} x_i$ and $\Sigma = \frac{1}{T}\sum_{i=1}^{T}(x_i - \mu)(x_i - \mu)'$ respectively, where T is the number of data samples. However, not all the motif instances are genuine motif instances, and they have different similarities to the common consensus. In addition, the fitness of the sets of the motif instances are not the same. Therefore, the motif instances should not be treated equally. Instead, we use the weighted updating formula in Eq. 9, where $\{z_i|i = 1, 2, ..., T\}$ are the weights to measure the importance of the motif instances. The term $\sigma \times I(4w)$ in Σ is an identity matrix multiplied with a small positive constant. In the evolution, the motif instances may converge to the common motif consensus. To enhance the diversity of the sampled PFMs, we need to keep the diagonal elements of Σ larger than 0.

$$\begin{aligned} \mu &= \frac{\sum_{i=1}^{T} z_i \times x_i}{\sum_{i=1}^{T} z_i} \\ \Sigma &= \frac{\sum_{i=1}^{T} z_i \times x_i x_i'}{\sum_{i=1}^{T} z_i} + \sigma \times I(4w) \end{aligned} \tag{9}$$

The motif instances are usually weakly conserved. They may be different from the common consensus, and they are different from each other. Therefore, even for the motif instances in the same set A found from a common PFM Q, the instances may have different conditional probability with respect to A. Intuitively, the weights associated with the instances should be $z_i = \psi(A) \times p(a_i|A), i = 1, 2, ..., |A|$, where a_i is a motif instance. A motif instance in a good set of motif instances surely has a large weight, but if it is dissimilar from the consensus, its weight should be scaled down. The conditional probability of motif instance a_i w.r.t A is calculated in Eq. 10, where S_{a_i} is the sequence containing a_i, $S \oplus S_{a_i}$ is the sequences S plus the sequence S_{a_i}, $A \oplus a_i$ is the set of motif instances A plus a_i, and $F = N(A^j)_b + N(a_i^j)_b + \alpha_b - 0.5$. The term $\Gamma(|A|+|\alpha|+1)$ is ignored since in the evolution, the number of motif instances $|A|$ is fixed to the number of sequences D.

$$\begin{aligned} \rho(a_i) &= \int p(a_i|A, S, \theta)p(A, S|\theta)p(\theta)d\theta \\ &\propto \frac{\prod_{j=1}^{w}\prod_{b \in B} F^F}{\theta_0^{M(A \oplus a_i)}} \end{aligned} \tag{10}$$

Alternatively, we can also use $\varphi(a_i)$ in Eq. 5 to calculate the weight. After all, given a set of potential binding sites, the order of their Bayesian factors and that of their posterior conditional probabilities are the same. If $\rho(a_1) > \rho(a_2)$, then $\varphi(a_1) > \varphi(a_2)$.

Algorithm 1: The Main Program of EDAMD

```
Input: The Sequences S
Output: The Best Motif Instance BA
$FIT \leftarrow 0$;
randomly initialize $\mathcal{G}$;
for $g$ from $0$ to $G$ do
    for $i$ from $0$ to $T$ do
        if $i < \frac{T}{2}$ then
            $\hat{N}(\tilde{A})_{(i)} \sim \mathcal{G}(x|\mu, \Sigma)$;
        else
            $\hat{N}(\tilde{A})_{(i)} \sim \mathcal{U}(x)$;
        end
        $N(\tilde{A})_{(i)} \leftarrow D \times \hat{N}(\tilde{A})_{(i)}$;
        $[A_{(i)}, S(A)_{(i)}, \psi(A)_{(i)}, \rho(A)_{(i)}] \leftarrow Greedy(N(\tilde{A})_{(i)}, S)$;
        if $\psi(A)_{(i)} > FIT$ then
            $FIT \leftarrow \psi(A)_{(i)}$;
            $BA \leftarrow A_{(i)}$;
        end
    end
    $[\mu, \Sigma] \leftarrow Update(\{S(A)_{(i)}\}, \{\psi(A)_{(i)}\}, \{\rho(A)_{(i)}\})$;
end
$BA \leftarrow Post(S, BA)$;
algocf
```

Algorithm 1 is the overall program of EDAMD. For a generation, half of the PFMs are sampled from a Gaussian distribution $\mathcal{G}$. To enhance the diversity of PFMs, the other half are sampled from a uniform distribution $\mathcal{U}$. After finding a set of motif instances $A_{(i)}$ based on a sampled PFM $\hat{N}(\tilde{A})_{(i)}$ via the Greedy Refinement function $Greedy$, the best set of motif instances BA is updated if $\psi(A_{(i)})$ is better than the best fitness FIT. At the end of the generation, the Gaussian distribution model is updated with the sets of motif instances $\{S(A)_{(i)}\}$, their fitness $\{\psi(A)_{(i)}\}$ and the conditional probabilities of the instances $\{\rho(A)_{(i)}\}$. Finally, Post Processing is applied on BA to find more motif instances.

V. Experiments

We have tested EDAMD on eight real DNA datasets. A dataset consists of sequences with motif instances already tagged. Therefore we can use these datasets as benchmark problems. Besides, we assume the widths of the motifs are known beforehand. We say a motif instance is correctly recovered if the predicted binding site is within 3 bp away from the true binding site. The 3 bp tolerance is reasonable since in a real dataset, the widths of the tagged motif instances vary around the known width, and they are sometimes larger than the indicated width. We think the true motif instance should lie somewhere between the two ends of the tagged instances. This criterion of successful prediction is also used in GAME [20] and GALF [21]. To measure the performance of EDAMD and other algorithms, we adopt the standard metrics of $Precision$, $Recall$ and $F{-}score$ as defined in Eq. 11, where the operator $|\cdot|$ is the cardinality of the set. After we find the candidate instances computationally, the results need to be verified in biological experiments. We hope for a high $Precision$ to avoid wasting too much effort on the false motif instances. In addition, we should miss as few true motif instances as possible, so a high $Recall$ is preferred. $F - score$ mixes $Precision$ and $Recall$ since there is a tradeoff between $Precision$ and $Recall$. Sometimes a high $Recall$ means a large number of candidate instances, which may consist of many false positives. On the contrary, some

TABLE I

THE SETTING OF THE BENCHMARK DATASETS: THE NUMBER OF SEQUENCES, THE LENGTH OF SEQUENCES, THE WIDTH OF MOTIFS AND THE NUMBER OF MOTIF INSTANCES

dataset	#sequence	length	width	#instance
CREB	17	350	8	19
CRP	18	105	22	23
ERE	25	200	13	25
E2F	25	200	11	27
MEF2	17	199	7	17
MYOD	17	200	6	21
SRF	20	345	10	36
TBP	95	200	6	95

true weakly conserved motif instances are deleted by mistake in order to achieve a high $Precision$.

$$\begin{aligned} Precision &= \frac{|correct\ motif|}{|motif\ found|} \\ Recall &= \frac{|correct\ motif|}{|true\ motif|} \\ F-score &= 2 \times \frac{Precision * Recall}{Precision + Recall} \end{aligned} \qquad (11)$$

The eight real datasets are CREB, CRP, ERE, E2F, MEF2, MYOD, SRF and TBP [30][20]. The cyclic Amp receptor protein (CRP) binds in *Escherichia coli*. The estrogen receptor binds in the sequences of estrogen response elements (ERE). The E2F family also contains known binding sites. The datasets of CREB, MEF2, MYOD, SRF and TBP are published in ABS database of annotated regulatory binding sites. The benchmark datasets have a variety of the numbers of sequences, the lengths of sequences, the widths of motifs and the numbers of motif instances as shown in Table I. We have run EDAMD for each dataset 20 times with different random seeds. The population size T is 100, and the maximal generation G is 10. Moreover, the motif widths we use are the same as used in GAME and GALF. The best and the average results in the 20 runs are recorded.

We have compared the performance of EDAMD to those of GAME and GALF, which also tested the eight problems. The results are reported in Tables II and III. Table II shows the best results of the three algorithms in 20 runs. As regard to the F-score, EDAMD is the best in 6 problems. In the remaining two problems, it is worse than GAME on E2F, and worse than GALF on TBP. Table III shows the average results of the three algorithms in 20 runs. As regard to the F-score, EDAMD is the best on 7 problems, and it is worse than GAME on E2F. A remarkable observation is that the average results are the same as the best results in EDAMD. Actually, EDAMD always get the same results no matter what the random seed is. An explanation is that Greedy Refinement always finds the same best motif instances even from the set of different initial PFMs. In addition, both GALF and GAME employ a population of 500 individuals. GALF runs up to 300 generations, and GAME runs up to 3000 generations. In addition, GALF and GAME use multi-start GA. In each run of GAME and GALF, GA was run 20 times. Consequently, the total numbers of fitness evaluations are 3,000,000, and 30,000,000 in GALF and GAME, respectively. On the contrary, the number of fitness evaluations of EDAMD is only 1000, which is significantly small compared to GALF and GAME. On the other hand, the Greedy Refinement on an individual is computation intensive, However, it is difficult to compare the running time fairly, because GALF was implemented in C, GAME was implemented in Java, and EDAMD was implemented in Matlab.

VI. CONCLUSION

We have proposed Estimation of Distribution Algorithm for Motif Discovery (EDAMD). It uses the fitness function derived by Bayesian analysis to measure the posterior conditional probability of a set of motif instances. Therefore, it is able to handle variable number of motif instances in the DNA sequences. It adopts a Gaussian distribution to model the distribution of the sets of motif instances. The Gaussian distribution is capable of capturing the bivariate correlation among the positions of motif instances. When a new Position Frequency Matrix (PFM) is sampled from the Gaussian distribution, a new set of motif instances is identified from the PFM via the Greedy Refinement operation. At the end of a generation, the Gaussian distribution is updated with the sets of motif instances considering the fitness and the probabilities of the instances. Since Greedy Refinement finds a single motif instance on a sequence, a Post Processing operation is used to find more motif instances after the evolution. The experiments have verified that EDAMD outperforms GAME and GALF on most of the real problems tested in the paper, and its results are much more stable.

REFERENCES

[1] D. J. Galas and A. Schmitz, “DNAse footprinting: a simple method for the detection of protein-DNA binding specificity,” *Nucleic Acids Res.*, vol. 5, no. 9, pp. 3157–3170, September 1987.

[2] M. M. Garner and A. Revzin, “A gel electrophoresis method for quantifying the binding of proteins to specific DNA regions: application to components of the escherichia coli lactose operon regulatory system,” *Nucleic Acids Res.*, vol. 9, no. 13, pp. 3047–3060, July 1981.

[3] M. Li, B. Ma, and L. Wang, “Finding similar regions in many sequences,” *Journal of Computer and System Sciences*, vol. 65, pp. 73–96, 2002.

[4] P. Pevzner and S. Sze, “Combinatorial approaches to finding subtle signals in dna sequences,” in *Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000, pp. 269–278.

[5] M. F. Sagot, “Spelling approximate repeated or common motifs using a suffix tree,” in *LATIN ’98: Theoretical Informatics, Lecture Notes in Computer Science*. Springer-Verlag, 1998, pp. 111–127.

[6] P. Bieganski, J. Riedl, J. V. Carlis, and E. Retzel, “Generalized suffix trees for biological sequence data: applications and implementations,” in *Proc. of the 27th Hawaii Int. Conf. on Systems Sci.*, 1994, pp. 35–44.

[7] J. Buhler and M. Tompa, “Finding motifs using random projections,” in *RECOMB*, 2001, pp. 69–76.

[8] B. Raphael, L. Lung-Tien, and G. Varghese, “A uniform projection method for motif discovery in dna sequences,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 2, pp. 91–94, 2004.

TABLE II

COMPARISONS OF EDAMD, GALF AND GAME ON THE EIGHT DATASETS: BEST RESULTS (PRECISIONS, RECALLS AND F-SCORES)

Dataset	GAME			EDAMD			GALF		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
CREB	**0.78**	0.74	0.76	0.73	**0.84**	**0.78**	0.76	0.68	0.72
CRP	0.86	**0.78**	0.82	**0.94**	0.74	**0.83**	**0.94**	0.74	**0.83**
ERE	0.53	**0.80**	0.63	**0.76**	0.76	**0.76**	**0.76**	0.76	**0.76**
E2F	**0.80**	**0.89**	**0.84**	0.71	0.80	0.75	**0.80**	0.74	0.77
MEF2	0.89	**1.00**	0.94	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
MYOD	0.48	0.48	0.48	0.86	**0.90**	**0.88**	**0.88**	0.71	0.79
SRF	0.73	**0.92**	0.81	0.77	**0.92**	**0.84**	**0.95**	0.53	0.68
TBP	0.80	0.85	0.83	0.85	**0.94**	0.89	**0.93**	0.93	**0.93**

TABLE III

COMPARISONS OF EDAMD, GALF AND GAME ON THE EIGHT DATASETS: AVERAGE RESULTS (PRECISIONS, RECALLS AND F-SCORES)

Dataset	GAME			EDAMD			GALF		
	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
CREB	0.43	0.42	0.42	0.73	**0.84**	**0.78**	**0.76**	0.68	0.72
CRP	0.79	**0.78**	0.78	**0.94**	0.74	**0.83**	0.93	0.73	0.82
ERE	0.52	**0.78**	0.62	**0.76**	0.76	**0.76**	**0.76**	0.76	**0.76**
E2F	**0.79**	**0.87**	**0.83**	0.71	0.80	0.75	0.76	0.70	0.73
MEF2	0.52	0.55	0.53	**1.00**	**1.00**	**1.00**	0.97	0.97	0.97
MYOD	0.14	0.14	0.14	0.86	**0.90**	**0.88**	**0.88**	0.71	0.79
SRF	0.71	0.86	0.78	0.77	**0.92**	**0.84**	**0.88**	0.49	0.63
TBP	0.81	0.74	0.77	0.85	**0.94**	**0.89**	**0.88**	0.88	0.88

[9] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," in *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 1994, pp. 28–36.

[10] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, "Bayesian models for multiple local sequence alignment and Gibbs sampling strategies," *J. Am. Stat. Assoc.*, vol. 90, no. 432, pp. 1156–1170, November 1995.

[11] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wooton, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," *Science*, vol. 262, no. 8, pp. 208–214, October 1993.

[12] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. D. Moor, P. Rouze, and Y. Moreau, "A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling," *Bioinformatics*, vol. 17, pp. 1113–1122, 2001.

[13] M. A. Lones and A. M. Tyrrell, "The evolutionary computation approach to motif discovery in biological sequences," in *GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation*, 2005, pp. 1–11.

[14] G. B. Fogel, D. G. Weekes, G. Varga, E. R. Dow, H. B. Harlow, J. E. Onyia, and C. Su, "Discovery of sequence motifs related to coexpression of genes using evolutionary computation," *Nucleic Acids Res.*, vol. 32, no. 13, pp. 3826–3835, 2004.

[15] F. F. M. Liu, J. J. P. Tsai, R. M. Chen, S. N. Chen, and S. H. Shih, "FMGA: Finding motifs by genetic algorithm," in *BIBE '04: Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*, 2004, pp. 459–466.

[16] D. Che, Y. Song, and K. Rasheed, "MDGA: motif discovery using a genetic algorithm," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, 2005, pp. 447–452.

[17] J. Gertz, L. Riles, P. Turnbaugh, S. W. Ho, and B. A. Cohen, "Discovery, validation, and genetic dissection of transcription factor binding sites by comparative and functional genomics," *Genome Research*, vol. 15, pp. 1145–1152, 2005.

[18] T. K. Paul and H. Iba, "Identification of weak motifs in multiple biological sequences using genetic algorithm," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 271–278.

[19] M. Stine, D. Dasgupta, and S. Mukatira, "Motif discovery in upstream sequences of coordinately expressed genes," in *CEC '03: Evolutionary Computation, The 2003 Congress on*, vol. 3, 2003, pp. 1596–1603.

[20] Z. Wei and S. T. Jensen, "GAME: detecting cis-regulatory elements using a genetic algorithm," *Bioinformatics*, vol. 22, no. 13, pp. 1577–1584, 2006.

[21] T.-M. Chan, K.-S. Leung, and K.-H. Lee, "TFBS identification by position- and consensus-led genetic algorithm with local filtering," in *GECCO '07: Proceedings of the 2007 conference on Genetic and evolutionary computation*, 2007, pp. 377–384.

[22] C. B. Congdon, J. Aman, G. M. Nava, H. R. Gaskins, and C. Mattingly, "An evaluation of information content as a metric for the inference of putative conserved noncoding regions in dna sequences using a genetic algorithms approach," *IEEE TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, vol. 5, June 2007.

[23] G. D. Stormo, "Computer methods for analyzing sequence recognition of nucleic acids," *Annu. Rev. BioChem.*, vol. 17, pp. 241–263, 1988.

[24] S. T. Jensen, X. S. Liu, Q. Zhou, and J. S. Liu, "Computational discovery of gene regulatory binding motifs: A bayesian perspective," Feb. 2004. [Online]. Available: http://ProjectEuclid.org/getRecord?id=euclid.ss/1089808282

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

[26] P. Larraanaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[27] R. A. O'Flanagan, G. Paillard, R. Lavery, and A. M. Sengupta, "Non-additivity in protein-DNA binding," *Bioinformatics*, vol. 21, no. 10, pp. 2254–2263, 2005. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bti361

[28] Q. Zhou and J. S. Liu, "Modeling within-motif dependence for transcription factor binding site predictions," *Bioinformatics*, vol. 20, no. 6, pp. 909–916, 2004. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bth006

[29] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan, "Modeling dependencies in protein-DNA binding sites," in *RECOMB*, 2003, pp. 28–37. [Online]. Available: http://doi.acm.org/10.1145/640075.640076

[30] E. Blanco, D. Farré, M. M. Albà, X. Messeguer, and R. Guigó, "ABS: a database of annotated regulatory binding sites from orthologous promoters," *Nucleic Acids Research*, vol. 34, no. Database-Issue, pp. 63–67, 2006. [Online]. Available: http://dx.doi.org/10.1093/nar/gkj116