Identification of Weak Motifs in Multiple Biological Sequences using Genetic Algorithm

Topon Kumar Paul and Hitoshi Iba
Department of Frontier Informatics
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi
Chiba 277-8561, Japan
topon@iba.k.u-tokyo.ac.jp, iba@iba.k.u-tokyo.ac.jp

ABSTRACT

Recognition of motifs in multiple unaligned sequences provides an insight into protein structure and function. The task of discovering these motifs is very challenging because most of these motifs exist in different sequences in different mutated forms of the original consensus motif and thus have weakly conserved regions. Different score metrics and algorithms have been proposed for motif recognition. In this paper, we propose a new genetic algorithm based method for identification of multiple motifs instances in multiple biological sequences. The experimental results on simulated and real data show that our algorithm can identify multiple occurrences of a weak motif in single sequences as well as in multiple sequences. Moreover, it can identify weakly conserved regions more accurately than other genetic algorithm based motif discovery methods.

Categories and Subject Descriptors: I.5.2 [PATTERN RECOGNITION]: Design Methodology—Pattern analysis; I.2.8 [ARTIFICIAL INTELLIGENCE] : Problem Solving, Control Methods, and Search—Heuristic methods; I.2.6 [ARTIFICIAL INTELLIGENCE]: Learning—Knowledge acquisition, parameter learning; J.3 [LIFE AND MEDICAL SCIENCES] : Biology and genetics

General Terms: Algorithms, performance

Keywords: Motif discovery, (l, d) motif, protein binding site, regulatory sites, DNA sequences, genetic algorithm, clustering

1. INTRODUCTION

In molecular biology, a motif is a weakly conserved nucleotide or amino-acid sequence pattern that is widespread and has, or is conjectured to have, a biological significance [22]. These motifs in the promoter region of a gene, where a transcription factor molecule binds, are called promoter sequences or transcription factor binding sites (TFBSs) motifs. Recognition of these promoter sequences is important in understanding the regulations of gene expression of a gene. The similarity between the promoter sequence and the consensus motif determines the strength of the binding of the transcription factor to promoters. In general, the more closely the promoter sequence resembles the consensus sequence, the stronger the promoter [8]. Mutations in the consensus motifs create different motifs instances that may present in different coregulated genes, and these motif instances have weakly conserved regions. Therefore, the methods of identifying the motifs in multiple DNA sequences that rely on common substrings or words will likely fail [23].

Many computational approaches have been proposed for recognition of motifs in DNA or amino acid sequences. Broadly, these algorithms can be divided into two groups: deterministic and nondeterministic. Most deterministic algorithms use regular expression based rules to specify some classes of allowable patterns for motifs, and these algorithms are in nature exhaustive. Pratt [12] and TEIRESIAS [19] are examples of two deterministic algorithms that use regular expression rules to identify motifs. MDScan [16] is another enumerative deterministic algorithm that, instead of regular expressions, uses higher order Markov model and maximum a posterior (MAP) score to evaluate candidate motifs. On the other hand, most nondeterministic motif discovery algorithms are non-exhaustive and stochastic in nature, and in different runs, they may find different motifs that may or not be the optimal one. For scoring of a subsequence as a possible motif, these algorithms usually use probabilistic models based on either position weight matrices or position specific score matrices. These stochastic algorithms typically contain more information than regular expressions and are well-suited for modeling phenomena poorly represented by regular expressions [24]. Some popular motif discovery tools are MEME [1], CONSENSUS [9], Gibbs sampling [17, 14], MotifSampler [25], AlignACE [20], and BioProspector [15]. Of these algorithms, AlignACE, BioProspector and MotifSampler are based on Gibbs sampling method while MEME is based on expectation maximization technique.

Recently, another stochastic approach, genetic algorithms, has been used for identification of motifs in multiple unaligned DNA sequences [5, 3, 6]. In these evolutionary computation methods, different methods of fitness calculation are used and only one motif per sequence is assumed. However, in a sequence, multiple similar motifs may exist, and identification of those motifs is equally important to identification of a single motif per sequence.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

The(l, d)-motif discovery problem, as introduced by Pevzner and Sze [18], is a mathematical abstraction of the biological motif discovery problem, and it was proposed to highlight the limitations of the most motif recognition algorithms. The (l, d)-motif problem is defined as the discovery of the locations of the l-length motifs in biological sequences that are all mutated at most d positions in the original motif. For this problem, various approaches have been proposed, e.g. graphical approach [26], voting algorithm [4], random projections [2], clustering algorithm [24], etc. Most of these algorithms use exhaustive searches to determine all the (l, d) motifs embedded in different sequences for a shorter l.

In this paper, we propose a novel genetic algorithm based method for identification of multiple (l, d) motifs in each of the given sequences. The main advantage of our proposed method is that it can handle longer motifs and can identify multiple positions of the motif instances of a consensus motif. We then extend this method for detection of weakly conserved regions in unaligned DNA sequences using an alignment score metric, which is an extension of the 'information content' method [23, 9]. We performed experiments on simulated and real data to demonstrate the effectiveness of our proposed method. The experimental results show that our method is able to detect multiple (l, d) motif instances in different sequences, can cope with longer motifs, and can identify weakly conserved regions in the given sequences effectively.

2. TERMS AND NOTATIONS

We use the term *motif instances* to mean the (l, d) degenerative motifs of length l that are mutated at most d symbol positions of an original motif. The pattern shared by these motif instances is referred to as a *consensus* or a *consensus motif*. The population of the genetic algorithm is a collection of these consensus motifs. Sometimes, we use *individual* to mean a consensus motif and denote it by X or Y. We use the term *weakly conserved regions* to mean those motifs that have some positional similarity in their subsequences, and those regions are identified by using an alignment score. Sometimes, we use *motif* to mean a single motif instance or a family of motif instances (like 'CRP' motif).

Throughout this paper, we use L_i to denote the length of the sequence i, N to denote the number of sequences and m_i to denote the number of motif instances in sequence i that are mutated at p positions where $0 \leq p \leq d$ or $d < p = d_{min}$. d_{min} is the minimum number of mutations needed to get the original consensus motif from a degenerative motif. Other terms and notations are described in places of their uses.

3. IDENTIFICATION OF MULTIPLE MOTIFS

Our multiple motif discovery starts with a candidate consensus motif. For this motif, we scan all the sequences one by one to detect subsequences that are either at most d mutations away or at minimum distance from the consensus motif. For example, if the given three sequences are:

CAGAGCAACAATTCATTTTCATAGAGAAA,
TAAGAGCAAATTGGCCAATAGCAATT and
AAGAGCACATTTGGCGTATAGCAATCGACTCT,

the $(7, 1)$ degenerative motifs for the consensus AGAGCAA are:

1: AGAGCAA, AGAGAAA;
2: AGAGCAA, ATAGCAA;
3: AGAGCAC, ATAGCAA.

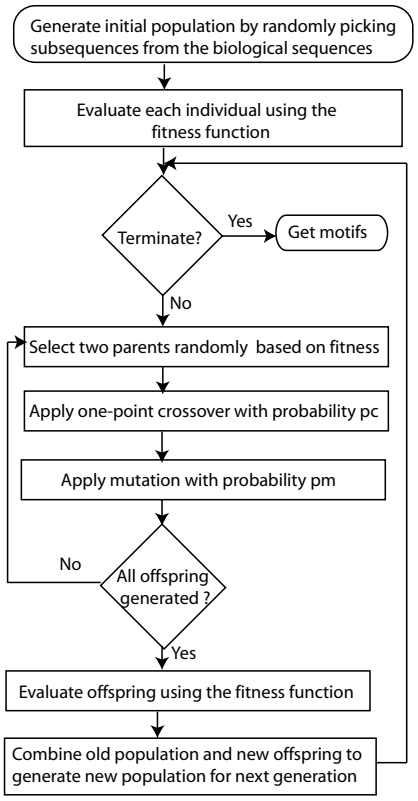

Figure 1: Flowchart of our motif discovery algorithm

Then we score that individual (consensus motif) using the proper fitness function. To maintain a population of multiple consensus motifs and to generate new candidate solutions, we use genetic algorithm [10, 7].

Genetic algorithm (GA) is a population based stochastic method that has been widely used to solve complex functional optimizations. GA starts with a population of randomly generated possible candidate solutions (individuals). The population of individuals is then bred over many generations in a domain independent way using the Darwinian principle of survival of the fittest and an analog of the naturally-occurring genetic operation of crossover. The crossover operation is designed to create syntactically valid offspring from parents that are probabilistically selected based on their fitness at solving the problem at hand [13]. The flowchart of genetic algorithm for motif discovery problem is shown in figure 1. In the next subsections, we discuss the initial population generation, fitness calculation, and offspring generation in a genetic algorithm for the motif discovery problem.

3.1 Initial population generation

Each individual of the population is a fixed-length string of symbols taken from the alphabet. Each of these individuals can be initialized by randomly taking a symbol for

each position. For an alphabet Σ, there are $|\Sigma|^l$ possible candidate consensus motifs; for example, for nucleotide sequences, there are 4^l candidate strings while for amino acid sequences, there are 20^l candidate strings. If we randomly generate an initial population, and population size$<<|\Sigma|^l$, we may miss some strings that are the true motifs of the sequences. Instead, we generate the initial population by randomly picking l-length subsequences from the given biological sequences (nucleotide /amino acid sequences). The motivation behind this is that since the motifs are embedded in the sequences, some subsequences or their mutated subsequences will be the true consensus motifs.

3.2 Evaluation of a candidate consensus motif

Each individual is evaluated depending on the problem at hand. We propose different fitness evaluations for (l,d)-motif discovery problem and for detection of weakly conserved regions.

3.2.1 Fitness calculation for (l,d)-motif problem

For (l,d)-motif discovery problem, we develop a fitness function with the following characteristics:

- it prioritizes a motif instance that has fewer number of mutated symbols from the consensus motif;
- it assigns the better score to a candidate consensus motif that has true motif instances in all sequences;
- it prefers a candidate consensus motif that has more than one motif instance in all sequences.

The fitness function is as follows:

$$fitness(X) = \gamma \sum_{i=1}^{N} \left(\sum_{j=1}^{m_i} \lambda^{(d-d_j)} w_{(j)} \right) \qquad (1)$$

where

γ is a bonus coefficient,
N is the number of sequences,
m_i is the number of motif instances of sequence i that have either $0 \le d_j \le d$ or $d < d_j = d_{min}$,
$\lambda \ge 2$ is a constant used to give distance based weight to a motif, and
$w_{(j)}$ is the frequency based weight of motif j.

The bonus coefficient (γ) is used to give a bonus to a consensus motif that has found at least one (l,d) motif instance in all sequences. This is necessary to distinguish the score of a consensus motif having one motif instance with $d_i = d$ in each sequence from the score of a consensus motif having some motif instances with $d_i = 0$ in some sequences. Let us give an example. Suppose that there are 6 sequences, and we are looking for (6,1) motifs in the sequences. If a consensus motif X has motif instances with distance 1 in all sequences, its score will be 6 ($w_{(1)} = 1$, $\lambda = 2$ and excluding γ). If another consensus motif Y has motif instances with distance 0 in 3 sequences and with distance 2 in other three sequence, its score will be greater than 6. In this case, Y will be assigned better fitness than X; however, X is more preferable to Y. Therefore, we need to give a bonus to X to make it better than Y. The bonus coefficient is calculated as follows:

$$\gamma = \begin{cases} B & \text{if each sequence has at least one } (l,d) \text{ motif;} \\ 1 & \text{otherwise} \end{cases}$$

where B is the bonus value, and its value should greater than or equal to λ^d (it is derived from the equation: $N\gamma > (N-1)\lambda^d$).

The frequency based weight $w_{(j)}$ is used to give preference to identification of motifs of similar types in different sequences than identification of motifs of the same type in a single sequence. This is needed because identification of similar type of motifs in co-regulated genes is biologically more important than identification of same type of motifs within a single sequence. During calculation of weights for multiple motifs within a single sequence, we assume that the motifs are sorted in ascending order of their distances from the consensus sequence; the minimum-distance motif gets the highest weight. The weight of a motif j is calculated as follows:

$$w_{(j)} = \begin{cases} \beta^{(1-j)} & \text{if } d_j \le d \text{ for } j = 1,2,\ldots,m_i, \\ \beta^{(d-d_{min}+1-j)} & \text{if } d < d_j = d_{min},\ j = 1,\ldots,m_i, \end{cases}$$

where $\beta > 1$ is a constant, and d_{min} is the distance of the motifs that need the minimum d_{min} mutations to get the consensus sequence X (individual). If for all motifs, $d < d_j = d_{min}$, we do not need to sort the motifs; otherwise, we have to sort them so that the best matching motif gets the highest weight. For example, suppose that we have got seven (10,2) motifs in a sequence with distances as follows:

Motif	1	2	3	4	5	6	7
Distance	2	2	0	1	0	1	1

If we do not sort them, the first motif with distance 2 will get wrongly the highest weight. Therefore, we sort them and assign their weights as follows:

j	1	2	3	4	5	6	7
Motif	3	5	4	6	7	1	2
Distance	0	0	1	1	1	2	2
Weight	1	$\frac{1}{\beta}$	$\frac{1}{\beta^2}$	$\frac{1}{\beta^3}$	$\frac{1}{\beta^4}$	$\frac{1}{\beta^5}$	$\frac{1}{\beta^6}$

The score will be:

$$\lambda^2(1+\frac{1}{\beta}) + \lambda(\frac{1}{\beta^2}+\frac{1}{\beta^3}+\frac{1}{\beta^4}) + (\frac{1}{\beta^5}+\frac{1}{\beta^6}).$$

However, if the above unsorted motifs are embedded in three sequences with motifs 2 and 3 in sequence 1, motifs 1 and 4 in sequence 2, and motifs 5, 6 and 7 in sequence 3, the score would be

$$(\lambda^2+\frac{1}{\beta}) + (\lambda+\frac{1}{\beta}) + (\lambda^2+\lambda\frac{1}{\beta}+\lambda\frac{1}{\beta^2}).$$

This score is better than the above score, and this one will get preference over the above one.

3.2.2 Fitness calculation for weakly conserved regions

The (l,d)-motif discovery algorithm can be used to identify weak motifs when exact value of d is known but in practice, we do not know how many positions are changed in a degenerative motif. If we assume very small d, we may not find any motif instance in any sequence; if we assume very large d, we will come across many degenerative motifs. For identification of weakly conserved regions, we need some kind of alignment of the degenerative motifs identified by the candidate consensus motif.

The two genetic algorithms [5, 3] that identify weak motifs from unaligned sequences search for the positions of the

motifs stochastically in the given sequences. Therefore, the individuals in the population are vectors of positions and a single motif per sequence is assumed. Afterwards, the consensus motif is determined by majority voting. In our approach, we start with a possible consensus motif (individual), search similar motifs in each sequence, and cluster those motifs of the sequences using an alignment score. Finally, we choose the cluster that has the highest alignment score as the representative of the all the clusters in an individual.

3.2.2.1 Alignment of subsequences.

Several methods for alignment of multiple sequences have been proposed in the literature. Hertz and Stormo [9] have proposed a relative entropy based score metric for alignment of multiple sequences. The score metric is as follows:

$$I_{align} = \sum_{i=1}^{l} \sum_{b \in \Sigma} f_{b,i} \log_k \frac{f_{b,i}}{p_b} \quad (2)$$

where l is the length of each sequence, $f_{b,i}$ is the observed frequency of the symbol b at position i, Σ is the alphabet of symbols (for DNA sequences, $\Sigma = \{A, C, G, T\}$), k is the base of logarithm, and p_b is the background probability distribution of symbol b. When all the symbols at each position are the same, the information content will be the highest, and the sequences will be optimally aligned. However, due to linear summation of each positional information content, the alignment may not be optimal. For example, consider two alignments of 4 DNA sequences of length 2: {AT, AC, AG, AA} and {AC, TC, AG, TG}. The information content of the first alignment using $k = 2$ is

$$I_{align1} = 1.0 * log_2 \frac{1.0}{0.25} + 4 * 0.25 * log_2 \frac{0.25}{0.25} = 2$$

while that of the second alignment is

$$I_{align2} = 2 * 0.5 * log_2 \frac{0.5}{0.25} + 2 * 0.5 * log_2 \frac{0.5}{0.25} = 2.$$

Using these scores, we can not determine the best alignment (alignment 1) that has a conserved position (position 1). We propose non-linear combination of positional information content as the score of an alignment. The score of an alignment is calculated as follows:

$$I_{align} = \sqrt[\alpha]{(I_1^\alpha + I_2^\alpha + \cdots + I_l^\alpha)/l} \quad (3)$$

where α is a positive integer greater than 1, and I_i is the information content at position i. Using the score metric (3), we get the scores of the above two alignments as follows ($\alpha = 2$):

$$I_{align1} = \sqrt{(I_1^2 + I_2^2)/2} = \sqrt{(2^2 + 0^2)/2} = 1.4142;$$

$$I_{align2} = \sqrt{(I_1^2 + I_2^2)/2} = \sqrt{(1^2 + 1^2)/2} = 1.0.$$

Since, $I_{align1} > I_{align2}$, the first alignment will be chosen as the optimum alignment, which is desired.

3.2.2.2 Clustering and scoring of the consensus motif.

For each consensus motif (individual) X, the following steps are performed:

1. In each sequence, the subsequences that are at minimum distance from X are determined. Let those subsequences are $\{S_{11}, S_{12}, \ldots S_{1n_1}\}, \cdots, \{S_{N1}, S_{N2}, \ldots S_{Nn_N}\}$ where S_{ij} is the subsequence j of sequence i, and n_i is the number of minimum distance subsequences of sequence i. Note here that $distance(X, S_{i1}) = distance(X, S_{i2}) = \cdots = distance(X, S_{in_i})$.

2. Next the number of clusters (groups) is determined. In our method:

   $$\text{Number of clusters (c)} = \max\{n_1, n_2, \ldots, n_N\}.$$

   Let the clusters are $C_1, C_2, \ldots, C_c$. Each of these clusters is initialized by taking one subsequence from the largest group of subsequences $\{S_{i1}, S_{i2}, \ldots, S_{ic}\}$. Then the consensus sequence X is added to each cluster.

3. Subsequences of the remaining sequences are added to the clusters. There are many possible ways to add the remaining subsequences to the clusters. One possible way is determination of the cluster that will give the best alignment for each subsequence; we use *sequence* $\rightarrow$ *cluster* to denote this method. Another way is the finding of the subsequence that best aligns with each cluster; we use *cluster* $\rightarrow$ *sequence* to denote this method. In the second case, one subsequence may be included in more than one cluster.

4. The clusters are saved, and the overall alignment score is calculated. This alignment score is used as the fitness of X.

Let us give an example of clustering of minimum-distance subsequences. Suppose the three given DNA sequences are: {AGAGCGACCGGAACCGTGCCCGGGACTGTATAAT, AAACGAAAATACCGGGACCGGCGAAACCGGGA CAGTTCAACTGGGACCG, CTGGGACCGATTCTA CAAGTTTCCTTTTCTTA}, and X=CCGGGACCG. The minimum distance subsequences are:

Sub_1 ={CCGGAACCG, CCGGGACTG};
Sub_2 ={CCGGGACCG,CCGGGACAG,CTGGGACCG};
Sub_3 ={CTGGGACCG}.

Since Sub_2 has the maximum number of subsequences, we create 3 clusters containing its subsequences:

CCGGGACCG	CCGGGACAG	CTGGGACCG.

Next we add the X to every cluster:

CCGGGACCG	CCGGGACAG	CTGGGACCG
CCGGGACCG	CCGGGACCG	CCGGGACCG.

At this stage, suppose we want to add the subsequences of Sub_1 using *sequence* $\rightarrow$ *cluster* method. In this case, the first subsequence goes to the first cluster, and the second subsequence goes to the second cluster. Thus, the three clusters become:

CCGGGACCG	CCGGGACAG	CTGGGACCG
CCGGGACCG	CCGGGACCG	CCGGGACCG.
CCGGAACCG	CCGGGACTG	

Finally the only subsequence of Sub_3 best aligns with cluster 3. Therefore, the final clusters are:

CCGGGACCG	CCGGGACAG	CTGGGACCG
CCGGGACCG	CCGGGACCG	CCGGGACCG
CCGGAACCG	CCGGGACTG	CTGGGACCG.

After the creation of the clusters, we have to combine them to get the fitness of X. There is no straightforward method to get a combined score. For example, if the clustering method uses *sequence* $\rightarrow$ *cluster*, the number of subsequences per cluster may not be the same. Some clusters may have perfect alignments with fewer number of subsequences; some will have weaker alignments with one subsequence from each DNA sequence. Another issue is how to include the number of clusters in the computation of the fitness of an individual. Calculation using an equation like (3) is not effective because if individual X has only a cluster with perfect alignment with score a and Y has two clusters: one with perfect alignment and one with weak alignment with scores a and b, respectively, fitness of X $(=a)$ will be greater than fitness of Y $(=\sqrt{(a^2+b^2)/2}$. However, Y seems to be a better individual than X. In our experiments, we have used *cluster* $\rightarrow$ *sequence* approach, and therefore each cluster corresponding to an individual contains same number of subsequences. We calculate fitness of an individual as follows:

$$fitness(X) = \max\{score_1, score_2, \ldots, score_c\} \quad (4)$$

where $score_i$ is the alignment score of cluster i, and c is the number clusters in the individual. For alignment score, we have used equation (3).

3.3 Offspring generation

We generate offspring from the selected parents through one point crossover and substitute mutation. For crossover, first a random crossover point is determined; then, the parts after crossing over point are swapped. Suppose two selected parents from the population are as follows:

AATATTCAT|ATCAGTTAGTCTT,
GCCTGCAAG|AGACGCGTTCAAG.

The crossover point is indicated by |. After crossing over, the new offspring would be:

AATATTCAT|AGACGCGTTCAAG,
GCCTGCAAG|ATCAGTTAGTCTT.

For mutation, we apply multiple substitutions' approach for (l, d). First we determine the number of positions to be mutated by randomly picking a value k from $[1, d]$. Then, we randomly choose k positions in the selected individual and mutate them. Suppose the selected parent and the mutation points (in overlined, bold faces) are as follows:

TGTCTTCA$\bar{\mathbf{C}}$TCTGCTATTAGAA$\bar{\mathbf{A}}$CC.

If the randomly chosen symbols for the selected positions are G and C, respectively, the offspring generated through mutation will be:

TGTCTTCA$\bar{\mathbf{G}}$TCTGCTATTAGAA$\bar{\mathbf{C}}$CC.

For detection of weakly conserved regions, we perform a single mutation in the selected individual.

3.4 Dealing with Poly-A and TATA box

In DNA sequences, there are abundant subsequences consisting of something like 'AAA...AAA' (Poly-A) or 'ATA... TAAT' (TATA-box). If precautions are not taken, any motif finding algorithm may terminate with either Poly-A or TATA-box. To cope with this problem, one should count the numbers of A, T, and (A,T) in the consensus and in the sites selected as the minimum distance motifs, and if they are more abundant than a threshold, the fitness of the consensus motif (individual) should be reduced. In our experiments, we set the threshold to $0.70 * l$; however, one should determine it by trial-and-error.

3.5 Complexity of the algorithm

Suppose the lengths of the sequences are $L_1, L_2, \ldots, L_N$, and the length of the consensus motif is l.

For (l, d)-motif discovery problem, the number of searches needed to find all minimum-distance motifs of an individual X is:

$$S_X = (l+1)\sum_{i=1}^{N}(L_i - l).$$

This includes the cost needed to calculate distances and to extract the minimum distance motifs from the sequences. If the population size is P, and the maximum number of generations the algorithm runs is G, the overall searching cost $(SC_{overall})$ of (l, d)-motif discovery problem will be:

$$SC_{overall} = GP(l+1)\sum_{i=1}^{N}(L_i - l). \quad (5)$$

However, to find weakly conserved regions in the minimum distance motifs, we need to cluster the motif instances using an alignment score. Therefore, the cost associate with the fitness of an individual is:

$$S_{cons} = \text{Searching Cost } (S_X) + \text{Clustering Cost } (C_X).$$

The clustering cost can be calculated as follows:

$$C_X = c(m_1 + m_2 + \cdots + m_N) * A$$

where A is the alignment cost, c is the number of clusters and m_i is number of minimum distance subsequences from sequence i. The total cost then becomes:

$$SC_{cons} = GP(S_X + C_X). \quad (6)$$

The total cost of an exhaustive search is $\prod_{i=1}^{N} L_i$, which may be much higher than the cost of our method (equation (6) or (5)) for larger L_i's and N. For example, if $N = 6$, all $L_i = 1000$, $G = 500$, $P = 4000$, $l = 19$, the cost of exhaustive search would be 10^{18} whereas the cost of our method would be approximately 24×10^{10} (ignoring clustering cost).

4. EXPERIMENTS

For our experiments, the values of different parameters are listed in table 1. We have set a very high value to mutation because a consensus motif is the the mutated version of motif instances. We have tried with different values of population size and offspring size, but the results presented in this paper did not change much. However, if the length of either the sequences or the consensus motif is larger, the population size should be increased. We have used a roulette wheel selection method for the selection of parents for crossover and mutation. To generate a new population from the old population and the new offspring, we have used elitism technique (elite size=50%).

We performed experiments on the six sequences of CRP motif, and 33 sequences of ArcA motif taken from `http://dragon.bio.purdue.edu/pmotif/` [11], and on the sequences of LEU3 and MCB transcriptional factors of *Saccharomyces cerevisiae* taken from `http://rulai.cshl.edu/SCPD/` [21].

Table 1: Values of different parameters

Parameters	Values
Population size	1000 (2000)
Offspring size	500 (1000)
Maximum generations	100
Crossover probability	0.5
Mutation probability	0.8
Background frequency (p_b)	0.25
α	2
λ ((l,d)-motif problem)	2
β ((l,d)-motif problem)	10

Table 2: Summary of (10,2) motifs found by our method from the sequences of MCB transcriptional factor

Consensus	Score	#Motifs	Comments
AAAGAATAAA	115.32	18	highest score
AAAGATCAAA	62.10	9	lowest score
AAAGAAAAAA	95.43	29	highest #motifs
CAAGAATATA	71.0	7	lowest #motifs

4.1 (l,d)-Motif discovery

First, we performed experiments to identify (l,d) motifs from the sequences of MCB transcriptional factors. For this problem, we set: $l = 10$, $d = 1$, and $B = 10$. Our algorithm did not find any consensus sequence that has (10,1) motifs in all sequences. The best consensus found is ACGCGTTAAA with score of 6.01 (rounded to two decimal points), and the identified motif instances in the six sequences are:

Seq	Motifs
1	ATGCCTTAAC, ACGCGTAACT, AAGCATTAAT
2	ACGCGTGAAA
3	ACGCGGGTAA, ACGCGTCGGA
4	ACGCGTTCAA
5	ACGCGTTAAA
6	ACGCGTTAAA.

However, by setting $d = 2$, we turned up with 19 consensus motifs that have at least one (10,2) motifs in the sequences of MCB transcriptional factors. Some of the characteristic consensus sequences are shown in table 2. Most of the 19 consensus sequences are sequences of Poly-A/TATA-box. By penalizing the consensus sequences containing Poly-A/TATA-box, we came up with only 10 consensus sequences instead of 19.

Next, we performed experiments on the sequences of ArcA motif family. For this problem, we first made 33 different (61,5) motif instances of the following motif consensus:

GATTAAGCGCAAATAGCGTTTGCTGTGTTAT
TGACAGTTAGCATAAACTAGGTGTGACGTT.

Then we insert these 33 motif instances at random positions of 33 original sequences of ArcA. By running our algorithm with B=32, population size=2000 and offspring size=1000, we found the 16 consensus sequences that had at least one (61,5) motif in each of the modified sequences. The consensus sequences are shown in table 3. Of these, the second one is the original consensus sequence, and the rest 15 consensus sequences are its rotated and/or mutated versions.

Table 4: CRP motifs identified by our method

Seq	Position	Motif
1	136	TATGTTATCCACATCACAA
2	64	AAAGTGAACCATATCTCAA
3	375	TATGTGATTGATATCACAC
4	59	TGTGTGATCGTCATCACAA
5	37	TGTGTGAAGTTGATCACAA
6	137	TCTGTGATTGGTATCACAT

The above experiments demonstrate the effectiveness of our GA based (l,d)-motif discovery algorithm. However, it remains unresolved whether evolution occurred in the algorithm or not. In all cases, we got at least one consensus sequence having at least one (l,d) motif in all the sequences and having higher fitness value. When the algorithm continued running, the best fitness value did not change giving the impression that evolution did not take place at all. To make it sure that evolution occurred, we investigated the number of consensus sequences of real (l,d) motifs in the initial population as well as in the final population. In the case of (10,2)-motif discovery problem of MCB sequences, the number of consensus sequences of (10,2) motifs in the initial population was 10 while that in the final population was 19. In the case of (61,5) motifs of ArcA sequences, the number of proper consensus motifs in the initial population was 2 while that in the final population was 16. This evidence strongly suggests that evolution took place in our algorithm.

4.2 Real motifs from biological sequences

For CRP motif, we collected six sequences each of length 502 from the database mentioned above. The motifs embedded in these sequences are:

- AATGTTATCCACATCACAA;
- AAAGTGAACCATATCTCAA;
- CTTGTGATTCAGATCACAA;
- TGTGTGATCGTCATCACAA;
- TGTGTGAAGTTGATCACAA;
- TATGTGATTGATATCACAC.

By performing experiments, we turned up with the results shown in table 4. The consensus determined by our method is TATGTGATCGATATCACAA. To compare our method, we performed additional experiments with binary GA that searched for the positions of the motifs using the score of equation (2) as fitness of an individual. Each individual was a vector of positions of possible motifs. The values of population size, offspring size, and maximum number of generations were the same as that of our algorithm. However, we set crossover and mutation probability to 0.9 and 0.1, respectively since the algorithm searched for positions instead of consensus sequences. The motifs identified by this binary GA from the sequences of CRP are presented in table 5. Our method identified the second, fourth and fifth motifs correctly but the binary GA identified none of the motifs correctly. Moreover, our method identified the conserved regions more accurately than the binary GA.

For MCB transcriptional factors, we extracted the six sequences from positions -500 to +50 of transcription start site of regulated genes of *Saccharomyces cerevisiae*. The motifs embedded in the sequences are {ACGCGT, ACGCGA, CCGCGT, TCGCGA, ACGCGT, ACGCGT} and the con-

Table 3: Different consensus sequences of (61,5) motifs of ArcA sequences

Serial#	Consensus motifs
1	TGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
2	GATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGTT
3	AGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
4	GGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
5	TGATTAAGCGCAAATAGCGTTTGCTGTGTTGTTGACAGTTAGCATAAACTAGGTGTGACGT
6	TGAATAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
7	TGATTAAGCGCAAATAGCGTTTGCTGTGTTTTTGACAGTTAGCATAAACTAGGTGTGACGT
8	TGAGTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
9	TGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATCAACTAGGTGTGACGT
10	TGATTAAGCGCAAATAGCGTTTGCTGTGTTCTTGACAGTTAGCATAAACTAGGTGTGACGT
11	TGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATGAACTAGGTGTGACGT
12	TGATTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATTAACTAGGTGTGACGT
13	TGGTTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
14	TGCTTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
15	TGTTTAAGCGCAAATAGCGTTTGCTGTGTTATTGACAGTTAGCATAAACTAGGTGTGACGT
16	TGATTAAGCGCAAATAGCGTTTGCTGTGTTGTTGACAGTTAGCTTAAACTAGGTGTGACGT

Table 5: CRP motifs identified by binary GA with fitness defined by equation (2)

Seq	Position	Motif
1	4	ATTAACCGCCCCTGACGAT
2	0	AATCACCTCATTTTTCGCT
3	264	ACACTACTCACATTTAAAT
4	16	AATTCATTAATATTTTAGT
5	11	CATCAATGAGTCCTAAACG
6	69	ATATAATTATTATTAACCT

Table 6: LEU3 motifs identified by our method

Consensus	Seq	Position	Motifs
CCGGGACCGG	1	301	CCGGAACCGG
	2	284	CCGGGACCGG
CCGTAACCGG	1	301	CCGGAACCGG
	2	314	CCGTAACCGG
CCGGAACCGG	1	301	CCGGAACCGG
	2	284	CCGGGACCGG
	2	314	CCGTAACCGG

sensus sequence is WCGCGW. By our method, we got the motifs as {ACGCGT, ACGCGT, ACGCGT, ACGCGT, AC GCGT, ACGCGT}, and the consensus is ACGCGT, which is a true consensus motif; the multiple occurrences of this consensus were correctly identified. With binary GA, we got the following motifs in the six sequences: {TTTCGA, TCACCA, TCACGT, TGACGA, TCACGA, TAACGG}; none of these motifs are the true motifs.

For LEU3 transcriptional factors, we extracted two sequences from positions -500 to +50 of transcription start site of two regulated genes of *Saccharomyces cerevisiae*. The consensus motif is CCGNNNNCGG. The motifs identified by our method are shown in table 6. Besides these motifs, our method also identified other 19 possible motifs (false positives) that had the same score as these motifs of LEU3.

By applying binary GA, we got the following motifs:

- Poly-A and TATA-box penalized: {CGAATCTCTT, CGGTTCTTTT}
- Poly-A and TATA-box are not penalized: {ATAATTATAC, ATACCTTTAC}, {TAATTATACT, TACCTTTACT}.

However, none of these are the motif instances of the consensus motif of LEU3.

5. DISCUSSION

For both the problems that we have addressed in this paper, the starting population is very important. If we start with the individuals that have each been initialized by randomly chosen symbols from the alphabet, we may not find any solution at all. If the alphabet size is $|\Sigma|$, there are $|\Sigma|^l$ possible consensus motifs of length l; of which, some will be the real weak motifs. Neither searching with all possible consensus motifs nor starting with some sequences will be feasible. Instead, if we start with randomly chosen subsequences from different sequences as the starting population, the possibility of finding a good solution will be higher as compared to exhaustive search because original motifs are embedded in the sequences in mutated form, and mutations of some of these sequences may restore the original motif.

In some of our experiments, we have found that the initial population contained some of the desired consensus motif sequences with high scores, and in subsequent generations, the best motif did not change, and in the final population, some of the motif consensus sequences were lost. However, evolution took place, and new consensus sequences were produced. To prevent loss of some initial motifs and to keep diversity, we can save in each generation those motifs that have higher score than the threshold and continue searching.

In our method for identification of conserved regions, we have used the maximum value of alignment scores of different clusters. Due to this reason, our method will not distinguish between an individual having one cluster with perfect matching subsequences, and another individual with two clusters–one having perfect matching sequences and another one having less perfect matching, though the second one is the best individual. By using a fitness function like the fitness function of (l, d)-motif discovery problem, this limitation may be overcome. We want to address this problem in our future works.

Prevention of selection of motifs containing either Poly-A or TATA-box is necessary because in DNA sequences, these type of sequences are abundant, and if we do not take measures to cope with them, we may not be able to identify potential motif and the motif instances. However, what threshold should be used to prevent their selection is problem specific and may be determined through trial-and-error.

6. CONCLUSIONS

In this paper, we have proposed a novel genetic algorithm based method for identification of multiple weak motifs in multiple biological sequences. Unlike past practices of finding single motif per sequence, we have emphasized on identification of multiple motif instances of a consensus motif in a single sequence. First, we have shown how multiple (l, d) motifs of a biological sequence should be scored to get a combined fitness of a consensus sequence, and then we have shown how this method can be extended to identify weakly conserved regions in multiple sequences. Moreover, our method is applicable when we do not know the exact value of d for the (l, d)-motif problem. By performing experiments on simulated and real data, we have shown the effectiveness our proposed method.

Though our method seems to be very effective for determination of either (l, d) motifs or weakly conserved regions, it is not 100% perfect for identification of transcriptional factor binding sites in the upstream regions of co-regulated genes. Like other computational methods of motif discovery, our method looks for similar subsequences in multiple biological sequences; many of these similar subsequences have no biological significance. However, some of the more frequent subsequences can be ignored if they follow the random distribution of the whole genome of the species and if some biological knowledge are incorporated in the experiments. In our future work, we want to utilize this information.

7. REFERENCES

[1] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51–80, 1995.

[2] J. Buhler and M. Tompa. Finding motifs using random projections. *J. Comput. Biol.*, 9(2):225–242, 2002.

[3] D. Che, Y. Song, and K. Rasheed. MDGA: Motif discovery using a genetic algorithm. In *Proceedings of GECCO2005*, pages 447–452, 2005.

[4] F. Y. L. Chin and H. C. M. Leung. Voting algorithms for discovering long motifs. In *APBC*, pages 261–271, 2005.

[5] 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 Research*, 32(13):3826–3835, 2004.

[6] 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*, 15:1145–1152, 2005.

[7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[8] D. L. Hartl and E. W. Jones. *Genetics, Analysis of Genes and Genomes*, page 407. Jones and Bartlett Publishers, sixth edition, 2005.

[9] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15:563–577, 1999.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[11] J. Hu, B. Li, and D. Kihara. Limitations and potentials of current motif discovery algorithms. *Nucleic Acids Research*, 33(15):4899–4913, 2005.

[12] I. Jonassen, J. F. Collins, and D. Higgins. Finding fexible patterns in unaligned protein sequences. *Protein Science*, 4(8):1587–1595, 1995.

[13] J. R. Koza and D. Andre. Automatic discovery of protein motifs using genetic programming. In X. Yao, editor, *Evolutionary Computation*, pages 171–197. World Scientific, 1999.

[14] 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*, 262:208–214, 1993.

[15] X. Liu, D. L. Burtlag, and J. S. Liu. Bioprospector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pacific Symposium on Biocomputing*, 6:127–138, 2001.

[16] X. S. Liu, D. L. Brutlag, and J. S. Liu. An algorithm for finding protein-DNA binding sites with applications to chromatin immunoprecipitation microarray experiments. *Nat. Biotechnol.*, 20:835–839, 2002.

[17] A. F. Neuwald, J. S. Liu, and C. E. Lawrence. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Science*, 4:1618–1632, 1995.

[18] P. A. Pevzner and S.-H. Sze. Combinatorial approaches to finding subtle signal in dna sequences. In *Intelligent System for Molecular Biology*, pages 269–278, 2000.

[19] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, 14:55–67, 1998.

[20] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnology*, 16:939–945, 1998.

[21] SCPD: The Promoter Database of Saccharomyces Cerevisiae. URL: http://rulai.cshl.edu/SCPD/.

[22] Sequence motif-Wikipedia. URL: http://en.wikipedia.org/wiki/Sequence_motif.

[23] G. D. Stormo and G. W. Hartzell. Identifying protein-binding sites from unaligned DNA fragments. *Proceedings of National Academy of Science*, 86:1183–1187, 1989.

[24] M. P. Styczynski, K. L. Jensen, I. Rigoutsos, and G. N. Stephanopoulos. An extension and novel solution to the (l,d)-motif challenge problem. *Genome Informatics*, 15(2):63–71, 2004.

[25] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moore, P. Rouze, and Y. Moreau. A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comput. Biol.*, 9:447–464, 2002.

[26] X. Yang and J. C. Rajapakse. Graphical approach to weak motif recognition. *Genome Informatics*, 15(2):52–62, 2004.