BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 3 2010, pages 302–309
doi:10.1093/bioinformatics/btp676

Sequence analysis

HIGEDA: a hierarchical gene-set genetics based algorithm for finding subtle motifs in biological sequences

Thanh Le[1], Tom Altman[1] and Katheleen Gardiner[2,*]

[1]Department of Computer Science and Engineering, [2]Department of Pediatrics, Computational Biosciences, Human Medical Genetics and Neuroscience Programs, University of Colorado, Denver, CO, USA

Received on June 30, 2009; revised on December 2, 2009; accepted on December 3, 2009

Advance Access publication December 8, 2009

Associate Editor: Limsoon Wong

ABSTRACT

Motivation: Identification of motifs in biological sequences is a challenging problem because such motifs are often short, degenerate, and may contain gaps. Most algorithms that have been developed for motif-finding use the expectation-maximization (EM) algorithm iteratively. Although EM algorithms can converge quickly, they depend strongly on initialization parameters and can converge to local sub-optimal solutions. In addition, they cannot generate gapped motifs. The effectiveness of EM algorithms in motif finding can be improved by incorporating methods that choose different sets of initial parameters to enable escape from local optima, and that allow gapped alignments within motif models.

Results: We have developed HIGEDA, an algorithm that uses the hierarchical gene-set genetic algorithm (HGA) with EM to initiate and search for the best parameters for the motif model. In addition, HIGEDA can identify gapped motifs using a position weight matrix and dynamic programming to generate an optimal gapped alignment of the motif model with sequences from the dataset. We show that HIGEDA outperforms MEME and other motif-finding algorithms on both DNA and protein sequences.

Availability and implementation: Source code and test datasets are available for download at http://ouray.cudenver.edu/~tnle/, implemented in C++ and supported on Linux and MS Windows.

Contact: katheleen.gardiner@ucdenver.edu

1 INTRODUCTION

Finding motifs in biological sequences is a challenging but important problem in computational biology. Motifs in DNA sequences can indicate potential binding sites for proteins regulating gene transcription. Motifs in protein sequences allow inferences regarding function. In both DNA and protein sequences, motifs are difficult to recognize because they are short, often highly degenerate, and may contain gaps. Motif-finding is not a new problem, nevertheless, challenges remain because no single algorithm both describes motifs and finds them effectively. Popular methods for motif-finding use a position specific score matrix (PSSM) that describes motifs statistically, or consensus strings representing a motif by one or more patterns that appear repeatedly with a limited number of differences. Of the two, PSSM is preferred because it is more informative and can be easily evaluated using statistical methods. In addition, a consensus motif model can be replaced by a PSSM one.

One of the most popular motif-finding algorithms using PSSM is MEME (Bailey and Elkan, 1995). The advantage of MEME is that it uses expectation-maximization (EM) which, as a probability-based algorithm, produces statistically significant results if it can reach a global optimum. The disadvantage is that for motif seeds it uses existing subsequences from within the sequence set and, as a result, may fail to discover subtle motifs. Chang *et al*. (2006) and Li *et al*. (2007) overcame this drawback by using the genetic algorithm (GA) to generate a set of motif seeds randomly. However, because they use GA with random evolution processes, a rapid convergence to a solution is not assured. Li *et al*. (2008) improved on MEME by using one instance of a position weight matrix (PWM), a type of PSSM, to represent a motif and statistical tests to evaluate the final model. However, because EM may converge to local optima, use of a single PWM may fail to find a globally optimal solution. The GA-based methods of Wei and Jensen (2006) and Bi (2007) use chromosomes to encode motif positions. The method in the former is appropriate for models with zero to one occurrence of the motif per sequence (ZOOPS); the latter, for models with one occurrence per sequence (OOPS), because one variable can be used to represent the single motif occurrence in each sequence. However, Li *et al*. (2008) recently showed that ZOOPS and OOPS are inadequate when not every sequence has the same motif frequency, and that the two-component mixture (TCM) model, which assumes a sequence may have zero or multiple motif occurrences, should be used. However, TCM requires a set of variables for every sequence to manage motif positions, and hence, the size of a chromosome can approach the size of the dataset. Lastly, the above algorithms are restricted to finding gapless motifs and, therefore, will fail to find many functionally important, gapped motifs. While some methods, e.g. pattern-based methods of Pisanti *et al*. (2005) and Frith (2008), allow gapped motifs, they require the gapped patterns to be well-defined and they generate gap positions randomly or using a heuristic method. Alternatively, Liu *et al*. (2006) used neural networks to find gapped motifs, but their approach required limited and specific definition of the neural network structure.

In this study, we propose a new algorithm, HIGEDA, applicable to either DNA or protein sequences, which uses the TCM model and combines the hierarchical gene-set genetic algorithm (HGA) (Hong and Wu, 2008) with EM and dynamic programming (DP) algorithms to find motifs with gaps. HGA helps HIGEDA manage motif seeds

*To whom correspondence should be addressed.

and escape from local optima; EM uses the best alignment of a motif model on the dataset, where the alignments are generated by DP to make the model fit the best conserved forms of motifs of interest.

2 METHODS

2.1 Motif finding problem

Given $A=\{\underline{a}_1,a_2,\ldots,a_l\}\cup\{\_\}$, a set of symbols used for sequence encoding and the gap symbol, $l=20$ for protein sequences and $l=4$ for DNA sequences. Let $S=\{S_1,S_2,\ldots,S_n\}$, be a set of biological sequences based on A. Assume $L=\{L_1,L_2,\ldots,L_n\}$, where L_i is the length of sequence i, $i=1\ldots n$. Assume that the length of shared motifs is a known value, W.

With $m_i=L_i-W+1$, the number of possible positions of a motif in sequence i, denote $Z=\{z_{ij}\}, i=1\ldots n, j=1\ldots m_i$, as the probability of motif occurrence at position j in sequence i.

Let Θ be the motif model.

The motif finding problem is to determine Θ and Z such that:

$$P(S,Z|\Theta)\rightarrow\max. \tag{1}$$

2.2 Motif model

Of the motif model Θ, two components Θ^M and Θ^B model the motif and non-motif (the background) positions in sequences. A motif is modeled by a sequence of discrete random variables whose values give the probabilities of each of the different letters occurring in each of the different positions of a motif occurrence. The background positions in the sequences are modeled by a single discrete random variable. The motif model Θ is as follows:

$$\Theta=\{\Theta^B,\Theta^M\}=\begin{bmatrix}\theta^B_{\_,0} & \theta^M_{\_,1} & \theta^M_{\_,2} & \ldots & \theta^M_{\_,w}\\ \theta^B_{a1,0} & \theta^M_{a1,1} & \theta^M_{a1,2} & \ldots & \theta^M_{a1,w}\\ \theta^B_{a2,0} & \theta^M_{a2,1} & \theta^M_{a2,2} & \ldots & \theta^M_{a2,w}\\ \ldots & \ldots & \ldots & \ldots & \ldots\\ \theta^B_{al,0} & \theta^M_{al,1} & \theta^M_{al,2} & \ldots & \theta^M_{al,w}\end{bmatrix}, \tag{2}$$

where, $\Theta_{a,k}$, $1\leqslant k\leqslant W$, is the probability that symbol a occurs either at a background position or at position k of a motif occurrence.

$$\sum_{a\in A}\theta_{a,k}=1,\ \forall k=1\ldots W \tag{3}$$

The Θ matrix in Equation (2) is a PWM but its use is different from conventional PWMs in that it contains 21 rows for protein motifs or five rows for DNA motifs. The first row stands for the gap symbol, which may occur in a motif, but not in the first or last positions. The remaining rows stand for residue symbols, 20 amino acid letters or four nucleotide letters. W is chosen large enough to accommodate all possible consensuses. Figure 1 shows a motif model Θ representing 'AC', 'AGC' and 'ATC'.

Sequence and motif model alignment with gaps Given an input subsequence s, the best alignment with gaps of s and Θ is created using DP algorithm. Since s is from a real dataset, gaps are allowed only in Θ. The first symbol of s is aligned with the first column in Θ. Consecutive symbols from s are aligned with either gap or symbol columns to achieve the best alignment score. A conventional dynamic alignment score is the sum of the pair wise alignment scores. Instead here, it is the multiplication of all pair wise alignment scores, and hence it is the probability that the best consensus from Θ matches s. To control the occurrence of gaps, we define PPM, POG and PEG as the reward for a perfect match and the penalties for opening and extending a gap, respectively. We choose, by empirical experiments, PPM = 1, POG = 0.00875 and PEG = 0.325. Let U_{jk} be the best alignment score up to the j-th symbol s_j in s and column k of Θ. U_{jk} is calculated as follows:

	0	1	2	3
-	0.00	0.00	0.25	0.00
A	0.02	0.50	0.10	0.04
G	0.02	0.30	0.35	0.02
C	0.03	0.10	0.01	0.90
T	0.03	0.10	0.29	0.04

Fig. 1. A motif model Θ.

	1	2	3
A	0.50	0.0043	-
C	-	0.0050	0.0039
G	-	-	0.0001

Fig. 2. Dynamic alignment of s ='ACG' w.r.t Θ from Figure 1.

$$U_{jk}=\begin{cases}\theta^M_{s_j,1}, & k=j=0\\ U_{j-1,k-1}*PPM*\theta^M_{s_jk+1}, & k=j\\ U_{j,k-1}*PG*(1-\theta^M_{s_{j+1},k+1}), & j=0,\ k>j\\ \max\{U_{j-1,k-1}*PPM*\theta^M_{s_jk+1}, U_{j,k-1}*PG*(1-\theta^M_{s_{j+1},k+1})\}\end{cases} \tag{4}$$

where PG is either PEG or POG depending on the gap status at column k in U. Let s='ACG' and Θ be as shown in Figure 1. Figure 2 shows the dynamic alignment matrix $\{U_{ij}\}$ of s w.r.t Θ. The best alignment score is 0.0039 which is found in the second row and the last column of the matrix. Hence, the best consensus for s='ACG' by Θ is 'A_C'. 'AC' is considered in-motif model and the last symbol 'G' is out-motif model for subsequence s. The motif model will be adjusted to fit 'A_C' instead of 'ACG'. This feature produces the occurrence probabilities for gap symbols in our motif model even if such a symbol does not appear in the sequences set.

Motif occurrence Let S^{ij} be the subsequence of length W at position j in sequence i. Denote $I(.)$ as the indicator function and $P_M(S^{ij})$ and $P_B(S^{ij})$ as the conditional probabilities that S^{ij} is generated using the motif and the background models, respectively.

$$P_M(S^{ij})=P(S^{ij}|\Theta^M)=\prod_{k=1}^{W}\prod_{a=1}^{L}\left(\theta^M_{ak}\right)^{I(S_{i,j+k-1}=a)}, \tag{5}$$

$$P_B(S^{ij})=P_B(S^{ij}|\Theta^B)=\prod_{k=1}^{W}\prod_{a=1}^{L}\left(\theta^B_{a0}\right)^{I(S_{i,j+k-1}=a)}. \tag{6}$$

Let λ be the prior probability of motif occurrence at every possible position in the sequence set. Similar to use of the TCM model in MEME (Bailey and Elkan, 1995), the motif occurrence probability at position j in sequence i is

$$z_{ij}=\frac{\lambda*P_M(S^{ij})}{\lambda*P_M(S^{ij})+(1-\lambda)*P_B(S^{ij})}. \tag{7}$$

The motif log-odds score of S^{ij} is defined by

$$los(S^{ij}) = \log\left(P_M(S^{ij}) \Big/ P_B(S^{ij})\right). \quad (8)$$

S^{ij} is considered a motif hit if it satisfies the inequality:

$$los(S^{ij}) \geqslant \log[(1-\lambda)/\lambda]. \quad (9)$$

Gapped motif When gaps are allowed in the alignment of S^{ij} and Θ, $P_M(S^{ij})$ and $P_B(S^{ij})$ are calculated using the recovered version with gaps of S^{ij}, similar to that in Xie (2004). Let s_m and s_c be the in-motif and out-motif parts, respectively. Let s* be the best consensus for S^{ij} by Θ.

$$P_M(S^{ij}) = P_B(s_c) * P_M(s^*) = P(s_c|Z,\Theta^B) * P(s^*|Z,\Theta^M), \quad (10)$$

$$P_B(S^{ij}) = P_B(s_c) * P_B(s_m) = P(s_c|Z,\Theta^B) * P(s_m|Z,\Theta^B). \quad (11)$$

Multiple-motif finding To find multiple motifs in given dataset, we use a mask variable M, $M = \{M_{ij}\}_{i=1,\ldots,n,\ j=1,\ldots,m}$, where M_{ij} represents the probability of the chance that a motif occurs again at position j in sequence i. M is initially set to $\{1\}$. The modified version of Equation (7) with respect to M is

$$z_{ij}^M = P(z_{ij}=1|M) = z_{ij} * \min_{k=1\ldots W}\{M_{i,j+k-1}\}. \quad (12)$$

Once a motif is found, its positions are updated to M.

$$M_{i(j+k)} = M_{i(j+k)} * (1 - \Delta M_{i(j+k)}), \quad (13)$$

where, $\Delta M_{i(j+k)} = P(z_{ij}=1) \times (W-k)/W$, $k = 0,\ldots,W-1$. The update mechanism in Equation (13) allows for multiple overlapping motifs.

2.3 HGA

The GA is a global optimization procedure that performs adaptive searches to find solutions to large-scale optimization problems with multiple local optima. Conventional GAs use crossover and mutation operators to escape local optima. These operators depend strongly on how the probabilities of crossover and mutation are chosen. Recent improvements in GA have focused on adaptively adjusting operator probabilities so that the genetics processes quickly escape local optima. However, setting up adaptive GAs is difficult and most approaches are based on heuristics.

Here we use HGA, a GA improved by Hong and Wu (2008). HGA treats a chromosome as a set of gene-sets, not a set of genes as in conventional GAs, as a mechanism to escape local optima. Starting with gene-sets of the largest size equal to half the chromosome length and ending with gene sets of size 1, HGA performs crossover and mutation operations based on gene-set boundaries. When the model is (δ,k)-convergent, HGA expects to find a global optimum and attempts to escape local optima, if any exist, by performing genetics operations with the largest size gene-sets. HGA is most appropriate to our genetics model because each gene-set represents a set of adjacent columns in the motif model that represents patterns of adjacent residues in biological sequences. Genetics operations based on gene-set boundaries allow residues to come together as well as to change simultaneously.

3 ALGORITHMS

3.1 EM algorithm used in HIGEDA

HIGEDA solves the motif finding problem using an EM algorithm to maximize the likelihood function [Equation (14)] over the entire dataset.

$$L(S,Z|\Theta) = \log P(S,Z|\Theta) \to \max \quad (14)$$

Estimation step: Z is estimated using Equation (7) [or Equation (12)].

$$z_{ij}^{(t+1)} = \frac{\lambda^t * P_M(S^{ij})}{\lambda^t * P_M(S^{ij}) + (1-\lambda^t) * P_B(S^{ij})}. \quad (15)$$

Maximization step: Θ and λ are computed with respect to Equation (15),

$$\begin{aligned} L(S,Z|\Theta) = &\sum_{i=1}^{n}\sum_{j=1}^{m}(1-z_{ij})\log P_B(S^{ij}) + z_{ij}\log P_M(S^{ij}) \\ &+ (1-z_{ij})\log(1-\lambda) + z_{ij}\log(\lambda) \to \max \end{aligned} \quad (16)$$

Such that,

$$\frac{\partial L}{\partial \lambda} = 0 \ \to\ \lambda = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}. \quad (17)$$

With respect to constrains [Equation (3)], the objective function is relaxed using Lagrange multipliers $\gamma_k, k = 0,\ldots,W$,

$$L = L(S,Z|\Theta) + \sum_{k=0}^{W}\gamma_k\left(\sum_{a=1}^{l}\theta_{ak} - 1\right),$$

$$\begin{aligned} \frac{\partial \log P_M(S^{ij})}{\partial \theta_{ak}^M} &= \frac{1}{P_M(S^{ij})}\frac{\partial P_M(S^{ij})}{\partial \theta_{ak}^M} \\ &= \frac{1}{P_M(S^{ij})}\left(\prod_{\substack{k'=1 \\ k'\neq k}}^{W}\prod_{a'=1}^{L}\left(\theta_{ak'}^M\right)^{I(S_{i,j+k'-1}=a')}\right) I(S_{i,j+k-1}=a) \cdot \\ &= \frac{I(S_{i,j+k-1}=a)}{\theta_{ak}^M} \end{aligned}$$

Hence,

$$\frac{\partial L}{\partial \theta_{ak}^M} = \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}\frac{I(S_{i,j+k-1}=a)}{\theta_{ak}^M} + \gamma_k = 0.$$

So,

$$\theta_{ak}^M = -\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} I(S_{i,j+k-1}=a)}{\gamma_k}. \quad (18)$$

$$\begin{aligned} (3)\ \&\ (18) \to \sum_{a=1}^{l}\theta_{ak}^M &= -\frac{\sum_{a=1}^{l}\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} I(S_{i,j+k-1}=a)}{\gamma_k} \\ &= -\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}\sum_{a=1}^{l} I(S_{i,j+k-1}=a)}{\gamma_k} = -\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}}{\gamma_k} = 1 \end{aligned}, \quad (19)$$

$$(18)\ \&\ (19) \to \theta_{ak}^M = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij} I(S_{i,j+k-1}=a)}{\sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}}. \quad (20)$$

Similarly,

$$\theta_{a0}^{B}=\frac{1}{W}\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(1-z_{ij})\sum_{k=1}^{W}I(S_{i,j+k-1}=a)}{\sum_{i=1}^{n}\sum_{j=1}^{m}(1-z_{ij})}. \quad (21)$$

3.2 Refining model parameters

The maximization step results in model parameters Θ and λ. Because the values of Z are estimated during EM processes, a straightforward update to Θ and λ using Equations (17), (20) and (21) may cause an oscillation in convergence. MEME estimates λ by trying values from $1/\left(m\sqrt{n}\right)$ to $1/(W+1)$. This requires significant computational time and squanders the maximization step benefit. We apply the gradient descent learning law to update model parameter, ξ, of the model,

$$\xi(t+1)=\xi(t)+\mu_t*\Delta\xi, \quad (22)$$

where $\Delta\xi=\xi_e(t+1)-\xi(t)$ and $\xi(t),\xi_e(t+1)$ and $\xi(t+1)$ are the current, estimated and new values of ξ, respectively. The learning rate μ_t may be slightly reduced by processing time. A popular form of μ_t is as in Equation (23) where T is the processing duration.

$$\mu_t=\mu_{\max}\left(1-\frac{t}{1.1*T}\right). \quad (23)$$

If the size of the dataset is small, some elements of Θ may be zero. These are not healthful to the Bayesian process and remain zero. To solve this problem, MEME uses Dirichlet mixture model. While this has a strong mathematical basis, the drawback lies in how to select the right number of mixture components. In addition, this number has no meaning in sequence analysis. Instead, we use pseudo-count methods. For DNA, we borrow an added pseudo-counts method from Henikoff and Henikoff (1996). The added portion used is 0.01. For proteins, we propose a method using the motif model and the substitution probability matrix from BLOSUM62, and based on a heuristic that the pseudo-counts should be position specific and depend on strong signals. The pseudo-count value for a given symbol a, in column k of Θ is calculated using Equation (24).

$$psc_{ak}=\sum_{b=1}^{L}\theta_{b,k+1}^{M}P_{a/b}, \quad (24)$$

where $P_{a/b}$ is the BLOSUM substitution probability for amino acid a from the observation of amino acid b. The motif model Θ is then refined using Equation (25) in which α, β are predefined and $\alpha+\beta=1$.

$$\Theta r_{a,k}=\alpha.\Theta_{a,k}+\beta.psc_{ak}. \quad (25)$$

3.3 HIGEDA algorithm

HIGEDA uses HGA to manage its population and to escape local optima. During the evolution process, members of the current generation are processed with the EM algorithm a small number of times, and ranked based on their goodness measured using a fitness function. The best members are used to create new ones using crossover and mutation operators. Newly created members replace the worst ones in the current generation using a tournament selection to form the new generation. This process repeats until the number of generations is equal to a predefined number or the current generation goodness is convergent. The best members from the last generation are taken as possible motif candidates.

Each *member* contains a variable λ and a chromosome encoding Θ which is described in Equation (2); each gene in the chromosome represents a column in Θ that has 21 elements for a protein motif or five elements for a DNA motif. There are W+1 such genes in each chromosome. We note that while Bi (2007) proposed encoding PWM using chromosomes, he did not discuss it in detail. Because the length W of a motif is small relative to the sequence lengths, our genetics model consumes less memory than those of Wei and Jensen (2006) or Bi (2007).

The *gene-set* maximum size is $l_0=W/2$, the initial size is 1 and the final size is l, such that,

$$l=2q,\ \text{where } 2q<l_0<2q+1. \quad (26)$$

We use $\delta=0.05$ and $k=9$ for (δ,k)-convergent criterion.

The crossover operator is used to produce two new children from a given pair of parents. We use a two-point gene-set crossover operator with probability $p_c=0.95$. The mutation operator is used to make changes in some portions of the chromosomes of newly created members. Because each chromosome encodes a Θ resulting from an alignment on the whole sequence set, a shift to the left or right one position may improve the quality of the alignment. We therefore propose two different tasks for the mutation operator: (i) change in a gene-set: two rows are selected randomly. Cells corresponding to the selected rows in the given gene-set are exchanged, and (ii) change to whole chromosome: gene-sets in the chromosome are shifted left or right by one position. The blank gene-sets are filled with average probability values. The fitness function of HIGEDA is a combination of the objective function [Equation (14)] and the posterior scoring function [Equation (27)]. The objective function assesses the best fit model, while the posterior scoring function determines how well the model is distinguished from the background in the entire dataset. We use the motif posterior scoring approach of Nowakowski and Tiuryn (2007) and estimate the prior probability of motif occurrence. It follows that the posterior score of subsequence S^{ij} at position j in sequence i has the form:

$$S_p(S^{ij}|\Theta,\lambda)=\log\frac{\lambda*P_M(S^{ij})}{(1-\lambda)*P_B(S^{ij})}. \quad (27)$$

Equations (14) and (27) are normalized and combined. Both are important to the model evaluation but not equally in all contexts. The fitness function is first applied to find best fit models, and then with the posterior scoring function to select the most significant model. To this rule, we add a scaling function $\nu(t)$ which is shown in Figure 3. The fitness function of HIGEDA is then defined as

$$\text{fit}(\Theta)=\frac{\nu}{m}L(S,Z|\Theta)+(\nu_{\text{M}}-\nu)\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{P(z_{ij}=1)}{|P(z_{\text{ij}}=1)|}*S_p(S^{ij}|\Theta,\lambda). \quad (28)$$

To find gapped motifs, in later phases, HIGEDA tries to improve the most significant model by introducing gaps. This is similar to a local alignment problem but instead of aligning every sequence to the sequence set, HIGEDA, using DP, aligns the sequence to the motif model that describes the local alignment of the sequence set. While the run time of gapless alignments is $O(W)$, that of DP

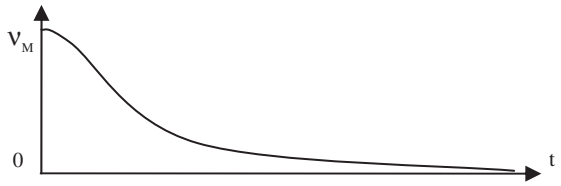

Fig. 3. $\nu(t)=\nu_M \times T/(T+t^2)$. ν_M is the maximum value of ν.

alignment is $O(W^2)$. By restricting the use of DP, the speed of HIGEDA is significantly improved.

HIGEDA algorithm

Input: Sequence set S, motif length W.
Output: Motif model Θ.

(1) Set $M=\{1\}$ once, set gene-set parameters using Equation (26), and randomly generate the first generation, set gNo=0.
(2) For each member in the current generation, apply EM once. Measure member goodness using the fitness function [Equation (28)].
(3) Select parents from the best members using ranking selection; apply genetics operator to create new members.
(4) Create new generation using tournament selection, gNo++.
(5) If the current generation is (δ,k)-convergent, then
save (l), set $l=l_0$,
Apply genetics operators to refresh current generation.
Restore (l), go to Step 2.
(6) If stop criteria are met, then
set $l=l/2$, set gNo$=0$.
If $l>0$, then go to (2).
(7) Take the best member as the motif candidate; run EM with the candidate until convergent to obtain the motif model.
(8) Output the motif model and update M using Equation (13).
(9) Stop, or go to Step (1) to find more motifs.

4 RESULTS AND DISCUSSION

We compared HIGEDA with five open source algorithms: GAME-2006, MEME v3.5, GLAM2-2008, BioProspector-2004 and PRATT v2.1, one web application, GEMFA (http://gemfa.cmh.edu) and for EM motif algorithm, we used results from Bi (2007). All algorithms were run 20 times on each dataset with the runtimes recorded; for GEMFA, this was obtained from the website. Default run parameters were used for all programs. For HIGEDA, the number of generations at each gene-set level was 50 and the population size for all gene-set levels was 60. For algorithms in Tables 1–3, motif lengths were known motif lengths. For Tables 4 and 5, we first ran HIGEDA to find statistically significant motifs, and then ran other algorithms using the lengths of best motifs found by HIGEDA.

Performance measurement As in Bi (2007), two quantities are used to evaluate the algorithms: LPC, the letter level performance coefficient and SPC, the motif site level performance coefficient. If we denote delta(.) as the indicator function, and O_i and A_i, respectively, as the set of known and predicted motif positions in sequence i, then,

$$LPC(S)=\frac{1}{n}\sum_{i=1}^{n}|A_i\cap O_i|/|A_i\cup O_i|, \quad (29)$$

$$\mathrm{SPC(S)}=\frac{1}{n}\sum_{i=1}^{n}delta(A_i\cap O_i\neq Empty). \quad (30)$$

Table 1. Average performance (LPC/SPC) on simulated DNA datasets

Motif identity	Algorithm	Uniform	AT-rich	CG-rich	Average runtime
91%	HIGEDA	1.00/1.00	1.00/1.00	1.00/1.00	32 s
	GEMFA	0.98/1.00	0.98/1.00	1.00/1.00	22 s
	GAME	0.86/0.88	0.88/0.90	0.91/0.94	2 min 26 s
	MEME	1.00/1.00	1.00/1.00	1.00/1.00	2 s
	GLAM2	1.00/1.00	1.00/1.00	1.00/1.00	52 s
	BioPro.	0.99/1.00	0.99/1.00	0.94/0.95	2 s
	PRATT	0.83/0.95	0.88/1.00	0.46/0.80	0.2 s
	EM	0.99/1.00	0.99/1.00	1.00/1.00	
79%	HIGEDA	0.87/0.96	1.00/1.00	1.00/1.00	32 s
	GEMFA	0.87/0.88	0.87/0.90	0.85/0.89	18 s
	GAME	0.43/0.55	0.55/0.61	0.64/0.71	2 min 11 s
	MEME	0.95/0.95	1.00/1.00	1.00/1.00	2 s
	GLAM2	1.00/1.00	0.07/0.25	1.00/1.00	57 s
	BioPro.	0.86/0.92	0.89/0.95	0.94/0.98	2 s
	PRATT	0.46/0.70	0.03/0.15	0.09/0.15	0.2 s
	EM	0.83/0.87	0.89/0.91	0.87/0.89	
70%	HIGEDA	0.79/0.81	0.84/0.89	0.65/0.76	32 s
	GEMFA	0.56/0.65	0.50/0.60	0.52/0.56	20 s
	GAME	0.14/0.28	0.19/0.34	0.20/0.34	1 min 55 s
	MEME	0.44/0.50	0.75/0.75	0.27/0.30	2 s
	GLAM2	0.95/0.95	0.01/0.05	0.00/0.05	1 min 20 s
	BioPro.	0.26/0.33	0.39/0.44	0.25/0.33	2 s
	PRATT	0.31/0.40	0.05/0.10	0.19/0.30	0.2 s
	EM	0.38/0.48	0.47/0.58	0.48/0.54	

4.1 Simulated DNA datasets

We generated simulated DNA datasets as in Bi (2007), with three different background base compositions: (i) uniform, where A, T, C, G occur with equal frequency, (ii) AT-rich (AT$=60\%$) and (iii) CG-rich (CG$=60\%$). The motif string, GTCACGCCGATATTG, was merged once or twice into each sequence, after a defined level of the string change: (i) 9% change representing limited divergence (i.e. 91% of symbols are identical to the original string), (ii) 21% change, or (iii) 30% change (essentially background or random sequence variation).

In Table 1, we compare results with those obtained with seven other algorithms. When motif sequences are 91% identical, all algorithms perform equally well regardless of base composition. However, when the identity drops to 79 or 70%, HIGEDA in general performs as well or significantly better than other algorithms on all base compositions. Hence, HIGEDA is not significantly affected by noise.

Table 2. Average performance of seven algorithms (LPC/SPC/run time) on eight DNA datasets (number of sequences/length of motif/number of motif occurrences)

Algorithm	E2F(27/11/25)	ERE(25/13/25)	crp(24/22/18)	arcA(13/15/13)	argR(17/18/17)	purR(20/26/20)	tyrR(17/22/17)	ihf(17/48/24)
HIGEDA	0.57/**0.96**/33 s	0.58/0.87/37 s	**0.75/0.90**/22 s	0.38/0.53/56 s	**0.75/0.94**/1min 22 s	**0.90**/0.93/1 min 38 s	0.34/0.41/1 min 19 s	**0.14**/0.31/3 min 44 s
GEMFA	0.64/0.85/39 s	**0.74/0.92**/43 s	0.57/0.88/12 s	0.32/0.42/26 s	0.35/0.38/30s	0.81/0.90/39s	0.32/0.49/27 s	0.11/0.26/1 min
GAME	0.24/0.90/2 min 45 s	0.24/0.75/2 min 11 s	0.45/0.80/2 min 33 s	0.05/0.10/1 min 26 s	0.36/0.55/2 min 31	0.33/0.53/3 min 23 s	0.11/0.23/2 min 39 s	0.07/0.18/10 min 7 s
MEME	0.71/0.85/16 s	0.68/0.68/4 s	0.55/0.68/1 s	**0.47/0.54**/4 s	**0.75/0.94**/5 s	0.58/0.85/4 s	**0.43/0.47**/3 s	0.00/0.00/5 s
GLAM2	**0.84**/0.93/1 min 28 s	**0.74/0.92**/1 min 10 s	0.54/0.64/1 min 13 s	**0.47/0.54**/1 min 17 s	0.38/0.47/1 min 32 s	0.17/0.50/3 min 11 s	0.35/0.35/1 min 55 s	0.12/**0.33**/9 min 55 s
PRATT	0.17/0.33/0.2 s	0.24/0.44/0.2 s	0.22/0.56/0.1 s	0.17/0.23/0.4 s	0.16/0.29/0.2 s	0.46/**0.95**/0.2 s	0.06/0.12/0.3 s	0.06/0.25/0.5 s
BioPros.	0. 50/0.63/3s	0.65/0.71/3s	0.37/0.45/1s	0.01/0.01/6s	0.62/0.69/8s	0.18/0.20/7s	0.33/0.42/6s	0.05/0.13/17s

Bold indicates the best performance on each dataset.

Table 3. Detection of protein motifs (nos. 1–8, PFAM; nos. 9–12, prosite)

Family (no. seq, max *L*) Algorithm (run time)	Known motif Predicted motif
1. DUF356 (20 158)	I H P P A H
HIGEDA (12 s)	I H P P A H
MEME (2 s)	I H P P A H
GLAM2 (1 min 58 s)	I H P P A H
PRATT (0.1 s)	I H P P x H
2. Strep-H-triad (21 486)	H x x H x H
HIGEDA (40 s)	H G D H Y H
MEME (5 s)	H G D H Y H
GLAM2 (1 min 51 s)	H G D H Y H
PRATT (0.1 s)	H x x H x H
3. Excalibur (26 296)	D x D x DG xx CE
HIGEDA (1 m)	D R D [RNGK] DG [IV] [AG]CE
MEME (4 s)	[WY] Q [GA] [NW] Y Y L K S D
GLAM2 (2 min 56 s)	D R D K D G V A C E
PRATT (0.5 s)	D x D xxx C
4. Nup-retrotrp (14 1475)	G R K I x x x x x R R K x
HIGEDA (3 min 18 s)	S G R K I K T A V R R K K
MEME (10 s)	W[DE]C[DE][TV]C[LC][VL]QNK[AP][ED]
GLAM2 (6 min 55 s)	S N G K N M F S S S G T S
PRATT (1 s)	F S S S [GP] T x x S x(1,2) R K
5. Flagellin-C (421,074)	N R F x S x I x x L
HIGEDA (1 min 29 s)	RA [NDQG] L G A [FV] Q N R
MEME (6 s)	R [AS] [DNQ] L G A [VF] Q N R
GLAM2 (3 min 35 s)	R A D L G A F Q N R
PRATT (0.07 s)	A-x-Q
6. Xin repeat (253 785)	GDV[KQR][TSG]x[RKT]WLFETxPLD
HIGEDA (5 min 45 s)	GDV[RK] [ST] [ACT] [RK] WLFETQPLD
MEME (1 min 1 s)	KGDV [RK] T [CA][RK]W[LM]FETQPL
GLAM2 (6 min 36 s)	H K G D V R T C R W L F E T Q P
PRATT (6 s)	G D V x T x x W x F E T x P
7. Planc. Cyt. C (121 131)	C{CPWHF}{CPWR}CH{CFYW}
HIGEDA (1 min)	[NHKSY]**C**[AQELMFTV][AGS]**CH**
MEME (9 s)	F S P D G K
GLAM2 (3 min 42 s)	R F S P D G
PRATT (0.02 s)	P D x x x L
8. zf-C2H2 (11 599)	Cx(1–5)Cx3#x5#x2Hx(3-6)[HC]
HIGEDA (2 min 12 s)	**C**[_Y][_RQEK][_C][_EKP][_EGIK]**C**[_G] K[ARST]**F**S[RQK][KS]S[NHS] **L**[NKT][RKST]**H**[QILKM]R[ISTV]**H**
MEME (45 s)	EIC[NG]KGFQRDQNLQLHRRGHNLPW
GLAM2 (19 min 17 s)	YKC__P_CGK_FS_KSSLT_H__RI__HT
PRATT (0.1 s)	Hx(3–5)H
9. EGF_2 (163 235)	CxCx2[GP][FYW]x(4–8)C
HIGEDA (4 min 19 s)	C[_EK][_C][_EILSV]C[NDE][NDQEPS]G
MEME (15 s)	[FWY][AQESTY]G[DS]DCS[GI]
GLAM2 (8 m)	GECxCNxGYxGSDCSI
PRATT (1 s)	CNx(0–9)GECxCNEGWSGDDC Gx(0–1)CxCx5Gx2C

(continued)

Table 3. Continued

Family (no. seq, max *L*) Algorithm (run time)	Known motif Predicted motif
10. LIG-cyclin-1 (231 684)	[RK]xLx(0–1)[FYLIVMP]
HIGEDA (1 min 8 s)	[RK]R[RL][RIL][DEIFY]
MEME (18 s)	PAPAP
GLAM2 (2 min 35 s)	Tx(0–1)RKP
PRATT (0.5 s)	Kx(0–1)R
11. LIG_PCNA (131 616)	x(0–3)x{FHWY}[ILM]{P}{FHILVWYP} [DHFM][FMY]
HIGEDA (1 min 32 s)	[RGIKP][_QK][_RDKS][DPST][IL] [KMTY]
MEME (9 s)	[SV]FFG
GLAM2 (4 min 59 s)	GQKTIMSFFS
PRATT (0.3 s)	Qx(0–1)SIDSFF[K_][R_] Lx2Sx(2–4)K
12. Mod_Tyr_Itam (10 317)	[DE]x(2)Yx(2)[LI]xx(6–12)Yx(2)[LI]
HIGEDA (51 s)	[RDQE][_S][_CET][_IL][_GM][_QTY] [_QELK][_DGK][_EI][_RS][_RN][_RGP] LQ[DGV][GHT]YxM[CIY]Q[NGT]L[ILS]
MEME (2 s)	GKEDDGLYEGLNIDDCATYEDIHM
GLAM2 (3 min 43 s)	E_H_____SLAQKSM_DH _SRQ_
PRATT (0.1 s)	Lx2Lx(0–1)L

Table 4. Motifs of *ZZ*, Myb and SWIRM domains by the four algorithms

Algorithm	Domain/found motif	*P*	% hit
	ZZ (675 125)		
HIGEDA	D[FY]DL**C**[AQES]x**C**-[EYV]	5.76e-7	91.67
	C[GHPY]D[FY]DL**C**[AQES]x**C**	2.25e-6	93.06
MEME	CLICPDYDLC		73.00
GLAM2	HSRDHP[ML][IL][QRK]		91.00
PRATT	Cx2C		92.00
	SWIRM (49 624): no motif found		
	ZZ-Myb (352 697)		
HIGEDA	[ND]W[ST]A[DE]EELLLL	3.81e-9	97.14
MEME	WTADEELLLLD		100.0
GLAM2	WTAEEELLLLE		97.00
PRATT	Ex2Lx(4–5)E		96.00
	ZZ-Myb-SWIRM (431 049)		
HIGEDA	GNW[AQ]DIADH[IV]G[NGST]	3.6e-10	100.0
	WGADEELLLLEG	2.4e-10	100.0
MEME	WGADEELLLLEG		100.0
GLAM2	KQLCNTLRILPK		85.00
PRATT	Wx(1–2)D[ADEQ]Ex(2–3)L[ILV]		100.0

Number of proteins and maximum protein length for each family are shown in parentheses.

Table 5. Motifs of Presenilin-1 and signal peptide peptidase by HIGEDA

Algorithm	Family (number of seq, max)/Found motif	*P*	% hit
	Presenilin-1 (86 622)		
HIGEDA	**GLGD**FIFYS	1.3e-10	85.00
	KL**GLGD**FIFY	1.6e-10	85.00
MEME	HWKGPLRLQQ		66.00
GLAM2	GDFIFYSLVL		85.00
PRATT	Lx(1–3)Lx(2–3)I		100.0
	Signal peptide peptidase (151 690)		
HIGEDA	F[AS]ML**GLGD**IVIPG	8.9e-08	88.74
	YDIFWVF[GF]T[NDP]	1.7e-08	86.75
	GLF[IFV]YDIFWVF	8.7e-08	86.75
	FGT[NDP]VMVTVA[KT]	8.6e-08	86.09
MEME	FWVFGT[ND]VMV		86.00
	YDIFWVFGT[NDP]VMV		87.00
	[AS]MLGLGDIVIPGI		92.00
GLAM2	[FL][FI]YD[IV]F[WF]VF[GF]		95.00
	GL_FFYDIFWVFGT		90.00
PRATT	Lx3F		100.0

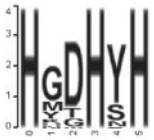

Fig. 4. Strep-H-triad motif by HIGEDA.

4.2 Finding motifs in biological DNA sequences

We used eight DNA transcription factor binding site datasets, two eukaryotic datasets, ere and e2f (Bi, 2007), and six *Escherichia coli* datasets: crp, arcA, argR, purR, tyrR and ihf (Osada *et al*., 2004).

Table 2 shows that HIGEDA performs as well or better than other algorithms on most datasets. One exception is ERE, possibly because it is an example of OOPS which is the motif finding model used by GEMFA. Also, GLAM2 performs better than HIGEDA on ihf, which contains a motif of length 48.

4.3 Finding motifs in biological protein sequences

We selected 12 protein families from PFAM and Prosite to examine motifs with different levels of sequence specificity, from completely defined to more subtle, degenerate and gapped.

Results obtained with HIGEDA are compared to those obtained with other algorithms. For algorithms that support gaps, parameters are set to those of the known gap structures. All algorithms identify the unambiguous DUF356 IHPPAH motif. For the Strep-H-triad motif, the HIDEGA, MEME and GLAM2 consensus sequences are more specific than the known motif because of the motif decoding procedure. Full degeneracy is shown in Figure 4.

HIGEDA and GLAM2 identify the exact motif for the moderately degenerate Excalibur. Only HIGEDA identifies Nup-retrotrp motif erring in only one residue and identifying the key arginine (R) residues. No algorithm identifies the Flagellin-C motif, possibly because of the weak motif signal. While all perform well in finding the Xin repeat motif, only HIGEDA identifies the Planctomycete Cytochrome *C* motif, erring in only one residue. For the gapped motif ZF-C2H2 family, only HIGEDA identifies the key C, H, F and L residues and the spacing between them. HIGEDA also provides better results on EGF_2 and LIG_CYCLIN_1. No algorithm identifies the LIG_PCNA and MOD_TYR_ITAM motifs. Together, these results show that HIGEDA can effectively find more subtle motifs, including those with gaps. Because HIGADA uses GA to find motif seeds, its run times are longer than those of MEME and PRATT but shorter than those of GLAM2, as indicated in Table 3.

4.4 New motif discovery

To test HIGEDA in prediction of novel motifs, we selected the following proteins from PFAM: those containing only a ZZ domain, only a SWIRM domain, a ZZ plus a Myb domain, and a ZZ plus both a Myb and a SWIRM domain. While ZZ domains have a consensus sequence in PFAM, SWIRM and MYB proteins are defined only by experimental demonstration of functional properties and protein sequence alignments. We ran HIGEDA on these groups varying the output motif length from 4 to 33. We defined statistically significant motifs as those found in more than 85% of sequences, and having a *P*-value <0.0001 [computed using Touzet and Varre (2007)]. We also ran MEME, GLAM2 and PRATT.

In Table 4, we first show that HIGEDA, MEME and PRATT successfully identify patterns of the known ZZ motif, CX(1–5)C. No motif was discovered in the SWIRM-containing proteins. But HIGEDA, MEME and GLAM2 each discovered a motif, WxAxEELLLL, common to Myb proteins, which we propose as a novel domain. Uniquely discovered by HIGEDA, we propose a second novel domain, GNW[AQ]DIADH[IV]G[NGST], of unknown function, which is specific to the ZZ-SWIRM-Myb protein set.

Lastly, we selected the protein families, Presenilin-1 and Signal peptide peptidase, that share the conserved sequence GXGD and are grouped in the same clan in PFAM. As shown in Table 5, only HIGEDA successfully discovered motifs containing GLGD in these families. In addition, two patterns, YDIFWVF, which is also identified by MEME and GLAM2, and FGT[NDP]VMVTVA[KT] which is also identified by MEME, appear frequently (in 86% of 151 protein sequences) in the Signal peptide peptidase family. We, therefore, propose these patterns as new motifs of this family.

5 CONCLUSION

We have demonstrated a new algorithm, HIGEDA that integrates HGA, EM and DP to find motifs in biological sequences. HIGEDA uses HGA to manage different motif seeds and to escape local optima more efficiently than conventional GAs by using gene-set levels. By applying DP and proposing a new technique for aligning sequences to a motif model, and then using EM with the dynamic alignments, the proposed method creates a new way to automatically insert gaps in motif models. Using the gradient descent learning law, HIGEDA effectively estimates its model parameters without the greedy searches used in MEME. Using the pseudo-counts method based on a simple mechanism and the BLOSUM62 substitution probability matrix, HIGEDA avoids the small dataset problem of zeros in motif model elements, without using the Dirichlet mixture prior knowledge used in MEME and similar approaches. Because HIGEDA uses a set of motif seeds generated randomly,

it outperforms MEME, GLAM2 and several other algorithms in finding subtle, more degenerate and gapped motifs. Lastly, we have shown that, as a TCM based algorithm, HIGEDA can identify novel motifs.

Funding: Vietnamese Ministry of Education and Training (to T.L.), the Coleman Institute for Cognitive Disabilities and the Anna and John J. Sie Foundation (to K.G.).

Conflict of Interest: none declared.

REFERENCES

Bailey,T.L. and Elkan,C. (1995). The value of prior knowledge in discovering motifs with MEME. *Proc. Intl. Conf. Intel. Syst. Mol. Biol.,* **3**, 21–29.

Bi,C. (2007) A genetic-based EM motif-finding algorithm for biological sequence analysis. *Proc. IEEE Symp. Comput. Intel. Bioinfo. Comput. Biol*., 275–282.

Chang,X. *et al.* (2006) Prediction of transcription factor binding sites using genetic algorithm. *1st Conf. Ind. Elec. Apps.,* 1–4.

Frith,M.C. e*t al.* (2008) Discovering sequence motifs with arbitrary insertions and deletions. *PLoS Comput. Biol.*, **4**, e1000071.

Henikoff,J.G. and Henikoff,S. (1996) Using substitution probabilities to improve position-specific scoring matrices. *Comp. App. Biosci.,* **12**, 135–143.

Hong,T.-P. and Wu,M-T. (2008) A hierarchical gene-set genetic algorithm. *J. Comp.*, **3**, 67–75.

Li,L. *et al.* (2007) GAPWM: a genetic algorithm method for optimizing a position weight matrix. *Bioinformatics*, **23**, 1188–1194.

Li,L. *et al.* (2008) fdrMotif: identifying cis-elements by an EM algorithm coupled with false discovery rate control. *Bioinformatics*, **24**, 629–636.

Liu,D. *et al*. (2006) Motif discoveries in unaligned molecular sequences using self-organizing neural networks. *IEEE Trans. Neural Networks*, **17**, 919–928.

Nowakowski,S. and Tiuryn,J. (2007) A new approach to the assessment of the quality of predictions of transcription factor binding sites. *J. Biomed. Info.,* **40**, 139–149.

Osada,R. *et al*. (2004) Comparative analysis of methods for representing and searching for transcription factor binding sites. *Bioinformatics*, **20**, 3516–3525.

Pisanti,N. *et al.* (2005) Bases of motifs for generating repeated patterns with wildcards. *IEEE/ACM Trans. Comput. Biol. and Bioinfo.*, **2**, 40–50.

Touzet,H. and Varré,J.S. (2007) Efficient and accurate P-value computation for position weight matrices. *Algorithms for Mol. Biol.*, **2**, 15–26.

Wei,Z. and Jensen,S.T. (2006) GAME: Detecting cis-regulatory elements using a genetic algorithm. *Bioninformatics*, **22**, 1577–1584.

Xie,J. *et al.* (2004) A Bayesian insertion/deletion algorithm for distant protein motif searching via entropy filtering. *J. Am. Stat. Assoc.,* **99**, 409–420.