Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Bioinformatics and Computational Biology (CIBCB 2007)

A Genetic-Based EM Motif-Finding Algorithm for Biological Sequence Analysis

Chengpeng Bi
Children's Mercy Hospitals and Clinics
Schools of Medicine, Computing and
Engineering, University of Missouri
2401 Gillham Road
Kansas City, MO 64108, USA

***Abstract*– Motif-finding in biological sequence analysis remains a challenge in computational biology. Many algorithms and software packages have been developed to address the problem. The Expectation Maximization (EM)-type motif algorithm such as MEME is one of the most popular *de novo* motif discovery methods. However, as pointed out in literature, EM algorithms largely depend on their initialization and can be easily trapped in local optima. This paper proposes and implements a Genetic-based EM Motif-Finding Algorithm (GEMFA) aiming to overcome the drawbacks inherent in EM motif discovery algorithms. It first initializes a population of multiple local alignments each of which is encoded on a chromosome that represents a potential solution. GEMFA then performs heuristic search in the whole alignment space using minimum distance length (MDL) as the fitness function which is generalized from maximum log-likelihood. The genetic algorithm gradually moves this population towards the best alignment from which the motif model is derived. Simulated and real biological sequence analysis showed that GEMFA performed better than the simple multiple-restart of EM motif-finding algorithm especially in the subtle motif sequence alignment and other similar algorithms as well.**

I. INTRODUCTION

The Expectation Maximization (EM) algorithm [1] is a standard algorithm for Maximum Likelihood (ML) and maximum a posterior (MAP) point estimation. There are two main applications of the EM algorithm: (i) when the data indeed have missing values due to problems with or limitations of the observation process; (ii) when optimizing the likelihood function is analytically intractable, but when the likelihood function can be simplified by assuming the existence of and values for additional but missing (or hidden/latent) parameters. The expectation maximization (EM) algorithm is one of the earliest and most powerful motif discovery algorithms that was formulated as a maximum likelihood function and treated motif sites as missing data. It was used to predict de novo motif sites and estimate parameters in motif maximum likelihood models. The EM-based motif discovery algorithm was first developed using Position Weight Matrix (PWM)-based statistical modeling in [2]. This methodology has been generalized to one of the most popular motif-finding software called MEME [3]. The EM algorithm is widely used due to its simplicity and stability [4]. However, as pointed out in literature, the EM algorithm largely depends on its initialization and can be easily trapped in local optima. This paper proposes and implements a Genetic-based EM Motif-Finding Algorithm (GEMFA) aiming to overcome the drawbacks inherent in EM motif discovery algorithms. It first initializes a population of multiple local alignments each of which is encoded on a chromosome that represents a potential solution. GEMFA then performs heuristic search in the whole alignment space using minimum distance length (MDL) as the fitness function which is generalized from maximum log-likelihood.

The rest of the paper is organized as follows: Section II briefly introduces the EM algorithm and its combination with GA; Section III proposes a framework for motif discovery problems; Section IV briefly summarizes EM motif discovery algorithms; Section V describes a new algorithm, Genetic-based EM Motif-Finding Algorithm (GEMFA) and then implements it; Section VI gives experimental results of GEMFA and compares with other EM algorithms using both simulated and real biological DNA sequences with annotated protein binding motifs; Section VII concludes the paper with discussion.

II. EM AND GA ALGORITHMS

A. EM algorithms

The maximum likelihood method is widely used to estimate an unobserved parameter vector that maximizes the log-likelihood function which is defined as,

$L(\mathbf{u} \mid \boldsymbol{x}) = \sum_i \log g(\boldsymbol{x}_i \mid \mathbf{u})$

where the observations $\boldsymbol{x} = \{\boldsymbol{x}_i \mid i = 1, \ldots, n\}$ are supposed to be independently drawn from the distribution $g(\boldsymbol{x})$ parameterized by $\mathbf{u}$. The EM algorithm [1] is an iterative procedure designed to find maximum likelihood (ML) estimates in the context of parametric models where the observed data can be viewed as incomplete. The objective is to estimate $\mathbf{u}$ by maximizing $L(\mathbf{u}) = \log g(\boldsymbol{x}|\mathbf{u})$. However, here $\boldsymbol{x}$ is *incomplete data* and therefore $\mathbf{u}$ cannot be directly estimated from L. The basic idea of EM is to take advantage of the full-data expectation of the ML estimate: $\boldsymbol{z}$

= (***x***, ***y***), here ***y*** denotes unobserved data. The joint density function is thus defined as,

$$f(\mathbf{z} \mid \mathbf{u}) = f(\boldsymbol{x}, \boldsymbol{y} \mid \mathbf{u}) = q(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{u}) g(\boldsymbol{x} \mid \mathbf{u})$$

Where $q(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{u})$ is the marginal distribution of the unobserved data and is dependent on both the observed data ***x*** and on the current parameters, and let **Y** be the space of values that ***y*** can assume. The new complete-data likelihood is: $L(\mathbf{u} \mid z) = L(\mathbf{u} \mid \boldsymbol{x}, \boldsymbol{y}) = \log f(\boldsymbol{x}, \boldsymbol{y} \mid \mathbf{u})$. The EM algorithm first finds the expected value (or *Q*-function) of the complete-data log-likelihood $f(\boldsymbol{x}, \boldsymbol{y} \mid \mathbf{u})$ with respect to the missing data ***y*** given the observed data ***x*** and the current parameter estimates ($\mathbf{u}^{(t)}$), that is,

$$Q(\mathbf{u}; \mathbf{u}^{(t)}) = \int_{\mathbf{Y}} \log f(\boldsymbol{x}, \boldsymbol{y} \mid \mathbf{u}) q(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{u}^{(t)}) d\boldsymbol{y}$$

where **u** is the next parameter vector to be optimized in the likelihood. The M-step determines $\mathbf{u}^{(t+1)} = \arg\max_{\mathbf{u}} \{Q(\mathbf{u}; \mathbf{u}^{(t)})\}$. These two steps are iterated until the algorithm converges. Each iteration is guaranteed to increase the log-likelihood, and the algorithm is guaranteed to converge to a local maximum of the likelihood function.

However, despite some appealing features, the EM algorithm has several well-documented limitations: its final position can strongly depend on its starting position; its rate of convergence can be painfully slow; and it can provide a saddle point of the likelihood function rather than a local maximum [5]. Recently, quite a few adaptations and extensions to the EM approach have been proposed in order to address the problem of convergence to a local optimum and the initialization issue such as multiple-restart of EM with random seeds [6], Markov Chain Monte Carlo strategies such as Monte Carlo EM [5], Gibbs sampling [7], and most recently Metropolis [8].

B. Combining GA with EM algorithms

Another strategy to overcome the inherent drawbacks mentioned above aims to combine EM with the genetic algorithms (GA's). GA's are adaptive search techniques designed to find near-optimal solutions of large scale optimization problems with multiple local minima. In fact GA's are extensively utilized in solving diverse biological problems such as sequence alignment [9] and biological data clustering [10,11]. References [12,13] reported the first genetic algorithm-based EM (GA-EM) algorithms for learning mixture models. However, GA algorithms are specific for problems studied and thus each unique problem usually needs a specific GA algorithm design. The EM-type motif-finding problem is very different from that of learning normal mixture models: (i) both observed (i.e. nucleic acid sequences) and unobserved (motif location) data in motif-finding are discrete rather than continuous, (ii) the motif and background models follows the product of multinomial distributions and (iii) multiple bio-sequence alignment has proven to be the NP-complete problem. Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad. This paper describes a new genetic algorithm, i.e. genetic-based EM motif-finding algorithm (GEMFA), which combines conventional EM motif-finding model with GA-based adaptive or heuristic search strategy. It is hypothesized that the new heuristic method shall be better than other simple strategies such as multiple-restart. The minimum distance length (MDL) criterion is used for selecting individual chromosomes. The new combination algorithm establishes a framework in a way such that both GA and EM are synergistically utilized. Such a union can explore the multiple alignment space more thoroughly than EM used alone and is especially suited for subtle motif-finding or training large data sets. GEMFA encodes each multiple alignment solution on a chromosome and performs standard genetic operations (i.e. crossover, mutation and selection) to evolve an optimal or near-optimal solution.

III. A Framework For Motif-Finding

Let $\mathbf{S} = \{S_1, \ldots, S_i, \ldots, S_N\}$ denote the sequence dataset of size *N*. Let L_i be the length of the sequence *i* (S_i) and S_{ij} can take on a value of the alphabet set *K*, for instance, *K* = {A, C, G, T} denoting the DNA sequence alphabet (i.e., 4 nucleotide types) at position *j* of the i-th sequence. Let $|K|$ be the number of letters in the bio-sequence alphabet (i.e., $|K| = 4$ for DNA, and $|K| = 20$ for protein sequences). If we assume only one motif per sequence (i.e. *oops* model), there are *N* motifs in total for *N* sequences. It may also be assumed there is zero or one motif per sequence (i.e. *zoops* model). Nonetheless, both *oops* and *zoops* models assume that sequence data come from a two-component multinomial mixture model: (i) the background model, assuming that each residue position of the motif is an independent and identical multinomial distribution ($\mathbf{u}_0$); and (ii) the *w*-mer motif model, assuming that each residue position within the motif is independent but not identical, thus, each position comes from a different multinomial distribution ($\mathbf{u}_j$). A motif sequence can be thought of drawing from a product of multinomial distributions: $\mathbf{U} = [\mathbf{u}_1, \ldots \mathbf{u}_j, \ldots \mathbf{u}_w]$.

Let A_i be the indicator variable drawing from the motif location space $\{0, 1\}^{Li - w + 1}$, $A = [A_1, \ldots, A_i, \ldots A_N]^T$ be the set of indicator variables representing the motif start sites in the sequence dataset, and *w* be the motif width to be user-defined. Total oops alignment space (V) is $O(L^{|A|})$, here *L* is the average sequence length and |A| is the total motif sites. The number of motif sites on sequence *i* can be defined as: $|A_i| = \sum_l A_{il}$. Therefore, if $|A_i| = 1$ for all *i*, then it is an *oops* model, otherwise it is a *zoops* or multiple-site model. The total motif sites is $|A| = \sum_i |A_i|$. The start sites are initially assumed to be uniformly distributed, i.e., $p(A_{il} = 1) = 1 / (L_i - w + 1)$ for all *l*. Alternatively, the position variable $a_i = l$ is used to represent the motif starting at

position l on sequence i, which is equivalent to $A_{il} = 1$. Note that $a_i = 0$ means no motifs found on sequence i. If multiple sites occur on a sequence, we use a vector (a_i) to store all the positions. Obviously the motif position vector a_i is a subset of {1, 2, …, $L_i - w + 1$}. The alignment of motif sites can be initialized by randomly generating a set of motif start sites (i.e. $\mathbf{A}^{(0)}$ or equivalently $[\mathbf{a}_1^{(0)},\ldots, \mathbf{a}_N^{(0)}]^T$) and then progressively or heuristically refined until a convergence criterion is satisfied.

Multiple sequence local alignment [14] is the most frequently used method to solve motif discovery problems. Each alignment can be thought of as a hidden individual state in the alignment space or total alignment population. The motif discovery problem can therefore be formulated as to finding the optimized alignment state (v*) among the entire alignment space (**V**). Index a state by $v \equiv [\mathbf{a}_1,\ldots, \mathbf{a}_i,\ldots, \mathbf{a}_N]^T = \mathbf{A}^{(v)}$, and let the energy of state v be $E(v) = E(\mathbf{S}, \mathbf{A}^{(v)})$ where $\mathbf{A}^{(v)}$ is the alignment corresponding to the state v. The energy is related to an alignment score or the motif sequence specificity/binding energy [15]. Then at equilibrium the population of state v is proportional to $\exp[-E(\mathbf{S}, \mathbf{A}^{(v)}) / k_B T]$. Here k_B is Boltzmann's constant and T is absolute temperature. The partition function (Z) of the total alignment population in the equilibrium distribution is defined as,

$$Z = \sum_{\boldsymbol{a}_1} \cdots \sum_{\boldsymbol{a}_i} \cdots \sum_{\boldsymbol{a}_N} e^{-E[\mathbf{S},\mathbf{A}^{(v)}]/k_B T}$$

where a_i indexes the motif start positions for sequence i. However, it is hard to compute the partition function because the total alignment space is usually prohibitively large. The equilibrium probability $p^{(v)}$ of alignment state v is defined as,

$$p^{(v)} = e^{-E[\mathbf{S},\mathbf{A}^{(v)}]/k_B T} \Big/ Z \tag{1}$$

Therefore, the optimized alignment state (v*) is the one with the maximum probability,

$$v^* = \underset{v \in \mathbf{V}}{\operatorname{argmax}}\, p^{(v)} \tag{2}$$

If v* is found, then the estimation of optimal motif model (**U***) is solved. However computing the partition function is usually very hard, because the multiple sequence alignment problems are NP-complete [16].

IV. EM Motif-Finding Algorithm

Since the random variable **A** is unobserved, maximum likelihood estimation can be done by maximizing the expectation of the full data log-likelihood given the observed data (**S**) with the EM algorithm. The full data for motif sequence model is (**S**, **A**) = {(S_i, A_i): i = {1, …, N}}. The conditional likelihood of sequence i, given the hidden variables (a_i), is as follows,

$$p(\boldsymbol{S}_i \mid a_i = l, \mathbf{U}, \boldsymbol{u}_0) = \prod_{y \in \boldsymbol{A}_{il}^c} \prod_{k=1}^{K} u_{0k}^{\mathrm{I}(S_{iy}=k)} \prod_{m \in \{1:w\}} \prod_{k=1}^{K} u_{mk}^{\mathrm{I}(S_{i,l+m-1}=k)} \tag{3}$$

where $\boldsymbol{A}_{il}^c$ denotes the background sites and I(•) is the indicator function. To simplify notation, **U** is used to contain the background parameters $\boldsymbol{u}_0$ in the following derivation. Let $\mathbf{U}^{(t)}$ be the parameter estimates after t-th iteration, then the conditional expected complete data log-likelihood given the observed data is often referred to as the Q-function which is defined as,

$$Q(\mathbf{U}; \mathbf{U}^{(t)}) = E[log\ p(\mathbf{S}, \mathbf{A}|\ \mathbf{U})|\mathbf{S}, \mathbf{U}^{(t)}] = \sum_{i=1}^{N} \sum_{l=1}^{L_i - w + 1} p(a_i = l \mid S_i, \mathbf{U}^{(t)})\ log\ p(\boldsymbol{S}_i \mid a_i = l, \mathbf{U}) \tag{4}$$

A. EM-type motif-finding algorithms

The EM-type motif-finding algorithms maximize the Q-function defined in (4) by iteratively performing the E- and M-steps [2]. The E-step calculates a conditional probability of each potential site:

$$p(a_i = l \mid S_i, \mathbf{U}^{(t)}) = \frac{p(\boldsymbol{S}_i \mid a_i = l, \mathbf{U}^{(t)})}{\sum_{j=1}^{L_i - w + 1} p(\boldsymbol{S}_i \mid a_i = j, \mathbf{U}^{(t)})} \tag{5}$$

The M-step maximizes the Q-function by re-estimating a new parameter matrix ($\mathbf{U}^{(t+1)}$) :

$$u_{0k}^{(t+1)} = \frac{N_{0k}}{\sum_{k=1}^{K} N_{0k}}$$
$$u_{jk}^{(t+1)} = \frac{N_{jk}}{\sum_{k=1}^{K} N_{jk}}, \qquad j \in \{1, \cdots, w\}, k \in K \tag{6}$$

where

$$N_{jk} = \begin{cases} \sum_{i=1}^{N} \sum_{l=1}^{L_i - w + 1} p(a_i = l \mid S_i, \mathbf{U}^{(t)}) \mathrm{I}(S_{i,(l+j-1)} = k), & j \neq 0, \\ \sum_{i=1}^{N} \sum_{l=1}^{L_i - w + 1} p(a_i = l \mid S_i, \mathbf{U}^{(t)}) \sum_{y \in A_{il}^c} \mathrm{I}(S_{iy} = k), & j = 0. \end{cases}$$

The EM starts from an initial model parameters $\mathbf{U}^{(0)}$ provided by the user or generated at random. The above EM algorithm simply iterates E- and M-step a number of times until a specified threshold is reached. To facilitate the combination of EM with GA and comparison with its genetic-based counterpart, it is necessary to re-implement an in-house EM motif-finding algorithm which has been done in C++ (detailed information appears in [8]). The newly implemented EM motif-finding algorithm is named as DEM and its performance is comparable to other popular EM motif software such as MEME [8].

B. w-mer motif model selection

The above described algorithm needs a pre-specified motif width (w). The motif width is usually unknown, however one can specify a range, the minimum description

length (*MDL*) criterion is defined and used to select the optimal *w*-mer motif model,

$$\begin{aligned} MDL &= -\log p(\mathbf{S} \mid \mathbf{U}) + (Kw+1)\log N \\ &= -\sum_{k=1}^{K} N_{0k} \log u_{0k} - \sum_{j=1}^{w}\sum_{k=1}^{K} N_{jk} \log u_{jk} + (Kw+1)\log N \end{aligned} \quad (7)$$

MDL is the most commonly used selection criterion [13]. Equation (7) has the intuitive interpretation that the log-likelihood is the code length of the encoded data. The term (Kw + 1)log$N$ models the optimal code length for all the estimated parameters (**U**). *MDL* is used as the fitness function for evolving a population of alignment solutions. The best individual alignment is the one with the lowest *MDL* value. However, if the motif length is known or fixed, *MDL* is thus reduced to maximize the log-likelihood function.

C. Scanning sequences for multiple sites

After a motif model is built as above, it can be used to scan a genomic sequence for putative sites of the same binding protein. Given a testing or genomic sequence (s), the estimated motif model (**U**), motif width (w) residue alphabet (K), and a motif site location (a), the motif score (*MS*) is given by,

$$MS(s, \mathbf{U}, a) = \sum_{m=1}^{w}\sum_{k=1}^{K} \log\left[\mathrm{I}(s_{i,a+m-1} = k)\left(\frac{u_{mk}^{(t)}}{u_{0k}^{(t)}}\right)\right]. \quad (8)$$

It is easy to see that equation (8) is derived from (3) and (5). The MS-score is positively proportionate to the level of motif conservation in the sequence. Equation (8) can be used to scan a sequence for more than one site. If a site has a score above a specified threshold, then the new found sequence can be thought of as a putative site. If it is experimentally verified, it can be added to the current validated motif site set. The minimum score among existing motif sites can be used as the threshold. On the other hand, if an existing motif site has very low MS-score, then it may imply that the sequence has no motif site. A more robust statistical investigation into the multiple-site scanning issue is under development.

V. Genetic-Based Motif-Finding

The genetic algorithm (GA) is a global optimization procedure that performs adaptive search to find optimal solutions of large scale optimization problems with multiple local minima. GA has a greater freedom of movement between different configurations (solutions) than simple Monte Carlo algorithms. They are well suited for solving some NP-complete problems such as multiple sequence alignments and motif discovery. SAGA is one of the earliest genetic algorithms used to perform multiple sequence alignment [9]. However, this study uses multiple local alignment to discover short motif sequences [2,3,6,7,8]. There are several studies applying GA into motif discovery problems either formulated as consensus sequence models [17,18] or PWM-based motif models [19,20]. Reference [19] used the relative entropy or information content as the fitness function and encoded the motif starting positions (i.e. solution) as a binary number. Most recently reference [20] presented another genetic algorithm (GAME) using Bayesain-derived scoring function as the fitness indicator. GAME encoded solutions on a string of integer numbers, and it is based on the simple genetic algorithm with two additional auxiliary genetic operators (i.e. ADUST and SHIFT). The PWM-based GA algorithms utilize a large number of randomly generated starting points and search for the whole local alignment space independent of any *de novo* motif discovery algorithms. However, these motif-finder-independent GA algorithms need a very large population to evolve a near-optimal solution which is not efficient. In contrast, this paper illustrates the power of a motif-finding algorithm (i.e. EM) based on GA. Although a good seed solution *per se* may have a lower score, it may grow to a much better final solution under the direction of a deterministic (EM) or stochastic (Gibbs sampling) search method.

A. GEMFA algorithm

This paper describes a new genetic algorithm GEMFA which is originated from the same idea as previously reported [19,20], i.e. treating each hidden random variable (a_i) as a locus encoded on a chromosome. However, GEMFA is dependent on the deterministic EM motif-finding algorithm. It treats each chromosome as a set of initial seeds or alignment (**A**) rather than an immediate solution (**U**) as previously reported [19,20]. Because of the deterministic properties in EM, it will converge to the same parameter estimation (**U**) given the same initial points (**A**). Thus one can write it as: **U** = *em*(**A**). Finding an optimal estimation (**U***) is equivalent to locating the best initial seeds (**A***) among the total alignment space (**V**) as described in section II. From this one can have two strategies in coding a chromosome: (1) encoding the parameters (real numbers) or (2) encoding the positions (integers). This paper implements the second strategy. It shall be very interesting to try the first strategy in the future.

A chromosome encodes an alignment solution consisting of a string of loci. Each locus is an object containing the motif position and its strand attribute (either forward coded as 0 or backward coded as 1). Assuming an oops motif model, a locus represents a starting position on one bio-sequence. However, this limitation can be easily relaxed by assigning a vector of such object. A population of chromosomes holds a set of potential multiple alignments. The maximum population size is 100 chromosomes (GAME used 500). The GA algorithm stops if no further

improvement or the maximum number of generations (g_{max}) specified is reached, or whichever comes first.

The genetic operations follow the standard or simple GA [10,20,21], i.e. a high crossover probability (P_c), a lower mutation rate (P_m) and a moderate population size (ps). Two-point crossover is performed in GEMFA with a crossover rate of 0.75 rather than one-point crossover used in GAME. Two-point crossover is commonly thought of as better than one-point [10,21]. The mutation rate is set as 0.01. A position mutation is generated by simply producing a new integer number within the potential range. A DNA strand attribute (0 or 1) mutates using the flip-flop operation. However, mRNA or protein sequence alignment is fixed on the forward strand. A binary tournament selection [22] is used to produce the offspring. A mixed pool, containing both parent and children populations, is first formed (2N) and then tournament selection is performed on the new 2N mixed population with a tournament selection probability of 0.75. However, only N individuals are selected to form the new generation [19].

EM motif-finding algorithm is applied to each new individual chromosome (i.e. starting seeds), and then the new parameters (**U**) are estimated. The fitness function is then calculated as in (7) or simply the log-likelihood function if motif width is fixed.

B. Implementation

It is straightforward to implement the GEMFA algorithm. Given a set of biological sequences (**S**) and a motif width (w), one can initialize a population of alignments ($\{\mathbf{A}_i^{(0)}: i = 1\ldots ps\}$ by randomly generating a set of motif start positions which are the initial population of chromosomes. Fitness is computed as: first applying EM to a chromosome and then calculating the *MDL*. The GEMFA algorithm proceeds by a series of genetic operations: crossover, mutation and selection until a specified number of generations or convergence. The GEMFA algorithm's pseudo-code is given below.

GEMFA algorithm:

1. Initializing: g_{max}, P_c, P_m, ps, $\{\mathbf{A}_i^{(0)}\}$, $t \leftarrow 0$
2. Calculate a population of initial motif matrix: $\{\mathbf{U}_i^{(0)}\} = em\{\mathbf{A}_i^{(0)}\}$ and $MDL\{\mathbf{U}_i^{(0)}\}$
3. **repeat:** $t \leftarrow t + 1$
4. $crossover\{\mathbf{A}_i^{(t)}\}$
5. $mutation\{\mathbf{A}_i^{(t)}\}$
6. $\{\mathbf{U}_i^{(t)}\} \leftarrow em\{\mathbf{A}_i^{(t)}\}$
7. $MDL\{\mathbf{U}_i^{(t)}\}$
8. $\{\mathbf{A}_i^{(t+1)}\} \leftarrow tournament\{\mathbf{A}_i^{(t)}, \mathbf{A}_i^{(t-1)}\}$
9. **until** ($t > g_{max}$ or convergence)
10. **output**: optimal alignment ($\boldsymbol{A}^*$) and its associated motif model ($\mathbf{U}^*$)

The notation $mutation\{\mathbf{A}_i^{(t)}\}$ denotes the mutation operation imposed on the t-th population. While initializing the motif sites in double helix DNA sequences, a coin is flipped to decide whether the motif is on the forward or backward strand.

C. Evaluation of algorithm performance

Two quantities to evaluate the motif-finding algorithm performance as described in [8] are: (1) the nucleotide-level performance coefficient (NPC); and (2) the motif site-level performance coefficient (SPC). Let O_i be the set of known motif positions in a sequence i and, A_i be the sets of motif positions located by an algorithm. Given the number of sequences (N), the performance metrics (NPC and SPC) are computed as,

$$NPC(N) = \sum_{i=1}^{N} |A_i \cap O_i| / |A_i \cup O_i| \Big/ N \qquad (9)$$
$$SPC(N) = \sum_{i=1}^{n} delta(A_i \cap O_i \neq empty) \Big/ N$$

The indicator function $delta(\bullet)$ simply enumerates the number of predicted sites with at least one-nucleotide overlap with the corresponding known sites. The NPC measures the nucleotide-level precision of the algorithm prediction that is equivalent to the performance coefficient defined in [23], while the SPC metric measures the site-level precision. The NPC gives a more exact performance measurement. However, the predicted sites usually have phase shift among different algorithms, so the SPC may be a fair metric for algorithm comparison.

VI. Experimental Results

This section describes motif-finding experiments using the GEMFA algorithm implemented in Perl and compares it with other related methods. To demonstrate the robust performance of GEMFA algorithm, simulated data sets are used to compare it with its counterpart: multiple-restart of the EM motif algorithm. Real biological sequences (i.e. CRP, ERE and E2F) are also used to compare the GEMFA genetic algorithm with other similar algorithms, i.e. GAME [20], MEME [6] and the Gibbs motif sampling algorithm BioProspector [22]. MEME and BioProspector are two of the most popular motif discovery algorithms. Two genetic-based algorithms (i.e. GAME and GEMFA) and the Gibbs sampling algorithm (i.e. BioProspector) use the known motif width in each case as their searching motif length (w). MEME is allowed to perform its powerful functionality of automatically determining the optimal motif width and therefore its motif width (frequently different from known motif width) is chosen such that it can achieve its best performance.

A. Algorithm comparison using simulated DNA sequences

Simulated data sets are generated to test the performance of the GEMFA algorithm in searching for motifs on DNA sequences. The planted motif width is set as 15 base pairs long. Each simulated dataset contains 20 sequences, each of which is 200 nucleotides in length. The simulated

datasets are made of the following combinations: (i) three background distributions: (a) uniform (A, C, G and T are equally likely occurring), (b) AT-rich (AT content = 60% and GC = 40%) and (c) GC-rich (GC = 60% and AT = 40%); (ii) motif conservation levels: (a) high, (b) mid and (c) low. A high conservation motif is formed such that at any position a dominant nucleotide has a probability of 0.91 and each of the rest is 0.03. A mid conservation motif is formed such that at any position a dominant nucleotide has a probability of 0.79 and each of the rest is 0.07. A low conservation motif is formed such that at any position a dominant nucleotide has a probability of 0.70 and each of the rest is 0.10. Each case (or training data set) is repeated twenty times. The 15-mer consensus sequence used in simulation is **GTCACGCCGATATTG** (the ratio of AT content to GC is 7/8). A fair Bernoulli coin is tossed in order for each motif sequence to have an equal likelihood of being planted either on the forward or backward strand. The location where a motif is implanted is randomly generated ranging from 1 to 186.

Table I shows the performance comparison results for the EM and GEMFA motif-finding algorithms. In high and mid conservation cases GEMFA performs equal to its deterministic counterpart EM. GEMFA perform better in low conservation cases of all backgrounds especially in the uniform background. There is about the same GC content as the AT in the planted motifs which is similar to the uniform background, EM shows poor performance while GEMFA successfully detected the motif sites (SPC = 0.65). GEMFA can detect all the motif models in high, mid or low conserved motif sequences in all backgrounds (SPC > 0.56). EM only detected the high and mid conserved motifs, but failed to identify the low conservation motifs in uniform background (SPC = 0.48). GEMFA illustrated its robust performance of motif finding in a subtle environment. The lowest performance in uniform background implies that finding low conserved motif sites similar to its background is extremely difficult. The genetic algorithm performs well in this circumstance.

TABLE I
ALGORITHM COMPARISON USING SIMULATED SEQUENCES[a]

Cons	Algorithm	Uniform	AT-rich	GC-rich
High	GEMFA	0.98 / 1.00	0.98 / 1.00	1.00 / 1.00
	EM	0.99 / 1.00	0.99 / 1.00	1.00 / 1.00
Mid	GEMFA	0.87 / 0.88	0.87 / 0.90	0.85 / 0.89
	EM	0.83 / 0.87	0.89 / 0.91	0.87 / 0.89
Low	GEMFA	**0.56 / 0.65**	0.50 / 0.60	0.52 / 0.56
	EM	0.38 / 0.48	0.47 / 0.58	0.48 / 0.54

[a]Each cell contains the average performance in the format of NPC / SPC. EM motif algorithm was run 500 times and the optimal model was kept. GEMFA was run 5 generations each with 100 individuals. NPC or SPC was calculated according to (9). Each simulation case was repeated 20 times. Bolded number shows the GA's significant improvement over EM ($P < 0.01$).

These simulation experiments illustrate that the EM motif algorithm encounters some difficulty in finding lower conserved motif because it easily converges to a local maximum (Table I). However, in high and mid conservation motif cases, EM performed as good as its genetic algorithm-based counterpart, i.e. the GEMFA algorithm. The GEMFA algorithm is more reliable in performance than EM alone, since it uses the power of EM and explores more alignment space in a systematic way driven by the heuristic technique and thus converges to a better solution than by simply restarting EM many times.

B. Algorithm comparison using biological sequences

Three annotated motif sequence data sets: CRP binding sequences from bacterial genomes and ERE and E2F binding site sequences from eukaryotic genomes, are used to demonstrate the performance of the GEMFA algorithm in comparison with two popular motif finding algorithms: MEME and Gibbs Sampler, as well as the recently reported genetic-based motif-finding algorithm (GAME).

The *CRP* data set is the gold standard testing data set [2]. Each determined binding site motif length is 22 base pairs. However, the best length set by MEME is 24 base pairs long. The remaining algorithms set width as 22 bp long. The binding motif sequences of E2F and ERE are short and extracted from [25] and [26] respectively. There are 25 ERE sequences, and its known motif length is 13 base pairs long. The E2F [25] binding data set contains 25 genomic sequences (known motif width is 11 bp), each 200 base pairs long with 27 embedded motif sites. The motif widths set by MEME are 15, and 13 base pairs for ERE and E2F, respectively. The remaining three algorithms set their widths the same as the known ones.

Table II summarizes the motif algorithm performance using the site-level precision (i.e. SPC). The GEMFA algorithm is the best predictor compared to the other three algorithms (0.85-0.92). GAME performed a little better than GEMFA in the case of E2F because there are two sequences each with 2 binding sites. However, GEMFA is based on the oops model and thus automatically ignores these extra sites. This functionality will be added to the GEMFA algorithm later on. MEME has better performance (0.67-0.76) than BioProspector's (0.46-0.68) on average, but is not as good as GEMFA and GAME in all cases. Figure 1 shows the sequence logos for three transcription factor binding motif models built from the GEMFA algorithm. The CRP motif has detected two core sub-motifs (5-mer) separated by six unspecified nucleotides: TGTGAnnnnnnTCACT, which is consistent with biological findings. Two short eukaryotic TF binding motifs (ERE and E2F) are slightly higher conserved due to the small number of verified sites, and their detected consensus sequences are comparable to their annotated motifs [25,26]. Figure 1 shows the sequence logos for three motif models.

TABLE II
ALGORITHM COMPARISON USING BIOLOGICAL DATA[b]

Motif	CRP	ERE	E2F
GEMFA	**0.88**	**0.92**	0.85
GAME	0.80	0.75	**0.90**
MEME	0.67	0.71	0.76
BioProspector	0.56	0.68	0.46

[b]Each cell contains the site-level performance coefficient (SPC). The SPC (site precision) data for GAME, MEME and BioProspector are copied from [20]. The number in bold corresponds to the best algorithm.

VII. DISCUSSION

One of the most important steps toward understanding gene regulation is the identification of the regulatory elements present in the genome. Motif-finding in bio-sequences remains a challenge in computational biology [27]. Many programs and software have been developed to address the problem [28]. The EM-type motif algorithms such as MEME are one of the most popular de novo motif discovery methods. However the EM algorithm largely depends on its initialization and can be easily trapped in local optima. This paper presents and validates a genetic algorithm to perform multiple local alignment of motif sequence binding sites according to maximum likelihood, or minimizing MDL in general principle.

The genetic-based EM motif-finding algorithm (GEMFA) is essentially an intelligent hybrid search method that aims to overcome the drawbacks inherent in EM motif algorithms. A population of multiple local alignments, each of which is encoded on a chromosome, represents a set of potential solutions. GEMFA gradually evolves this population generation by generation through standard genetic operations such as crossover, mutation and selection. It performs a heuristic search in the whole alignment space using minimum distance length (MDL) as the objective function which is related to maximum likelihood. The genetic algorithm gradually moves this population towards the best alignment. Results showed that GEMFA performed better than the simple multiple trials of EM motif-finding in subtle motif sequence alignment of simulated sequences. In addition, GEMFA performed equal to or better than other similar or popular motif-finding algorithms (i.e. GAME, MEME and BioProspector) while applying to real biological motif sequences.

Multiple local alignment (MLA) is the most frequently used method to solve motif discovery problems [14]. It is recognized as a prominent NP-complete problem in computational biology [16]. The models can be used to scan any genomic sequence for putative binding sites and their associated genes. Conventional machine learning approaches to MLA or motif discovery are susceptible to local minima, resulting in incorrect sequence alignments. Aside from compromising classification accuracy, sub-optimal solutions may impact correct interpretation of biological significance. By comparison, more structured search approaches such as GA's or other computational intelligence methods may obviate this potential problem through efficient global searches that bypass the requirement for exhaustive enumeration.

Assuming the oops motif model, i.e. each sequence contains exactly one motif site, and one encodes a single locus only for one motif coordinate per sequence. This idea may be extended to multiple same motif sites per sequence or distinct motif sites per sequence. Each locus can be designed as an object-oriented data structure capturing all the motif information on a sequence. A group of member functions are needed to operate on or access to the data structure and information. Therefore a chromosome is encoded as a list of such objects. Given this fundamental data structure and associated function, genetic operations can be performed as usual.

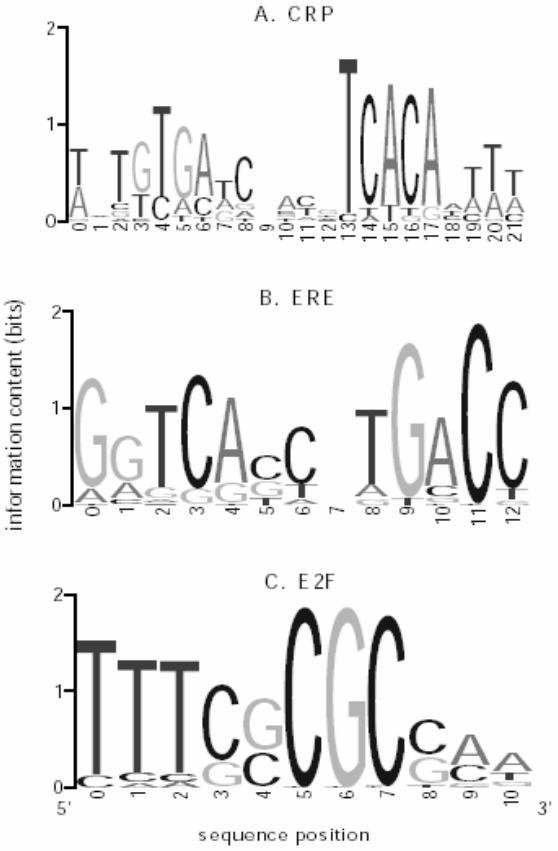

Fig. 1. Sequence logos [29] of the predicted protein binding sites plotted using WebLogo [30]. (A.) CRP (22-bp motif built from 18 sequences each 108 bp long), (B.) ERE (13-bp motif from 25 sequences each 200 bp long), and (C.) E2F (11-bp motif from 25 sequences each 200 bp long)

A gene is frequently not controlled by a single transcription factor or its cognate cis-regulatory element but a set of distinct cooperative binding cis-elements called a *cis*-regulatory module [31]. Therefore, a more challenging, yet biologically important problem is to *de novo* locate a set of distinct motifs simultaneously via multiple local alignments, which is an extension of the problem discussed above. It is evident that genetic algorithms and other evolutionary computational methods shall play essential roles in the challenging cis-module *de novo* discovery.

While a motif-finding algorithm is applied to a large number of sequences each longer than 500 bp, such as CHIP-chip location genomic data [28], it becomes very slow or eventually impossible if only a limited computing facility is available. Parallel computational strategies can be utilized to alleviate the computing burden of a large-scale alignment problem. Genetic algorithms are able to take advantage of PC cluster computing power. One can send each alignment or individual chromosome to one node, and thus a population of individual alignment jobs can be simultaneously processed. This simple strategy will greatly improve the motif-finding efficiency and thus facilitate the large-scale multiple sequence alignment.

ACKNOWLEDGEMENTS

I express my gratitude to Dr. Michael C. Saunders for his supervision during my graduate study at the Pennsylvania State University. I thank Dr. Heather L. Newkirk for proofreading. The research is supported in part by the Katharine B. Richardson Foundation.

REFERENCES

[1] A.P. Dempster, N. M. Laird and D. B. Rubin, " Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *Journal of the Royal Statistical Society B*, vol 39, pp. 1-38, 1997.

[2] C.E. Lawrence, and A.A.Reilly, "An expectation maximization algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins: Structure, Function and Genetics*, vol 7, pp. 41-51, 1990.

[3] T.L.Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, vol 1, pp. 28-36, 1994.

[4] C.F.J. Wu, "On the convergence properties of the EM algorithm," *The Annals of Statistics*, vol 11, pp. 95-103, 1983.

[5] G. Celeux, D. Chauveau and J. Diebolt, "Stochastic versions of the EM algorithm: an experimental study in the mixture case," *J. Statist. Comput. Simul.,* vol 55, pp. 287-314, 1996.

[6] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, vol 21, pp. 51-80, 1995.

[7] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald and J.C. Wootton, "Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment," *Science*, vol 262**,** pp. 208-214, 1993.

[8] C.-P. Bi, "SEAM: A stochastic EM-type algorithm for motif-finding in biopolymer sequences," *J. Bioinformatics & Comput. Biol.*, in press.

[9] C. Notredame and D.G. Higgins, "SAGA: sequence alignment by genetic algorithm," *Nucl. Acids Res.,* vol 24, pp. 1515-1524, 1996.

[10] C.-P. Bi, *Pattern Classification of the Rhagoletis pomonella Species Complex.* University Park, PA: Pennsylvania State University Graduate School, 2002.

[11] A.P. Gasch and M.B. Eisen, "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering," *BMC Bioinformatics*, vol 3, pp. 1, September 2002.

[12] A.M. Martinez and J. Vitria, "Learning mixture models using a genetic version of the EM algorithm," Pattern Recognition Letters, vol 21, pp. 759-769, 2000.

[13] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol 27, pp. 1344-1348, 2005.

[14] W.S. Waterman, *Introduction to Computational Biology: Maps, Sequences and Genomes*. Florida: Chapman & Hall/CRC Press, 1995.

[15] O.G. Berg and P.H. von Hippel, "Selection of DNA binding sites by regulatory proteins: sta-tistical-mechanical theory and application to operators and promoters," *Journal of Molecular Biology*, vol 193, pp. 723-750, 1987.

[16] P. Bonizzoni and G.D. Vedova, "The complexity of multiple sequence alignment with SP-score that is a metric," *Theoretical Computer Science*, vol 259, pp. 63–79, 2001.

[17] M. Stine, D. Dasgupta and S. Mukatira, "Motif discovery in upstream sequences of coordinately expressed genes," *Evol. Comput., CEC'03*, vol 3, pp. 1596-1603, 2003.

[18] F.F.M. Liu, J.J.P. Tsai, R.M. Chen, S.N. Chen and S.H. Shih, "FMGA: finding motifs by genetic algorithm," BIBE, vol 04, pp. 459, 2004.

[19] D. Che, Y. Song and K. Rasheed, "MDGA: motif discovery using a genetic algorithm," *GECCO*, vol 05, pp. 447-452, 2005.

[20] Z. Wei and S.T. Jensen, "GAME: detecting cis-regulatory elements using genetic algorithm," *Bioinformatics*, vol 22, pp. 1577-1584, April 2006.

[21] M. Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1998.

[22] D.E. Goldberg et al., "Do not worry, be messy," In: *Proceeding of the Fourth International Conference on Genetic Algorithms*. California: Morgan Kaufmann Publishers, pp. 24-30, 1991.

[23] P. A. Pevzner and S-H. Sze, "Combinatorial approaches to finding subtle signals in DNA sequences," *Proc. Int. Conf Intell. Syst. Mol. Biol.*, vol 8, pp. 269-278, 2000.

[24] X. Liu, D.L. Brutlag and J.S. Liu, "BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," *Pacific Symposium on Biocomputing*, vol 6, pp. 127-138, 2001.

[25] A.E. Kel et al., "Computer-assisted identification of cell cycle-related genes: new targets for E2F transcription factors," *J. Mol. Biol.*, vol 309, pp. 99-120, 2001.

[26] C.M. Klinge et al., "Estrogen receptor interaction with estrogen response elements," *Nucleic Acids Res.*, vol 29, pp. 2905-2919, 2001.

[27] M. Tompa et al., "Assessing computational tools for the discovery of transcription factor bind-ing sites," *Nature Biotechnology*, vol 23, pp. 137-144, 2005.

[28] K.D. MacIsaac and E. Fraenkel, "Practical strategies for discovering regulatory DNA sequence motifs," *PLoS Comput Biol*, vol 2, pp. e36, 2006.

[29] T.D. Schneider and R.M. Stephens, "Sequence logos: a new way to display consensus sequences," *Nucleic Acids Res.*, vol 18, pp. 6097-6100, 1990.

[30] G.E. Crooks, G. Hon, J.M. Chandonia, and S.E. Brenner, " WebLogo: A sequence logo generator," *Genome Research*, vol 14, pp. 1188-1190, 2004.

[31] Q. Zhou and W.H. Wong, "CisModule: *De novo* discovery of cis-regulatory modules by hierarchical mixture modeling," *PNAS USA*, vol 101, pp. 12114-12119, 2004.