BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 13 2006, pages 1577–1584
doi:10.1093/bioinformatics/btl147

Sequence analysis

GAME: detecting *cis*-regulatory elements using a genetic algorithm

Zhi Wei[1,*] and Shane T. Jensen[2]

[1]Genomics and Computational Biology Graduate Group, University of Pennsylvania School of Medicine and [2]Department of Statistics, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104, USA

Received on January 10, 2006; revised on March 24, 2006; accepted on April 12, 2006
Advance Access publication April 21, 2006
Associate Editor: Martin Bishop

ABSTRACT

Motivation: Identification of a transcription factor binding sites is an important aspect of the analysis of genetic regulation. Many programs have been developed for the *de novo* discovery of a binding motif (collection of binding sites). Recently, a scoring function formulation was derived that allows for the comparison of discovered motifs from different programs [S.T. Jensen, X.S. Liu, Q. Zhou and J.S. Liu (2004) *Stat. Sci.*, **19**, 188–204.] A simple program, BioOptimizer, was proposed in [S.T. Jensen and J.S. Liu (2004) *Bioinformatics*, **20**, 1557–1564.] that improved discovered motifs by optimizing a scoring function. However, BioOptimizer is a very simple algorithm that can only make local improvements upon an already discovered motif and so BioOptimizer can only be used in conjunction with other motif-finding software.

Results: We introduce software, GAME, which utilizes a genetic algorithm to find optimal motifs in DNA sequences. GAME evolves motifs with high fitness from a population of randomly generated starting motifs, which eliminate the reliance on additional motif-finding programs. In addition to using standard genetic operations, GAME also incorporates two additional operators that are specific to the motif discovery problem. We demonstrate the superior performance of GAME compared with MEME, BioProspector and BioOptimizer in simulation studies as well as several real data applications where we use an extended version of the GAME algorithm that allows the motif width to be unknown.

Availability: http://mail.med.upenn.edu/~zhiwei/GAME/

Contact: zhiwei@mail.med.upenn.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

A crucial component of gene regulation is the binding of transcription factor proteins to specific locations on the genomic sequence in close proximity (100–1000 bps) to a target gene, leading to changes in transcriptional activity for that gene. These specific locations (binding sites) are short (≤30 bps) and share a common sequence of nucleotides, though there is usually some variability in sequence between binding sites. These binding sites must have a specific enough shared pattern so that the TF protein does not bind to many random locations throughout the genome, but the specificity cannot be absolute in that varying binding affinities between the transcription and its target sites are required for different genes. The collection of sequences that can act as binding sites for a particular transcription factor is called its binding motif. In many cases, the locations of these binding sites must be discovered without prior knowledge of their motif appearance, which we refer to as *de novo* motif discovery.

Experimental validation of these binding site locations is labor-intensive, which makes computational approaches an attractive alternative for *de novo* motif discovery. The benefit of high-throughput approaches has grown even more in recent years, with the dramatic increase in availability of annotated genomic sequences for many related species, leading to *de novo* motif discovery across sequences from different species (e.g. McCue *et al*., 2001; Jensen *et al*., 2005). Many computational approaches are based on the formulation of a binding motif as a position-specific weight matrix (PWM), which contains the desirability (relative frequency) of each nucleotide at each position of a binding site of fixed width.

Popular motif discovery algorithms based on a PWM formulation include AlignACE (Roth *et al*., 1998), BioProspector (Liu *et al*., 2001), Consensus (Hertz and Stormo, 1999), Gibbs Motif Sampler (Liu *et al*., 1995; Neuwald *et al*., 1995), MDscan (Liu *et al*., 2002) and MEME (Bailey and Elkan, 1994). Jensen *et al*. (2004) provide a review of the statistical models upon which these programs are based. It is often unclear which software should be used, however, since the relative performance of each program varies between real-data applications. In an effort to alleviate this uncertainty, Jensen and Liu (2004) proposed a scoring function formulation, based on a comprehensive Bayesian model, that can be used to quantitatively evaluate the fitness of discovered motifs, thereby allowing for comparison between the results from different programs. In addition to providing this scoring function, Jensen and Liu (2004) presented the BioOptimizer program, which can be used to locally optimize discovered motifs with respect to this scoring function. The scoring function was also extended to allow for unknown site abundance and unknown motif width. However, because BioOptimizer is based on a simple hill-climbing algorithm, it is reliant on a good starting motif which must be provided by one of the aforementioned *de novo* motif discovery programs.

Our goal is an optimization algorithm for motif discovery that is capable of a more exhaustive search of the space of possible motifs, and thus eliminates this dependence on the use of other motif discovery programs. We present a program, GAME, which

[*]To whom correspondence should be addressed.

Table 1. Example of a motif

Sequences		Aligned Sites			Alignment matrix					
S1 : atcATCCGTgtagctcaaaa		S1 : ATCCGT		Pos	1	2	3	4	5	6
S2 : agATCCGTaacgaagtttac		S2 : ATCCGT		A	5	0	0	0	0	0
S3 : ccccATCCGTaattacctat	⇒	S3 : ATCCGT	⇒	C	0	0	4	5	0	0
S4 : ggccgacttagccaatcga		S5 : ATCCGT		G	0	0	1	0	5	0
S5 : tATCCGTtagATACGTgccga		S5 : ATACGT		T	0	5	0	0	0	5

uses a genetic algorithm to move around the space of possible motifs in order to find an optimal motif under the same PWM-based Bayesian model used by BioOptimizer. GAME utilizes a large number of randomly generated starting points and does not require the use of any additional motif-finding algorithms. We demonstrate the superior performance of GAME in both simulation and real-data applications compared with popular programs MEME and BioProspector, as well as the optimization algorithm BioOptimizer.

2 METHODS

Genetic algorithms have greater freedom of movement between different configurations than simpler algorithms (Goldberg, 1989; Michalewicz, 1996), making them a valuable tool for the discovery of optimal motifs. Several studies (Stine *et al*., 2003; Liu *et al*., 2004) have used genetic algorithms for the identification of binding sites. In those analyses, the motif is not formulated as a PWM, but rather as a consensus sequence. Since we want to utilize the natural variability between binding sites within a motif, we prefer to formulate the unknown motif as PWM and employ the scoring function given in Jensen *et al*. (2004), which we review below. We then present our GAME (Genetic Algorithm for Motif Elicitation) software designed to find optimal motifs under this model. GAME uses the general genetic operators (Michalewicz, 1996) as well as two additional operators, SHIFT and ADJUST, which were included to help GAME avoid local optima.

2.1 Restricted solution space of possible motifs

Our dataset consists of m upstream sequences, each of length l_i, where S_{ij} is the nucleotide in position j of sequence i. A motif is represented by a matrix of binding site locations $\mathbf{A}$ where each $A_{ij} = 1$ if a motif site starts in position j of sequence i and 0 otherwise. Each configuration of $\mathbf{A}$ is a possible motif, such as the artificial example given in Table 1. The subsequences in upper case are true motif sites. The second motif site **ATACGT** is weaker than the first motif site **ATCCGT** (with one mutant) in sequence S5. The alignment matrix can be easily converted into a PWM when taking into account the distribution of bases in the background (non-motif) portions of each sequence (Hertz and Stormo, 1999). The set of all possible motifs of width w in our sequence dataset is $O\ (\prod_i 2^{l_i - w + 1})$, where l_i is the length of sequence i. This set of all possible motifs is prohibitively large for any optimization procedure, but the set of all reasonable motifs is much more restricted in the sense that we do not expect much more than one binding site in each sequence. Though it is possible that there is more than one binding site in a sequence for the same regulatory factor, these additional motif sites comprise a small portion of the whole set of binding sites. We build this prior expectation into our algorithm by initially restricting ourselves to discovering the strongest single site in each sequence, which we believe is the majority of all sites. Under this restriction, the matrix $\mathbf{A}$ can be reduced to a vector $A = (a_1, \ldots, a_m)$, where $a_i = 0$ indicates that there are no motif sites; otherwise a_i gives the location site of the motif site in sequence i. Note that we still allow any sequence to contain no sites, since there is usually a strong possibility of false positive sequences in any real data application. For the example in Table 1, our site vector A is (4, 3, 5, 0, 1). The set of all likely motifs is now approximately $\prod l_i$, which is much smaller than the original solution space. After we have a likely motif based on these strongest sites, we employ a simple scan procedure (Section 2.5) to identify additional weaker motif sites. We now descibe the scoring function we are trying to optimize with our discovered motifs.

2.2 Bayesian scoring function for motifs

The Bayesian approach presented in Jensen *et al*. (2004) models each potential site location A_{ij} as a random indicator variable with an a priori probability p_0 of equaling 1. This parameter p_0 is called the site abundance parameter. Each A_{ij} is assumed to be independent, allowing for the possibility that some sequences will have multiple motif sites (i.e. several $A_{ij} = 1$ in sequence i) as well as the possibility that some sequences may have no motif sites (i.e. all $A_{ij} = 0$ in sequence i). The composition of the motif is represented by the frequency matrix Θ, where θ_{jk} is the frequency of nucleotide k in column j of the motif. The nucleotide composition of the background (portions of the sequences that are not motif sites) is represented by the vector θ_0, where θ_{0k} is the frequency of nucleotide k in the background. This vector is treated as known since it can be usually estimated a priori. The posterior distribution of our unknown parameters can be written symbolically as follows:

$$p(\Theta, \mathbf{A} \mid \mathbf{S}, \theta_{\mathbf{0}}, p_0) \propto p(\mathbf{S} \mid \theta_{\mathbf{0}}, \Theta, \mathbf{A}) \times p(\mathbf{A} \mid p_0) \times p(\Theta) \times p(p_0) \qquad (1)$$

where our sequence data $\mathbf{S}$ and background frequencies θ_0 are known, and the site locations $\mathbf{A}$, motif composition Θ and site abundance p_0 is unknown. Details of these distributions are given in Jensen *et al*. (2004). This posterior mode is equivalent to the maximum-likelihood estimate when using non-informative prior distributions $p(\Theta)$ and $p(p_0)$. Since we are interested in comparing motifs based just on their site locations, $\mathbf{A}$, we can mathematically integrate over parameters Θ that specify the motif appearance, giving us the marginal posterior distribution for $\mathbf{A}$ alone:

$$p(\mathbf{A} \mid \mathbf{S}, \theta_{\mathbf{0}}, p_0) \propto \int p(\Theta, \mathbf{A} \mid \mathbf{S}, \theta_{\mathbf{0}}, p_0)\ \mathrm{d}\Theta \qquad (2)$$

An 'optimal' configuration of start sites $\mathbf{A}$ is defined as a maximum of the posterior distribution (2). Maximizing this posterior distribution (2) is equivalent to maximizing the log-posterior distribution.

$$\psi_{\text{post}}(\mathbf{A}) = \log p(\mathbf{A} \mid \mathbf{S}, \theta_{\mathbf{0}}, p_0) \qquad (3)$$

This log-posterior distribution $\psi_{\text{post}}(\mathbf{A})$ can be used as a scoring function which allows us to quantify the 'fitness' of different configurations of $\mathbf{A}$ in terms of their fit to the full probability model. As described in Jensen *et al*. (2004), the scoring function (3) is closely related to the following simpler scoring function:

$$\psi_{\text{ent}}(\mathbf{A}) = |\mathbf{A}| \cdot \left(\log\left(\frac{\hat{p}_0}{1 - \hat{p}_0} \right) - 1 + \prod_{j=1}^{w} \prod_{k=1}^{4} \hat{\theta}_{jk} \log\left(\frac{\hat{\theta}_{jk}}{\theta_{0k}} \right) \right) \qquad (4)$$

where $|\mathbf{A}|$ is the number of predicted sites and $\hat{p}_0 = |\mathbf{A}|/L$ is the estimated motif abundance out of $L = \sum_i l_i - w + 1$ possible site locations in $\mathbf{A}$. The term $\prod \prod \hat{\theta}_{jk} \log(\hat{\theta}_{jk}/\theta_{0k})$ is the relative entropy between the estimated motif matrix frequencies $\hat{\theta}_{jk}$ and background frequencies θ_{0k}. Note that a small number of prior counts β are usually added to each entry of the estimated motif matrix to ensure that all motif matrix frequencies $\hat{\theta}_{jk}$ are

Table 2. Examples of crossover and mutation operations

$A(a_1,\ldots,a_5)$		$B(b_1,\ldots,b_5)$	$A(a_1, a_2, a_3, a_4, a_5)$
GATTACA		GAGGACA	GATTACA
GATTAGA	↔	GAGGACA	GATTAGG
GATTACA		GAGGAGA	GATTACA
GAGGACA		GATTAGA	GATTACA
GAGGACA		GATTACA	GATTACA
⇓			⇓
Crossover Move			Mutation Move
⇓			⇓
GATTACA		GAGGACA	GATTACA
GATTAGA		GAGGACA	GATTACA
GATTACA		GAGGAGA	GATTACA
GATTAGA		GAGGACA	GATTACA
GATTACA		GAGGACA	GATTACA
$A(b_1, b_2, b_3, a_4, a_5)$		$B(a_1, a_2, a_3, b_4, b_5)$	$A(a_1, \mathbf{a}_2', a_3, a_4, a_5)$

non-zero. $\psi_{\text{ent}}(\mathbf{A})$ is used by our GAME genetic algorithm as the fitness function for finding the best configuration of site locations $\mathbf{A}$. In the following sections, we describe the various operations of our genetic algorithm.

2.3 Standard genetic operations

As mentioned in Section 2.1 above, each individual motif configuration was represented by a vector $A(a_1,\ldots,a_m)$ where $0 \le a_i \le (l_i - w + 1)$. We allowed simple changes of an individual configuration by using a standard point mutation operation: $A(a_1,\ldots,a_i,\ldots,a_m) \to A(a_1,\ldots,a_i',\ldots,a_m)$ with a certain mutation probability r. A standard one-point crossover move was also used that allows individual configurations to share and exchange alignment information with each other. De Jong (1975) suggests that the choice of a high crossover probability, a low mutation probability and a moderate population size improves performance of genetic algorithms. Based on this suggestion, we set the mutation probability $r = 0.001$ and the population size $N = 500$ in our GAME software, and we involve every member of the population in a crossover move in each generation. Our population of N individual configurations is randomly grouped into $N/2$ pairs of configurations. For each pair of individual configurations $A(a_1,\ldots,a_m)$ and $B(b_1,\ldots,b_m)$, a crossover point c is randomly generated that gives rise to two children configurations $A'(a_1,\ldots,a_c, b_{c+1},\ldots,b_m)$ and $B'(b_1,\ldots,b_c, a_{c+1},\ldots,a_m)$. The effects of our crossover and mutation moves are illustrated in Table 2. We retain both the parent configurations and children configurations from each crossover move, which increases our population size to $2 \cdot N$ since we used $N/2$ pairs of parents to produce $N/2$ pairs of children. In order to reduce the population size back down to N configurations, GAME uses a series of selection moves. Specifically, tournament selection (Goldberg *et al.*, 1991) with size = 2 was used: our $2 \cdot N$ individuals were randomly paired together and the one with a better fitness score $\psi_{\text{ent}}(\mathbf{A})$ was selected into the next generation. Thus, modifications that give better alignments between sites will tend to be favored by selection, and consequently well-aligned blocks of sites tend to spread through the population of site configurations $\mathbf{A}$. Through this evolutionary process, we expect that all individuals will converge to a single optimal configuration $\mathbf{A}_{\text{opt}}$. Our program has two separate stopping criteria for this evolutionary process: GAME will terminate if a user-defined maximum number of generations has been reached, or if there is no improvement in the best configuration $\mathbf{A}_{\text{opt}}$ after 50 consecutive iterations of the entire mutation/crossover process.

2.4 Additional genetic operations

Unfortunately, the *evolutionary process* described above is prone to premature convergence to local optima when used in this motif discovery setting.

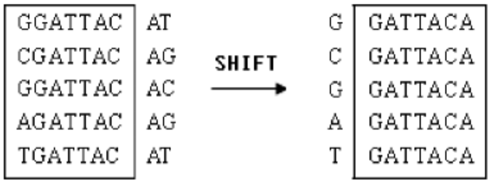

Fig. 1. SHIFT operation. The individual $A(a_1, a_2, a_3, a_4, a_5)$ changes to $A(a_1 + 1, a_2 + 1, a_3 + 1, a_4 + 1, a_5 + 1)$ by the SHIFT operation.

Table 3. ADJUST and SHIFT operators

```
ADJUST: $A(a_1,\ldots,a_i,\ldots,a_m)$
$i \leftarrow 1$
DO
    $a_i' = \text{argmax}\ \psi_{\text{ent}}(a_1,\ldots,a_i,\ldots,a_m),\ 1 \le a_i \le l_i - w + 1$
    $a_i \leftarrow a_i'$
    $i \leftarrow i + 1$
    if $i > m$ then $i \leftarrow 1$
UNTIL no further improvements obtained
SHIFT: $(A(a_1,\ldots,a_i,\ldots,a_m))$
$k' = \text{argmax}\ \psi_{\text{ent}}(a_1 + k,\ldots\ldots\ldots,a_m + k),\ -w \le k \le w$
$A \leftarrow A(a_1 + k' \ldots\ldots\ldots, a_m + k')$
Note 1: if $a_i = 0$ then $a_i + k = 0$ for all $k$ (no added sites)
Note 2: If $a_i + k < 0$ or $a_i + k > l_i - w + 1$, then set $a_i + k = 0$
```

We designed two additional operations, ADJUST and SHIFT, which are applied to our $\mathbf{A}_{\text{opt}}$ configuration in order to alleviate this problem. Viewing our motif discovery application as a problem of optimally aligning our motif sites, we encounter two types of local optima. The first type of local optima occurs when a majority of the motif sites have been aligned, with a few sites remaining to be aligned correctly. Even if only one motif site has not been aligned correctly, there are so many such local optima surrounding the true optimum that our standard genetic algorithm is easily trapped in one of them. We avoid this situation in our 'best' configuration $\mathbf{A}_{\text{opt}}$ by exhaustively checking every possible site position in a sequence and choosing the best match to the sites in the other sequences. As an example of this ADJUST move, consider the sequences given in Table 1. The fifth sequence in this table contains two sites, with the first site (ATCCGT) being a stronger match to the sites in the other sequences than the second site (ATACGT). If the current 'best' configuration $\mathbf{A}_{\text{opt}}$ contained the second site (ATACGT), our ADJUST move would remove that second site location from $\mathbf{A}_{\text{opt}}$ and replace it with the location of the first site (ATCCGT).

A second type of local optima occurs when all motif sites are slightly mis-aligned, as shown in the first part of Figure 1. This local optimum is nearly impossible to correct via crossover or mutation moves, since it requires simultaneous shifting of all sites in one direction. Instead, our SHIFT operator considers simultaneous moves in either direction of all sites in $\mathbf{A}_{\text{opt}}$, and again chooses the shift that gives the best fitness. Since these two additional operators are designed to improve near-optimal motif configurations, they are used only at the end of the evolutionary process in order to reduce their computational burden. The pseudo codes of ADJUST and SHIFT are shown in Table 3.

2.5 PWM-Scan

We used a simple scan procedure to extract additional motif sites within a set of sequences. Starting from our $\mathbf{A}_{\text{opt}}$ configuration, our scan algorithm cycles through all remaining potential motif sites, and selects any additional sites that give a superior fitness score. In other words, if we denote $\mathbf{A}'_{\text{opt}}$ as our configuration of sites with a new site added, then we accept this addition only if $\psi(\mathbf{A}'_{\text{opt}}) > \psi(\mathbf{A}'_{\text{opt}})$.

Table 4. Framework of GAME

```
BEGIN
  Initialization: $i \leftarrow 0$
    Setting parameters:
      population size $N = 500$
      mutation rate $r = 0.001$
      maximum generation $G = 3000$:
    Generating initial population $P_0$:
  Repeat: $i \leftarrow i + 1$
    Mutate individuals;
    Crossover individuals;
    Selection of individuals;
  Until ($i \geq G$ or $convergence$)
  Choose best individual $A_{\text{opt}}$;
  Repeat
    ADJUST($A_{\text{opt}}$);
    SHIFT($A_{\text{opt}}$);
  Until no further improvements obtained
  PWM-scan on $A_{opt}$ to extract additional weaker motif sites
END
```

Note: *convergence* has occurred when A_{opt} doesn't improve in 50 consecutive interactions.

2.6 Multiple motifs and the repeated sequences trap

Practitioners are often interested in finding more than one motif in a set of sequences. GAME addresses this situation with an iterative-masking approach: the binding sites of a discovered motif are masked out of the sequence dataset and then GAME is re-applied to this masked dataset to find additional motifs. A common occurrence in genomic sequences is the presence of repeated segments of DNA which are not transcription factor binding sites, but will be detected as motifs by *de novo* discovery programs. With our iterative-masking approach, these repeated segments will be discovered by GAME and then masked out of the sequence dataset, allowing GAME to discover additional motifs that may be of greater interest.

2.7 Extension to unknown motif width

The operations described above form the basic framework of our GAME program, as outlined in Table 4. We also have an extended version of our GAME program that allows the motif width to be unknown. In real applications, there is often very little known about the motif width w. Jensen *et al*. (2004) address this issue by considering the motif width w to also be a random variable with a prior distribution $p(w)$, such as a Poisson(w_0) with prior motif width w_0. This user-specified parameter w_0 allows the user to supply additional information to the analysis in terms of their prior expectation of the width for the unknown motif. This variable-width model has a more complicated scoring function:

$$\psi(\mathbf{A}, w) = \log p(w) + \log B(|\mathbf{A}|, L - |\mathbf{A}|) + \sum_{k=1}^{4} n_{0k} \log \theta_{0k} + \sum_{j=1}^{w} \log \left(\frac{\Gamma(4\beta)}{4 \cdot \Gamma(\beta)} \cdot \frac{\prod_k \Gamma(n_{jk} + \beta)}{\Gamma(|\mathbf{A}| + 4\beta)} \right) \quad (5)$$

where n_{jk} is count of nucleotide k in column j in the motif matrix and n_{0k} is the count of nucleotide k in the background. As before, β is a small number of prior counts added to each entry of the motif matrix to ensure non-zero motif matrix frequencies ($\beta = 1$ in our GAME program). $\Gamma(\cdot)$ denotes the gamma function, which is $\Gamma(x + 1) = x!$ for integer x, and $B(c, d)$ is the Beta function $\int_0^1 x^c (1 - x)^d dx$. Further details of this score function are given in Jensen *et al*. (2004) and Jensen and Liu (2004). Our variable-width version of GAME initially assumes the user-specified expected width w_0 is correct, and uses the fixed-width GAME program to find the best configuration $\mathbf{A}_{\text{opt}}$ for this particular width w_0. Following that optimization, the program then finds the optimal motif width w_{opt} by accepting the extension or reduction of the starting width which leads to the best score $\psi(\mathbf{A}, w)$. Obviously, if the expected motif width w_0 has the best score then no change is made.

3 RESULTS

3.1 Simulation evaluation of GAME

In order to evaluate the performance of GAME in motif site prediction, we designed the following set of simulations approximating different biological scenarios. A total of 200 sequence datasets were generated under each combination of several conditions:

(1) Number of sequences: small (20) or large (100).

(2) Width of motif: short (8 bp) or long (16 bp).

(3) Degree of conservation: high or low.

(4) Data scenario: noise-free or noisy.

High conservation means that each column of the true motif matrix had a dominant nucleotide with 91% probability (all others 3% equally). Low conservation means that each motif position had a dominant nucleotide with 70% probability (all others 10% equally). The 'noise-free' data scenario means that a single true motif site was placed in every sequence of the dataset. However, in reality, there are often some *false positive* sequences (which contain no sites) in a sequence dataset. We simulate this situation with our 'noisy' data scenario, where 10% sequences in each dataset contain no motif sites, representing the false positive part of the data. For the sequences containing at least one motif site, the number of motif sites follow the geometric distribution with $p = 0.9$, i.e. $P(n) = 0.1 \cdot (0.9)^{(n-1)}$. Further details of the construction of these simulated sequences is given in the supplemental materials.

For each simulated dataset, we applied the fixed-width version of GAME and two other popular motif-finding programs BioProspector and MEME. We also used the optimization program BioOptimizer, which needs to be used in conjunction with MEME or BioProspector. To compare the performance of each program, we used the standard information retrieval metrics of precision and recall (Shaw *et al*., 1997). In our case,

$$\text{Precision} = \frac{\#\ \text{of predicted motif sites that are true sites}}{\#\ \text{of predicted motif sites}}$$

$$\text{Recall} = \frac{\#\ \text{of predicted motif sites that are true sites}}{\#\ \text{of true sites}}$$

Note that shifting up to three base pairs was allowed for predicting correctly a true site. These two metrics were combined into the F score as $F = 2 * \text{precision} * \text{recall}/(\text{precision} + \text{recall})$, which is a standard method of comparison (Shaw *et al*., 1997). High values of F occur only when both precision and recall are high. The precision, the recall and F score were calculated for the best GAME, BioProspector, and MEME discovered motifs for each dataset, as well as the motifs that resulted from applying the optimization program BioOptimizer to the motifs discovered by BioProspector and MEME. The average precision, the recall and F score for each simulation condition (averaged over the 200 datasets within each condition) are shown in Table 5.

Table 5. Simulation results of GAME, MEME and BioProspector

Different conditions			Program	Noise-free scenario			Noisy scenario		
Number of sequences	Motif width	Conservation		Precision (%)	Recall (%)	*F*-score	Precision (%)	Recall (%)	*F*-score
Large	Long	High	GAME	100	99	99	99	98	99
			BioOptimizer based on MEME	99	99	99	99	99	99
			BioOptimizer based on BioProspector	99	99	99	99	99	99
			MEME	99	98	99	99	98	99
			BioProspector	99	91	95	100	92	96
Small	Long	High	GAME	100	99	99	99	97	98
			BioOptimizer based on MEME	96	99	98	96	99	97
			BioOptimizer based on BioProspector	96	99	98	96	99	97
			MEME	99	98	98	98	98	98
			BioProspector	100	93	96	100	93	96
Large	Short	High	GAME	92	84	88	92	83	87
			BioOptimizer based on MEME	92	84	88	92	85	88
			BioOptimizer based on BioProspector	92	84	88	92	85	88
			MEME	92	78	85	93	78	85
			BioProspector	90	72	80	90	72	79
Small	Short	High	GAME	90	86	88	86	82	84
			BioOptimizer based on MEME	88	85	86	87	85	86
			BioOptimizer based on BioProspector	88	85	86	87	85	86
			MEME	88	85	86	87	85	86
			BioProspector	85	81	83	85	81	83
Large	Long	Low	GAME	88	68	77	88	65	75
			BioOptimizer based on MEME	92	58	71	92	58	71
			BioOptimizer based on BioProspector	91	58	71	91	58	71
			MEME	93	55	69	93	54	68
			BioProspector	90	43	58	89	44	58
Small	Long	Low	GAME	70	60	65	66	55	60
			BioOptimizer based on MEME	78	50	61	77	55	63
			BioOptimizer based on BioProspector	79	43	55	76	53	62
			MEME	78	50	61	78	50	60
			BioProspector	79	43	55	78	42	54

In the noise-free scenario, Table 5 shows that GAME performs better, in terms of F score, than MEME, BioProspector and BioOptimizer in each condition. The advantage of GAME becomes more predominant as conditions become more difficult (e.g. lower motif conservation). In these difficult cases, GAME actually shows lower precision but much higher recall. In the noisy scenario, GAME generally shows superior performance (in terms of F score) over MEME, BioProspector and BioOptimizer. GAME has better recall than MEME and BioProspector, and has comparable precision except in the last difficult case ('Small-Long-Low'), where GAME is slightly out-performed by MEME and BioOptimizer in terms of F score. Note that the two most difficult condition combinations ('Small-Short-Low' and 'Large-Short-Low') were not included because the performance of all programs was very poor. Overall, this simulation study suggests that the optimization provided by GAME is generally superior to the optimization provided by BioOptimizer, with the additional advantage that our GAME algorithm does not require the use of another motif discovery program.

We also used these simulation datasets to evaluate the contribution of our additional ADJUST and SHIFT operations (Section 2.4). For each simulated dataset, we tabulated the total number of times that the ADJUST or SHIFT operations were used to improve the score of the best configuration for that dataset. Figure 2 shows the distribution of the total number of needed ADJUST or SHIFT operations across the 200 simulated datasets within each simulation condition. In almost all cases, the total number of ADJUST and SHIFT operations is non-zero, which indicates that our additional genetic operations are useful for improving the best configuration $\mathbf{A}_{\mathrm{opt}}$ in almost all datasets.

3.2 Real-data applications

The cyclic AMP receptor protein (CRP) functions as a transcription factor in *Escherichia coli*. We analyzed 18 sequences, each 105 bp long, which contain 23 sites that have been experimentally determined. This dataset has been previously analyzed by Stormo and Hartzell (1989), Lawrence and Reilly (1990) and Liu (1994). The estrogen receptor (ER) is a ligand-activated enhancer protein which binds to specific DNA sequences called estrogen response elements (EREs) with high affinity and activates gene expression in response to estradiol. We analyzed 25 genomic sequences, each of which is 200 bp long and contains a single known ERE (Klinge, 2001). Finally, we examined the regulation of 25 mammalian sequences of 200 bp width which contained 27 known binding sites for transcription factors in the E2F family (Kel *et at*., 2001; Berman *et al*., 2002; Frith *et al*., 2004). In previous analyses of these

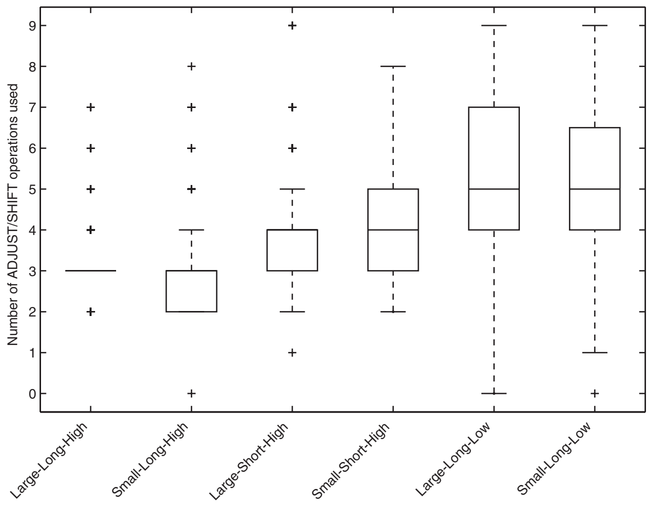

Fig. 2. Distributions of the total number of ADJUST and SHIFT operations needed by GAME in different simulation conditions.

datasets, researchers used motif widths of 22 bp for CRP, 13 bp for ERE and 11 bp of E2F, which we used as the expected motif widths w_0 in these three applications. We also selected five additional datasets for the transcription factors CREB, MEF2, MYOD, SRF and TBP from the recently-published ABS database of annotated regulatory binding sites (Blanco *et al*., 2006). Based on their published patterns, we used expected motif widths w_0 of 8, 7, 6, 10 and 6 for CREB, MEF2, MYOD, SRF and TBP respectively. For each application, we used the variable-width version of GAME to find the optimal width within a range of 3 bps on either side of the expected motif width w_0.

In Table 6, we compare the prediction results of our variable-width GAME to BioOptimizer, MEME and BioProspector. BioProspector doesn't allow the motif width to vary, so we used the expected widths w_0 directly in BioProspector. We see that GAME gives superior recall and comparable precision when compared to BioOptimizer, MEME and BioProspector, which results in a better F score for GAME in each application. BioProspector, MEME and GAME took an average of 2, 33 and 233 seconds to run these applications, respectively, so in terms of computational speed, GAME is slower than its competitors MEME and BioProspector. However, the greater computational cost for GAME is not substantial (several minutes versus less than a minute for moderately large datasets) and is far outweighed by the considerable improvement in performance of GAME compared with its competitors.

In Figure 3, we use the software WebLogo (Crooks *et al*., 2004) to display the sequence logos (Schneider and Stephens, 1990) for GAME's 'predicted' motifs of CRP, ERE and E2F, each of which is very consistent with the 'true' motif based on the experimentally determined sites and their surrounding genomic sequence.

Table 6. Real-data application results

TF	Predictor	w	$\|A\|$	Precision	Recall	F-score
CREB	GAME	8	22	15/22	15/19	0.73
	BioOpt. based on MEME	12	15	10/15	10/19	0.59
	BioOpt. based on BioPro.	9	17	12/17	12/19	0.67
	MEME	11	15	10/15	10/19	0.59
	BioProspector	8	20	13/20	13/19	0.67
CRP	GAME	19	17	16/17	16/23	0.80
	BioOpt. based on MEME	24	13	12/13	12/23	0.67
	BioOpt. based on BioPro.	24	13	12/13	12/23	0.67
	MEME	24	13	12/13	12/23	0.67
	BioProspector	22	9	9/9	9/23	0.56
ERE	GAME	13	26	19/26	19/25	0.75
	BioOpt. based on MEME	15	22	17/22	17/25	0.72
	BioOpt. based on BioPro.	16	23	18/23	18/25	0.75
	MEME	15	17	15/17	15/25	0.71
	BioProspector	13	16	14/16	14/25	0.68
E2F	GAME	11	24	23/24	23/27	0.90
	BioOpt. based on MEME	13	27	20/27	20/27	0.74
	BioOpt. based on BioPro.	13	27	19/27	19/27	0.70
	MEME	13	23	19/23	19/27	0.76
	BioProspector	11	21	11/21	11/27	0.46
MEF2	GAME	9	17	15/17	15/17	0.88
	BioOpt. based on MEME	13	15	14/15	14/17	0.88
	BioOpt. based on BioPro.	11	19	11/19	11/17	0.61
	MEME	9	15	14/15	14/17	0.88
	BioProspector	7	17	12/17	12/17	0.71
MYOD	GAME	7	21	10/21	10/21	0.48
	BioOpt. based on MEME	10	10	0/10	0/21	0.00
	BioOpt. based on BioPro.	11	11	0/11	0/21	0.00
	MEME	9	8	0/8	0/21	0.00
	BioProspector	6	18	0/18	0/21	0.00
SRF	GAME	10	47	33/47	33/36	0.80
	BioOpt. based on MEME	14	51	32/51	32/36	0.74
	BioOpt. based on BioPro.	14	50	32/50	32/36	0.74
	MEME	13	48	28/48	28/36	0.67
	BioProspector	10	35	25/35	25/36	0.70
TBP	GAME	7	91	78/91	78/95	0.84
	BioOpt. based on MEME	12	79	35/79	35/95	0.40
	BioOpt. based on BioPro.	9	78	65/78	65/95	0.75
	MEME	12	50	26/50	26/95	0.36
	BioProspector	6	69	58/69	58/95	0.71
Average	GAME			0.78	0.77	0.77
	BioOpt. based on MEME			0.64	0.57	0.59
	BioOpt. based on BioPro.			0.65	0.60	0.61
	MEME			0.67	0.53	0.58
	BioProspector			0.66	0.51	0.56

Note: $|A|$ is the number of predicted motif sites.

4 DISCUSSION

We have introduced genetic algorithms as a general optimization strategy, which we implemented in our software GAME. GAME starts from a population of randomly generated motifs and performs an extensive search through operations such as crossover and mutation for optimal motifs. In both simulation studies and real-data applications, GAME showed superior overall performance to MEME and BioProspector, two popular motif-finding programs. Most of the improvement from GAME comes from increased levels of recall, while maintaining comparable precision. Compared to BioOptimizer, GAME also shows superior performance in both simulation and real-data analyses, and also eliminates the dependence of BioOptimizer on other motif-finding programs. Although each of the compared motif finding programs is based on a similar

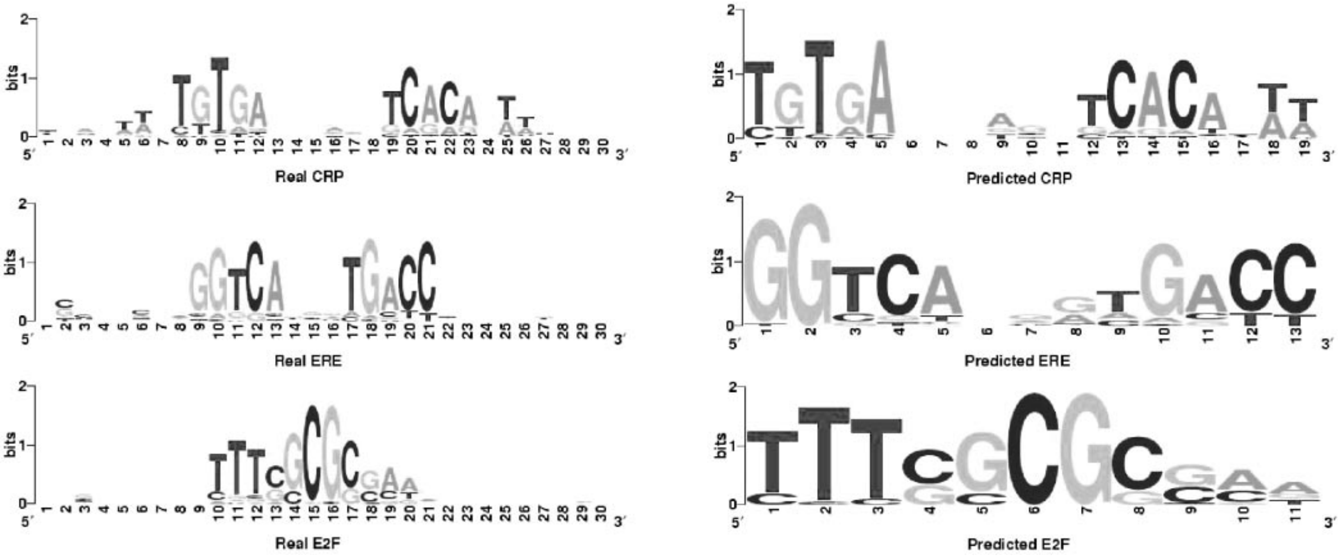

Fig. 3. Sequence logos (Schneider and Stephens, 1990) for the experimentally confirmed binding sites of CRP, ERE and E2F (left) compared with the binding sites predicted by GAME (right), created using the software WebLogo (Crooks *et al*., 2004).

statistical model, as outlined in Jensen *et al*. (2004), there are still considerable differences in results from these different programs due to the existence of a large number of possible solutions. Optimization algorithms such as BioOptimizer or MEME can locally optimize their motif discovery results, but the inherent multimodality of the solution space restricts these local optimization procedures from exploring many different solutions. Our genetic algorithm framework allows a greater flexibility of movement around the solution space by applying an evolutionary process to an entire population of possible solutions. We also present an extended version of GAME that also attempts to find the optimal motif width in situations where the motif width is unknown. Despite considerable effort to date, it remains a complex challenge for computational biologists to convincingly predict regulatory elements in DNA sequences. Current motif discovery models are a rather simplistic approximation of biological reality, though more recent efforts have attempted to include correlation between motif positions (Zhou and Liu, 2004) and synergistic relationships between transcription factors (Gupta and Liu, 2005). As the complexity of these models increases, the need for sophisticated algorithms for finding optimal solutions to these models will become increasingly important.

ACKNOWLEDGEMENTS

We are grateful to Zhenyu Yan for his advice on designing genetic algorithm operators. This research was supported by a grant to S.T.J. from the University of Pennsylvania Research Foundation.

Conflict of Interest: none declared.

REFERENCES

Bailey,T.L. and Elkan,C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In: *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*. AAAI Press, Menlo Park, CA, pp. 28–36.

Berman,B.P. *et al*. (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proc. Natl Acad. Sci. USA*, **99**, 757–762.

Blanco,E. *et al*. (2006) ABS: a database of annotated regulatory binding sites from orthologous promoters. *Nucleic Acids Res*., **34**, D63–D67.

Crooks,G.E. *et al*. (2004) WebLogo: A sequence logo generator. *Genome Res*., **14**, 1188–1190.

De Jong,K.A. (1975) An analysis of the behavior of a class of genetic adaptive systems. Ph.D. Thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI.

De Jong,K.A. and Spears,W.M. (1989) Using genetic algorithms to solve NP-complete problems. In: *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, San Francisco, CA, pp. 124–132.

Frith,M.C. *et al*. (2004) Finding functional sequence elements by multiple local alignment. *Nucleic Acids Res*., **32**, 189–200.

Goldberg,D.E (ed.) (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Wesley, New York.

Goldberg,D.E. *et al*. (1991) Do not worry, be messy. In: *Proceeding of the Fourth International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers, Altos, CA, pp. 24–30.

Gupta,M. and Liu,J.S. (2005) De novo cis-regulatory module elicitation for eukaryotic genomes. *PNAS*, **102**, 7079–7084.

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

Jensen,S.T. *et al*. (2004) Computational discovery of gene regulatory binding motifs: a Bayesian perspective. *Stat. Sci*., **19**, 188–204.

Jensen,S.T. and Liu,J.S. (2004) BioOptimizer: a Bayesian scoring function approach to motif discovery. *Bioinformatics*, **20**, 1557–1564.

Jensen,S.T. *et al*. (2005) Combining phylogenetic motif discovery and motif clustering to predict co-regulated genes. *Bioinformatics*, **21**, 3832–3839.

Kel,A.E. *et al*. (2001) Computer-assisted identification of cell cycle-related genes: new targets for E2F transcription factors. *J. Mol. Biol*., **309**, 99–120.

Klinge,C.M. (2001) Estrogen receptor interaction with estrogen response elements. *Nucleic Acids Res*., **29**, 2905–2919.

Kullback,S. and Leibler,R.A. (1951) On information and sufficiency. *Ann. Math. Stat*., **22**, 78–86.

Lawrence,C.E. and Reilly,A.A. (1990) An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, **7**, 41–51.

Levine,M. and Tjian,R. (2003) Transcription regulation and animal diversity. *Nature*, **424**, 147–151.

Liu, Falcon,F.M. *et al*. (2004) FMGA: Finding motifs by Genetic algorithm. *Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04)*, bibe 459.

Liu,J.S. (1994) The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *J. Am. Stat. Assoc*., **94**, 958–966.

Liu,J.S. *et al*. (1995) Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *J. Am. Stat. Assoc*., **90**, 1156–1170.

Liu,X. *et al*. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput*., **6**, 127–138.

Liu,X.S. *et al*. (2002) An algorithm for finding protein-DNA interaction sites with applications to chromatin immunoprecipitation microarray experiments. *Nat. Biotechnol*., **20**, 835–839.

McCue,L.A. *et al*. (2001) Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Res*., **29**, 774–782.

Michalewicz,Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd ed. Springer-Verlag, Berlin.

Neuwald,A.F. *et al*. (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Sci*., **4**, 1618–1632.

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

Schneider,T.D. and Stephens,R.M. (1990) Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*., **18**, 6097–6100.

Stine,M. *et al*. (2003) Motif discovery in upstream sequences of coordinately expressed genes. *Evol. Comput., CEC '03*, **3**, 11596–1603.

Stormo,G.D. and Hartzell,G.W. (1989) Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl Acad. Sci*., USA**86**, 1183–1187.

Shaw,W.M.,Jr *et al*. (1997) Performance standards and evaluations in IR test collections: cluster-based retrieval models. *Inf. Process. Manage*., **33**, 114.

Zhou,Q. and Liu,J.S. (2004) Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics*, **6**, 909–916.