Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 1039–1047

Expert Systems with Applications

www.elsevier.com/locate/eswa

MOGAMOD: Multi-objective genetic algorithm for motif discovery

Mehmet Kaya

Department of Computer Engineering, Firat University, 23119 Elazığ, Turkey

Abstract

We propose an efficient method using multi-objective genetic algorithm (MOGAMOD) to discover optimal motifs in sequential data. The main advantage of our approach is that a large number of tradeoff (i.e., nondominated) motifs can be obtained by a single run with respect to conflicting objectives: similarity, motif length and support maximization. To the best of our knowledge, this is the first effort in this direction. MOGAMOD can be applied to any data set with a sequential character. Furthermore, it allows any choice of similarity measures for finding motifs. By analyzing the obtained optimal motifs, the decision maker can understand the tradeoff between the objectives. We compare MOGAMOD with the three well-known motif discovery methods, AlignACE, MEME and Weeder. Experimental results on real data set extracted from TRANSFAC database demonstrate that the proposed method exhibits good performance over the other methods in terms of accuracy and runtime.

Keywords: Motif discovery; Multi-objective genetic algorithms; Transcription factors

1. Introduction

Motif discovery is one of the fundamental problems that have important applications in locating regulatory sites and drug target identification. Regulatory sites on DNA sequence normally correspond to shared conservative sequence patterns among the regulatory regions of correlated genes (D'heaseleer, 2006). These conserved sequence patterns are called motifs. The actual regulatory DNA sites corresponding to a motif are called instances of the motif. Identifying motifs and corresponding instances is very important, so biologists can investigate the interactions between DNA and proteins, gene regulation, cell development and cell reaction under physiological and pathological conditions.

Motifs are usually very short (up to 30 nucleotides) and gapless. But, the regulatory regions that contain regulatory sites are very long (vary from several hundreds to more than 1000 nucleotides). Every instance of a motif normally has the same length, but they may have slightly different sequence compositions. This variability of regulatory site makes biological sense. Better gene expression control can be achieved by having regulatory sites with different intrinsic affinities for regulatory proteins. Biological experimental approaches for finding these regulatory sites include linker-scanning mutagenesis, DNA footprinting, and gel-shift analysis (EMSA) (Hames & Higgins, 1993). Such approaches are tedious and time-consuming. As genome sequencing and gene expression micro-array studies make a large amount of genome sequences and information correlated genes available, computational methods to identify potential regulatory sites must be developed to complement the traditional experimental approaches.

A number of algorithms have been proposed to find motifs in DNA sequences. Consensus (Hertz & Stormo, 1999) Gibbs Sampler (Thompson, 2003) and MEME (Bailey & Elkan, 1994) are the most popular software developments for profile motif discovery. All these approaches are statistical approaches. The fundamental assumption of these kind of approaches is that the motif alignments contained in the data set are those whose letter distribution most differs from the background distribution. Therefore, these approaches are trying to maximize the likelihood

E-mail address: kaya@firat.edu.tr

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.11.008

ratio of the motif model to the background model or the information entropy of the motif model. The log likelihood ratio and information content are the two main scoring functions for profile motif models. Theoretically, the best approach for finding consensus motifs is exhaustive pattern-driven search (Brazma, Jonassen, Eidhammer, & Gilbert, 1998). However, for DNA sequences, there are $|\sum|^W$ patterns to be investigated. This is extremely time-consuming for large W (Pesole, 1992; Tompa, 1999; Van Helden, 1998). Since most of the $|\sum|^W$ patterns are far from the true motif, the sample-driven search (Bailey & Elkan, 1995; Buhler & Tompa, 2001; Fraenkel, 1995; Gelfand, 2000; Hertz & Stormo, 1995; Li, Ma, & Wang, 1999; Pevzner & Sze, 2000), which only explores the patterns related to the given data set may be a very good approximate alternative.

The automatic motif discovery problem is a multiple sequence local alignment problem under the assumption that the motif model gives the optimal score for some appropriate scoring function. Solving this problem is NP-complete theoretically.

There are several cases for this problem:

(1) *The simple sample*: Each sequence in the data set contains exactly one motif instance.
(2) *The corrupted sample*: An instance may not appear in every sequence.
(3) *The invaded sample*: More than one instance may exist in some sequences.
(4) *The multiple patterns*: The sequences may contain more than a single common motif.

To handle the motif discovery problem, including one or more of the above cases, various approaches and tools were proposed (Bailey & Elkan, 1994; Hertz & Stormo, 1999; Pavesi, 2004; Roth, 1998; Sinha & Tompa, 2003; Thijs, 2002; Thompson, 2003).

Recently, another stochastic approach, genetic algorithms (GA), has been used for discovering motifs in multiple unaligned DNA sequences. Of these, Stine (2003) presented a structured GA (st-GA) to evaluate candidate motifs of variable length. Fitness values were assigned as function of high scoring alignment performed with BLAST (Tatusova & Madden, 1999). Liu (2004) developed a program called FMGA for the motif discovery problem, which employed the general GA framework and operators described in SAGA (Notredame & Higgins, 1996). In their method, each individual represents a candidate motif generated randomly, one motif per sequence. Then, Che (2005) proposed a new GA approach called MDGA to efficiently predict the binding sites for homologous genes. The fitness value for an individual is evaluated by summing up the information content for each column in the alignment of its binding site. Congdon (2005) developed a GA approach to motif inference, called GAMI, to work with divergent species, and possibly long nucleotide sequences. The system design reduces the size of the search space as compared to typical window-location approaches for motif inference. They presented preliminary results on data from the literature and from novel projects. Finally, Paul and Iba (2006) presented a GA based method for identification of multiple (l, d) motifs in each of the given sequences. The method can handle longer motifs and can identify multiple positions of motif instances of a consensus motif and can extract weakly conserved regions in the given sequences.

However, all of the above studies employ single objective to discover motifs, although different methods of fitness calculation are used. Also, while in most of the proposed methods the length of motif to be extracted is given beforehand; only one motif per sequence is assumed in some methods. Whereas, multiple similar motifs may exist in a sequence, and identification of those motifs is equally important to the identification of a single motif per sequence. Moreover, almost all of the methods try to find motifs in all of the given sequences. However, some sequences may not contain any motif instance. If it is assumed that a motif instance should be included in all the target sequences, the similarity value used to compare sequences decrease. In this paper, to address all the problems listed and mentioned above, we propose a multi-objective GA based method for motif discovery. The paper demonstrates advantages of multi-objective approach over single-objective ones to discover motifs efficiently and effectively. For this purpose, we use the following three-objective formulation to find a large number of longer and stronger motifs:

$$\begin{aligned}&\text{Maximize similarity},\ \text{Maximize motif length},\\&\text{Maximize support}\end{aligned}$$

The meanings of these objectives will be discussed in the objectives subsection in detail. Next, we compare the three-objective formulation with the single-objective approach based on the following weighted scalar-objective function:

$$\begin{aligned}\text{Maximize}\quad & w_1\cdot\text{Similarity}+w_2\cdot\text{Motif length}\\&+w_3\cdot\text{Support}\end{aligned}$$

We performed experiments on three real data sets to demonstrate the effectiveness of our method. The experimental results show the superiority of the proposed method, in terms of motif length, strongness and the runtime required to find the motifs, over three well-known motif discovery methods, AlignACE, MEME and Weeder.

2. Multi-objective optimization

Many real world problems involve multiple measures of performance or objectives, which should be optimized simultaneously. Multi-objective optimization (MOO) functions by seeking to optimize the component of a vector-valued-objective function. Unlike single-objective optimization, the solution to a MOO problem is a family of points known as the Pareto-optimal set. Each point in the set is optimal in the sense that no improvement can be achieved in one component of the objective vector that does not lead to degradation in at least one of the remaining components. Given a set of possible solutions, a candidate is said to be Pareto-optimal if there are no other solutions in the solutions set that can dominate any of the candidate solutions. In other words, the candidate solution would be a non-dominated solution. A general minimization problem of M objectives can be mathematically stated as

$$\begin{aligned}&\text{Minimize} f(x) = [f_i(x), i = 1, \dots, M]\\&\text{subject to :}\\&\qquad g_j(x) \leqslant 0 \quad j = 1, 2, \dots, J\\&\qquad h_k(h) = 0 \quad k = 1, 2, \dots, K\end{aligned}$$

where $f_i(x)$ is the ith-objective function, $g_j(x)$ is the jth inequality constraint. The MOO problem then reduces to finding x such that $f(x)$ is optimized.

In general, the objectives of the optimization problem are often conflicting. Optimal performance according to one objective, if such an optimum exists, often implies unacceptable low performance in one or more of the other objective dimensions, creating the need for a compromise to be reached. A suitable solution to such problems involving conflicting objectives should offer *acceptable*, though possibly sub-optimal in the single-objective sense, performance in all objective dimensions, where acceptable is a problem dependent and ultimately subjective concept. An important concept in MOO is that of domination, where a solution x_i is said to dominate another solution x_j if both the following conditions are true:

- The solution x_i is not worse than x_j in all objectives.
- The solution x_i is strictly better than x_j in at least one objective.

This, in turn, leads to the definition of *Pareto-optimality*, where a decision vector $x_i \in U$, where U stands for the universe, is said to be *Pareto-optimal* if and only if there exists no x_j, $x_j \in U$, such that x_i is dominated by x_j. Solution x_i is said to be *nondominated*. The set of all such *nondominated* solutions is called the *Pareto-optimal set* or the *nondominated set*. In general, MOO problems tend to achieve a family of alternatives which must be considered the relevance of each objective relative to the others (Zitzler, 2000). Recently, some researchers have studied on different problems by using multi-objective genetic algorithms. We have already participated in some of these efforts with data mining area (Kaya & Alhajj, 2004a, 2004b; Kaya, 2006).

3. The MOGAMOD algorithm

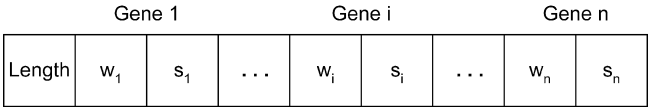
We use a well-known high-performance multi-objective genetic algorithm called NSGA II (Deb, 2002) to find a large number of motifs from biosequences with respect to three objectives, which will be discussed in the objectives subsection. The NSGA II algorithm is also employed in the two-objectives case. On the other hand, we use a standard single-objective GA with a single elite solution in the case of the single-objective formulation. We first give background information about it in the following because our algorithm is based on GAs.

A GA is a search and optimization methodology from the field of evolutionary computation that was invented by Holland (1975). A GA is based on the Darwin's natural selection principle of the survival of the fittest, and is widely used for hard problems in engineering and computer science. A GA is a population-based method where each individual of the population represents a candidate solution for the target problem. This population of solutions is evolved throughout several generations, starting from a randomly generated one, in general. During each generation of the evolutionary process, each individual of the population is evaluated by a fitness function, which measures how good the solution represented by the individual is for the target problem. From a given generation to another, some parent individuals (usually those having the highest fitness) produce "offsprings", i.e., new individuals that inherit some features from their parents, whereas others (with low fitness) are discarded, following Darwin's principle of natural selection. The selection of the parents is based on a probabilistic process, biased by their fitness value. Following this procedure, it is expected that, on average, the fitness of the population will not decrease every consecutive generation. The generation of new offsprings, from the selected parents of the current generation, is accomplished by means of genetic operators. This process is iteratively repeated until a satisfactory solution is found or some stop criterion is reached, such as the maximum number of generations.

3.1. Structure of the individuals

An individual in MOGAMOD represents the starting locations of a potential motif on all the target sequences. An individual expect for its part showing the motif length is divided into n genes, where each gene corresponds to starting location of a motif, if any, in the corresponding sequence. The genes are positional, i.e., the first gene deals with the first sequence, the second gene deals with the second sequence, and so on. Each ith gene, $i = 1, \dots, n$, is subdivided into two fields: weight (w_i) and starting location (s_i), as shown in Fig. 1.

The filed weight (w_i) is a real-valued variable taking values in the range $[0, \dots, 1]$. This variable indicates whether or not the potential motif is present in the corresponding sequence. More precisely, when w_i is smaller than a user-



	Gene 1			Gene i			Gene n	
Length	w_1	s_1	. . .	w_i	s_i	. . .	w_n	s_n

Fig. 1. Representation of an individual.

defined threshold (called *Limit*) the motif will not be extracted from the *i*th sequence. Therefore, the greater the value of the threshold *Limit*, the smaller is the probability that the corresponding sequence will be included in discovering that motif.

The field starting location (s_i) is a variable that indicates the starting location of the potential motif in the *i*th sequence.

In this study, an extra part in each individual was used to determine the length of the motif. The value of this part changes between 7 and 64 because we restricted the minimum and the maximum length of the predicted motif in those values.

Note that the above encoding is quite flexible with respect to the length of the motifs. A traditional GA is very limited in this aspect, since it can only cope with fixed-length motifs. In our approach, although each individual has a fixed length, the genes are interpreted (based on the value of the weight w_i) in such a way that the individual phenotype (the motif) has a variable length. Hence, different individuals correspond to motifs with different length.

The start of the first population consists of generating, arbitrarily, a fixed number of individuals during the evolution.

3.2. Objectives and selection

The fitness of an individual in MOGAMOD is assessed on the basis of similarity, motif length and support.

3.2.1. Similarity

It performs a measure of similarity among all motif instances defining an individual. To calculate it, we first generate a position weight matrix from the motif patterns found by MOGAMOD in every sequence. Then, the dominance value (*dv*) of the dominant nucleotide in each column is found as follows:

$$dv(i) = \max_b\{f(b,i)\}, \quad i = 1,\ldots,l$$

where $f(b,i)$ is the score of the nucleotide *b* on column *i* in the position weight matrix, $dv(i)$ is the dominance value of the dominant nucleotide on column *i*, and *l* is motif length.

We define the similarity objective function of motif *M* as the average of the dominance values of all columns in the position weight matrix. In other words

$$\text{Similarity } (M) = \frac{\sum_{i=1}^{l} dv(i)}{l}$$

The closer the value of the similarity (M) to one, the larger the probability that the candidate motif *M* will be discovered as a motif. The following example shows how to compute the similarity measure in given two position weight matrixes with different size:

Example:

The first matrix (Table 1) implies that the number of target sequences in a dataset may be 5 and the motif length is found to be 4. In such a case, the dominant nucleotide for column 1 is T and its dominance value is 0.6. The dominance values of the other columns are 1. As for the similarity value, it is computed as:

Table 1
The position weight matrix of a motif with length 4

	1	2	3	4
A	0.2	1	0	0
C	0.2	0	1	1
T	0.6	0	0	0
G	0	0	0	0

$$\frac{(0.6 + 1 + 1 + 1)}{4} = 0.9$$

Similarly, in the second matrix (Table 2), the number of target sequences may be 8 and the motif length is determined to be 5. The similarity value is computed as 0.85 for this longer motif.

3.2.2. Motif length

In motif discovery, a motif with large length is always desired because the longer the motif, the lesser is the chance that it simply occurred by chance in the given target sequence.

3.2.3. Support

Here, the meaning of this objective is the same with that in the data mining field. Sometimes, a candidate motif may not appear in every sequence. In other words, one or more sequence may not include any candidate motif. In this case, the aim should become to discover optimal motif, leaving these corrupted sequences out. So, support is the number of sequences composing candidate motifs. The greater the support value, the stronger is the motif covered by most of the sequences in the dataset.

Overall, the optimal motif discovery problem is converted into the following three-objective optimization problem:

$$\text{Maximize similarity(M)},\ \text{Maximize motif length(M)},$$
$$\text{Maximize support(M)}$$

The individuals in a population are first sorted based on their domination status using the procedure nondominated sort (Deb, 2002). Here, the individuals that are not dominated by any other member of the population form the first front, and are put in rank 1. These individuals are then

Table 2
The position weight matrix of a motif with length 5

	1	2	3	4	5
A	0.25	0	0.75	1	0
C	0.5	0	0.25	0	0
T	0.25	1	0	0	1
G	0	0	0	0	0

removed from consideration, and the individuals which thereafter become nondominated are assigned rank 2. The process is repeated until all the individuals have been assigned a rank. The individuals are then put in a sorted order according to their ranks. The overall complexity of the nondominated sort algorithm described in detail in Deb (2002) is shown to be $O\ (M\ N^2)$, where M is the number of objectives and N is the number individuals in the population. The first front represents a nondominated set with respect to the current population, since none of the solutions in this front is dominated by any other solution in the population.

After applying the nondominated sort algorithm, the individuals are assigned a crowding distance and selection is performed using the crowded tournament selection. The crowding distance is computed as the sum of the difference of the objective values of the solutions preceding and following the current solution (corresponding to the individual under consideration) for each objective. This provides a measure of the density of the solution surrounding a particular point in the population.

In crowded tournament selection, a pair of individuals is selected randomly, and the one with the lower rank is selected. If the ranks of the individuals are equal, then the individual with a larger crowding distance is chosen. The large crowding distance ensures that the solutions are spread along the Pareto-optimal front.

3.3. Genetic operators

In MOGAMOD, the usual one-point crossover operator is stochastically applied with a predefined probability, using two individuals of the selected pool. The crossover point is a percentage of the length of the individual that defines the starting point from where the crossover breaks the string. We use arithmetic crossover method in the experiments (Herrera, 1998). The employed method works as follows:

Consider two chromosomes $C_1 = (c_1^1, \ldots, c_n^1)$ and $C_2 = (c_1^2, \ldots c_n^2)$. Applying the crossover operator on C_1 and C_2 generates two offspring $H_1 = (h_1^1, \ldots, h_i^1, \ldots, h_n^1)$ and $H_2 = (h_1^2, \ldots, h_i^2, \ldots, h_n^2)$, where for $i = 1$ to n, $h_i^1 = \lambda c_i^1 + (1 - \lambda)c_i^2$ and $h_i^2 = \lambda c_i^2 + (1 - \lambda)c_i^1$. In our experiments, λ varies with respect to the produced number of generations, as non-uniform arithmetical crossover.

The mutation operator is used to foster more exploration of the search space and to avoid unrecoverable loss of genetic material that leads to premature convergence to some local minima. In general, mutation is implemented by changing the value of a specific position of an individual with a given probability, denominated mutation probability. MOGAMOD developed three mutation operators tailored for our genome representation.

3.3.1. Shift the starting location towards the right

The value in the starting location of a randomly selected gene is increased by one.

3.3.2. Shift the starting location towards the left

The value in the starting location of a randomly selected gene is decreased by one.

3.3.3. Random-changing

The mutation produces a small integer number that is then added to or subtracted from the current content of any of length, weight or starting location. This is implemented in such a way that the lower and upper bounds the domain of the field are never exceeded.

To sum up, MOGAMOD process employed in this study can be summarized by the following algorithm,

3.3.3.1. The algorithm. Input: Population size N; Maximum number of generations G; Crossover probability p_c; Mutation rate p_m.

Output: Nondominated set

```
Step 1: P:=Initialize (P)
Step 2: while the termination criterion is not satisfied do
Step 3: C:=Select From (P)
Step 4: $C^I$:=Genetic Operators (C)
Step 5: P:=Replace ($PUC^I$)
Step 6: end while
Step 7: return (P)
```

First, an initial population P is generated in Step 1. Pairs of parent solutions are chosen from the current population P in Step 3. The set of the selected pairs of parent solutions is denoted by C in Step 3. Crossover and mutation operations are applied to each pair in C to generate the offspring population C^I in Step 4. The next population is constructed by choosing good solutions from the merged population PUC^I. The pareto-dominance relation and a crowding measure are used to evaluate each solution in the current population P in Step 3 and the merged population PUC^I in Step 5. Elitism is implemented in Step 5 by choosing good solutions as members in the next population from the merged solution PUC^I.

4. Experimental results

We conducted some experiments in order to analyze and demonstrate the efficiency and effectiveness of MOGAMOD. Further, the superiority of the proposed approach has been demonstrated by comparing it with three existing motif discovery methods, namely AlignACE (Roth, 1998), MEME (Bailey & Elkan, 1994), and Weeder (Pavesi, 2004). In our experiments, we concentrate on testing the time requirements as well as changes in the main factors that affects the proposed multi-objective process, namely finding nondominated sets, support, length and similarity. All of the experiments have been conducted on a Pentium IV 3.0 GHz CPU with 1 GB of memory and running Windows XP. As data sets are concerned, we used three different data sets of sequences utilized as a benchmark for assessing computational tools for the discovery of transcription factor binding sites (Tompa, 2005), which were selected from TRANSFAC database (Wingender, 1996).

We concentrated our analysis on yst04r, yst08r and hm03r sequence data sets. Further, in all the experiments conducted in this study, MOGAMOD process started with a population of 200 individuals. As the termination criteria, the maximum number of generations has been fixed at 3000. Finally, while the crossover probability is chosen to be 0.8, the mutation rate of 0.3 was used for each kind of mutation. The standard single-objective GA was also executed using the same parameter values; and the weight values $w_1 = 5$, $w_2 = 1$ and $w_3 = 1$.

Three sets of experiments for each data set were carried out. The first set of experiments is dedicated to evaluate the yst04r sequence data set. The data set contains 7 sequences of 1000 bps each. Some nondominated solutions found by MOGAMOD are reported in Table 3. Here, the values of length and similarity of some nondominated solutions are given for four different values of support. As can be easily seen from Table 3, as the support value increases, the motif length decreases. However, for each number of the supports, as the motif length decreases, the similarity value raises. Thus, the tradeoff between the similarity and the motif length is clearly observed for four values of support. Table 3 gives the solution found by the standard single-objective GA process as well.

In the second experiment of the first set, we showed the consensus motif patterns obtained for five different motif discovery methods. Table 4 gives the results of this experiment. An important point here is that while MOGAMOD finds alternative solutions for different values of support, the other methods extract only one motif pattern. For example, CGAGCTTCCACTAA and CGGGATTCCTCTAT are two motifs predicted by MOGAMOD which have the same length but different support and similarity values.

Table 3
The objective values of the nondominated solutions for yst04r

	Support	Length	Similarity
MOGAMOD	4	24	0.76
		20	0.78
		15	0.87
	5	15	0.82
		14^{*}	0.84
	6	14^{+}	0.77
		13	0.81
	7	9	0.80
		8	0.84
Single-objective GA	5	9	0.88

Table 4
Comparisons of the conserved motifs predicted by five different methods for yst04r

Method	Predicted motif
AlignACE	CGGGATTCCA
MEME	CGGGATTCCCC
Weeder	TTTTCTGGCA
Single-objective GA	CTGGCATCC
MOGAMOD	*CGAGCTTCCACTAA $^{+}$CGGGATTCCTCTAT

The final experiment for this set compares the runtimes of four different methods. The results are reported in Fig. 2. The runtime reported for MOGAMOD represents a nondominated solution at the end of 3000 generations. As a result of this experiment, it has been observed that MOGAMOD outperforms the other two approaches for yst04r data set. The runtime of Weeder is not available.

In the second set of the experiments, we applied MOGAMOD on a data set having larger number of sequences, yst08r, as its sequence length is the same with the previous data set. This new data set contains 11 sequences. The first experiment obtains the values of objectives of the nondominated set for yst08r data set. Table 5 reports the results. As can be easily seen from Table 5, for two solutions having the same motif length, the similarity value of the solution whose support value is higher, is lower than that of the other. Another fact seen from the table is that MOGAMOD exhibits the superiority with respect to single-objective GA case because in case the support value is 8, while the motif length of MOGAMOD is 13, that of the single-objective GA is 10. However, the similarity values of both methods are almost the same.

The second experiment deals with comparing the conserved motifs predicted by five different approaches. It can be easily realized from Table 6 that MOGAMOD

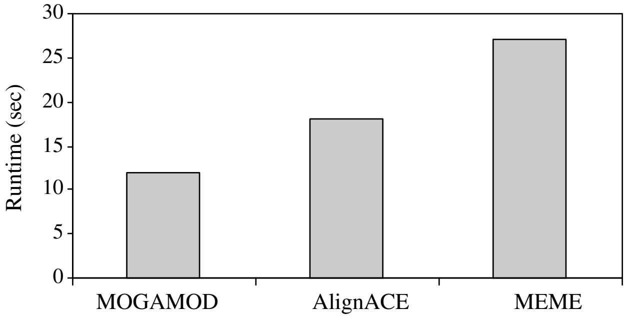


Fig. 2. Comparison of runtimes for yst04r data set.

Table 5
The objective values of the nondominated solutions for yst08r

	Support	Length	Similarity
MOGAMOD	7	20	0.75
		$15^{*}{}_{1}$	0.84
		15^{+1}	0.87
	8	15	0.79
		$14^{*}{}_{2}$	0.83
		13^{+2}	0.85
	9	13	0.82
		12	0.84
	10	12	0.79
		11	0.82
	11	11	0.77
		11	0.80
Single-objective GA	8	10	0.86

Table 6
Comparisons of the conserved motifs predicted by five different methods for yst08r

Method	Predicted motif
AlignACE	CACCCAGACAC TGATTGCACTGA
MEME	CACCCAGACAC
Weeder	ACACCCAGAC
Single-objective GA	AACCCAGACA
MOGAMOD	${}^{*}_{1}$TCTGGCATCCAGTTT ${}^{+1}$GCGACTGGGTGCCTG ${}^{*}_{2}$GCCAGAAAAAGGCG ${}^{+2}$ACACCCAGACATC

Table 7
The objective values of the nondominated solutions for hm03r

	Support	Length	Similarity
MOGAMOD	5	38	0.70
		25	0.77
	6	25	0.71
		22	0.76
	7	22^{*}_{1}	0.74
		18^{+1}	0.82
	8	18	0.76
		13	0.81
	9	13	0.77
		11	0.78
	10	11^{*}_{2}	0.74
		10^{+2}	0.79
		9^{-2}	0.81
Single-objective GA	7	14	0.84

Table 8
Comparisons of the conserved motifs predicted by five different methods for hm03r

Method	Predicted motif
AlignACE	TGTGGATAAAAAA
MEME	AGTGTAGATAAAAGAAAAAC
Weeder	TGATCACTGG
Single-objective GA	AATGCAGATAAAGG
MOGAMOD	${}^{*}_{1}$TATCATCCCTGCCTAGACACAA ${}^{+1}$TGACTCTGTCCCTAGTCT ${}^{*}_{2}$TTTTTTCACCA ${}^{+2}$CCCAGCTTAG ${}^{-2}$AGTGGGTCC

and AlignACE produce multiple motifs. Furthermore, Table 6 shows two multiple motifs discovered by MOGAMOD whose lengths are different, as well. The length of the first set of multiple motifs is 15 and their support value is 7 while the lengths of the second set of multiple motifs are 14 and 13, and their support value is 8.

The third experiment investigates the runtimes of the proposed method and the other approaches in the case of a data set having large number of sequences. The results in Fig. 3 demonstrates that the performance of MOGAMOD decreased with respect to the previous data set. This is an expected result since the length of individuals in population raises with increasing sequence number in the data set.

The last set of experiments is dedicated to testing the performance of our method on a data set having longer sequence lengths as the sequence number is almost the same with the previous data set, which is only changing from 11 to 10. The results are shown in Table 7,8 and Fig. 4. Table 7 gives the nondominated solutions found by MOGAMOD for six different values of support and obtained by single-objective GA. The tradeoff observations are similar to those described above for the two data sets. However, it should be noted that as different from the previous two sets of experiments, longer sequences provide larger number of multiple motifs. The consensus motifs predicted by this and the other methods are exhibited in Table 8. In the same support number, MOGAMOD finds three multiple motifs whose lengths are 9, 10 and 11. Finally, Fig. 4 shows the runtimes of all the methods. It should be noted that the performance of MOGAMOD in this data set increased with respect to those of the other data sets and the value of the runtime remained almost the same. This is true because the individuals in the population are generated independent of the sequence length. However, if the population size is increased slightly, the diversity of the population is performed better. This means that more appropriate individuals which have higher prediction accuracy can be obtained. But, the runtimes of the other methods increases exponentially with the length of the sequences.

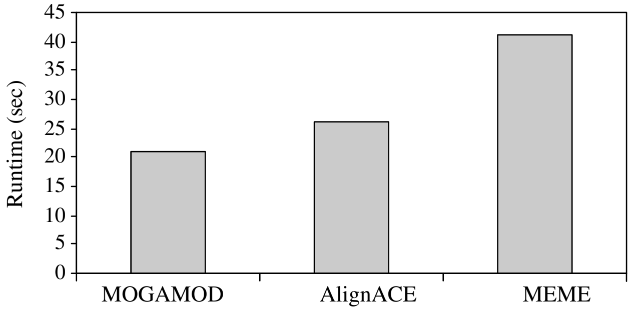


Fig. 3. Comparison of runtimes for yst08r data set.

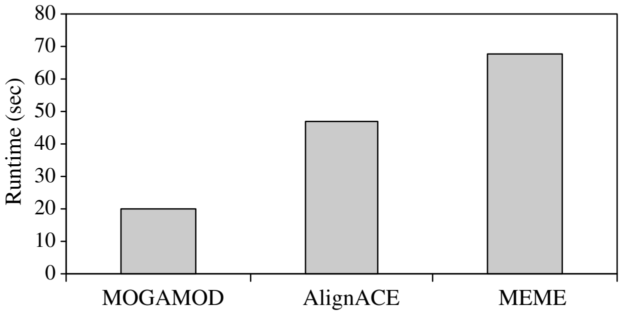


Fig. 4. Comparison of runtimes for hmr03r data set.

In the last experiments, we examine the nucleotide-level correlation coefficient (nCC) by species for every four methods in TRANSFAC database. The results shown in Fig. 5

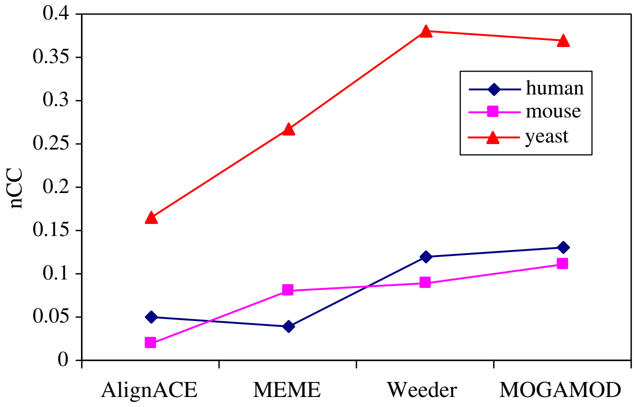


Fig. 5. Comparison of the nucleotide-level correlation coefficients (nCC) by species for four methods.

demonstrates that MOGAMOD outperformed almost all the well-known methods while it was slightly inferior to the Weeder's result at yeast dataset.

5. Discussion and conlusions

In this paper, we contributed to the ongoing research by proposing a multi-objective GA based method for discovering optimized motifs (MOGAMOD) with respect to criteria we defined. These criteria are similarity of instances, predicted motif length and support exhibiting the strongness of motif.

MOGAMOD includes five contributions. First, the algorithm is equally applicable to any variety of sequential data. Second, it allows arbitrary similarity measure. Although we used relatively a simple similarity measure in the paper, it can be easily changed or extended. Another contribution is that a large number of nondominated sets are obtained by its single run. Thus, the decision maker can understand the tradeoff between the similarity, motif length and support by the obtained motifs. Next, by MOGAMOD, more than one instance may be discovered in the same sequence and multiple motifs may be extracted. Fourth, the optimal motifs are obtained without giving motif length. Finally, MOGAMOD outperforms the two well-known motif discovery methods in terms of runtime.

The experiments conducted on three data sets illustrated that the proposed approach produces meaningful results and has reasonable efficiency. The results of three data sets are consistent and hence encouraging. MOGAMOD can also be directly applied to other diverse types of sequential data sets, or it can be extended to address problems not yet considered.

The similarity measure used in our approach is simple. In the future, we will revise our similarity measure to make this score more realistic, improve our algorithm such that one could have better performance in lower similar sequences, and experiment with the realistic data sets having longer sequences and large number of sequences. Currently, we are also investigating how to find gapped motifs.

Acknowledgements

This study was supported by TUBITAK (The Scientific and Technological Research Council of Turkey) under Grant No. 106E199.

References

Bailey, T. L., & Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the second international conference ISMB, USA* (pp. 28–36).

Bailey, T. L., & Elkan, C. (1995). Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning, 21*, 51–80.

Brazma, A., Jonassen, I., Eidhammer, I., & Gilbert, D. (1998). Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology, 5*, 279–305.

Buhler, J., & Tompa, M. (2001). Finding motifs using random projections. In *Proceedings RECOMB'01, Canada* (pp. 69–76).

Che, D. (2005). MDGA: Motif discovery using a genetic algorithm. In *Proc. GECCO'05, USA* (pp. 447–452).

Congdon, C. B. (2005). Preliminary results for GAMI: A genetic algorithms approach to motif inference. In *Proc. CIBCB'05, USA* (pp. 1–8).

Deb, K. et al. (2002). A fast and elitist multi-objective genetic algorithm: NSGA II. *IEEE Transactions on Evolutionary Computation, 6*, 182–197.

D'heaseleer, P. (2006). What are DNA sequence motifs? *National Biotechnology, 24*, 423–425.

Fraenkel, Y. et al. (1995). Identification of common motifs in unaligned DNA sequences: Application to *Escherichia coli* Irp regulon. *Computer Applications in the Biosciences*, 11.

Gelfand, M. et al. (2000). Prediction of transcription regulatory sites in archaea by a comparative genomic approach. *Nucleic Acids Research, 28*, 695–705.

Hames, B. D., & Higgins, S. J. (1993). *Gene transcription: A practical approach*. Inc., NewYork: Oxford University Press.

Herrera, F. et al. (1998). Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review, 12*, 265–319.

Hertz, G. Z., & Stormo, G. D. (1995). Identification of consensus patterns in unaligned DNA and protein sequences: A large-deviation statistical basis for penalizing gaps. In *Proceedings of the international conference on bioinformatics and genome research* (pp. 201–216).

Hertz, G. Z., & Stormo, G. D. (1999). Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics, 15*, 563–577.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: Univ. Mich. Press.

Kaya, M., & Alhajj, R. (2004a). Integrating multi-objective genetic algorithms into clustering for fuzzy association rules mining. In: *IEEE international conference on data mining (ICDM 2004), Brighton, UK, 1–4 November 2004*.

Kaya, M., & Alhajj, R. (2004b). Multi-objective genetic algorithm based approach for optimizing fuzzy sequential patterns. In *16th IEEE international conference on tools with artificial intelligence*. Boca Raton: FL, USA.

Kaya, M. (2006). Multi-objective genetic algorithm based approaches for mining optimized fuzzy association rules. *Soft Computing Journal, 10*(7), 578–586.

Li, M., Ma, B., & Wang, L. (1999). Finding similar regions in many strings. In *Proceedings STOC, USA* (pp. 473–482).

Liu, F. M. (2004). FMGA: Finding motifs by genetic algorithm. In *Proceedings BIBE'04, Taiwan* (pp. 459–466).

Notredame, C., & Higgins, D. G. (1996). SAGA: Sequence alignment by genetic algorithm. *Nucleic Acids Research, 24*, 1515–1524.

Paul, T. K., & Iba, H. (2006). Identification of weak motifs in multiple biological sequences using genetic algorithm. In *Proceedings GECCO'06, USA* (pp. 271–278).

Pavesi, G. et al. (2004). Weeder Web: Discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Research, 32*, W199–W203.

Pesole, G. et al. (1992). Wordup: An efficient algorithm for discovering statistically significant patterns in DNA sequences. *Journal of Nucleic Acids Research, 20*, 2871–2875.

Pevzner, P., & Sze, S. H. (2000). Combinatorial approaches to finding subtle in DNA sequences. In *Proceedings ISMB'00* (pp. 269–278).

Roth, F. P. et al. (1998). Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *National Biotechnology, 16*, 939–945.

Sinha, S., & Tompa, M. (2003). YMF: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research, 31*, 3586–3588.

Stine, M. (2003). Motif discovery in upstream sequences of coordinately expressed genes. *CEC'03, USA* (pp. 1596–1603).

Tatusova, T. A., & Madden, T. L. (1999). Blast2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiology Letters, 2*, 247–250.

Thijs, G. et al. (2002). A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *Journal of Computational Biology, 9*, 447–464.

Thompson, W. et al. (2003). Gibbs recursive sampler: Finding transcription factor binding sites. *Journal of Nucleic Acids Research, 31*, 3580–3585.

Tompa, M. (1999). An exact method for finding short motifs in sequences with application to the ribosome binding site problem. In *Proceedings of the international conference on ISMB, Germany* (pp. 262–271).

Tompa, M. et al. (2005). Assessing computational tools for the discovery of transcription factor binding sites. *National Biotechnology, 23*, 137–144.

Van Helden, J. et al. (1998). Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology, 281*, 827–842.

Wingender, E. et al. (1996). TRANSFAC: A database on transcription factors and their DNA binding sites. *Nucleic Acids Research, 24*, 238–241.

Zitzler, E. et al. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation, 2*, 173–195.