PSOMF: An algorithm for pattern discovery using PSO

F. Zare-Mirakabad[1], H. Ahrabian[1,2], M. Sadeghi[1,3], J. Mohammadzadeh[2], S. Hashemifar[2], A. Nowzari-Dalini[1,2], and B. Goliaei[1,⋆]

[1] Department of Bioinformatics, Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran.
[2] Center of Excellence in Biomathematics, School of Mathematics, Statistics, and Computer Science, University of Tehran, Tehran, Iran.
[3] National Institute of Genetic Engendering and Biotechnology, Tehran, Iran.

Abstract. The task of transcription factor binding sites discovery from the upstream region of gene, without any prior knowledge of what look likes, is very challenging. In this paper we propose an algorithm based on Particle Swarm Optimization (PSO) to identify motif instances in multiple biological sequences. The experimental results on yeast saccharomyces Cerevisae transcription factor binding sites, demonstrate that the proposed method is working analogous to YMF, MEME and AlignACE algorithms.

Key words: motif, motif discovery, PSO algorithms

1 Introduction

Recognition of pattern in a set of DNA sequences is a useful step in understanding the regulation of gene expression [1]. Gene expression is the process whereby a gene is transcribed to form an RNA sequence which is then used to produce the corresponding protein sequence. In this level of transcription a protein called transcription Factor (TF) binds to specific sites that are called Transcription Factor Binding Sites (TFBSs). The TFBSs and other genomic regulatory elements with specific structure and function are called motifs or signals. The motifs are usually defined by a subsequence with most occurrence in a set of unaligned DNA sequences. In the simplest form, motif finding problem is defined as follows: for a given promoter sequence set $S = \{s_1, \ldots, s_t\}$ find all the overpresented subsequences of length ℓ (ℓ-mers) that occur with some mismatches in the sample set S [2].

In the literature, for this problem many algorithms are proposed [3, 4, 5, 6, 7, 1, 8, 9]. These algorithms can be divided into two major groups: deterministic and nondeterministic [10]. Most deterministic algorithms use regular expression based rules to specify some classes of allowable patterns for motifs, and these algorithms are in exhaustive nature, YMF [11], Pratt [12] and TEIRESIAS [13] are

⋆ Corresponding author Email:goliaei@ibb.ut.ac.ir

M. Chetty, S. Ahmad, A. Ngom, S. W. Teng (Eds.): PRIB2008, Supplementary Conf. Proc., pp.61-72, 2008.

examples of deterministic algorithms that use regular expression rules to identify motifs. On the other hand, most nondeterministic motif discovery algorithms are non-exhaustive and stochastic in nature, and in different runs find different motifs that may or may not be the optimal one. These stochastic algorithms are based on position weight matrix (PWM). By allowing different credits for each nucleotide, PWM contains more information than regular expression. Some popular stochastic motif discovery tools are MEME [3], CONSENSUS [6], Gibbs sampling [7], MotifSampler [14], AlignACE [15], and BioProspector [16]. Some of these algorithms, AlignACE, BioProspector and Motif-Sampler are based on Gibbs sampling method while MEME is based on expectation maximization technique.

Recently evolutionary algorithms such as genetic algorithms has been used for pattern discovery in multiple unaligned DNA sequences [17, 18, 19, 20, 10, 21, 22]. This problem also can be attacked by Particle Swarm Optimization (PSO). PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy [23], inspired by social behavior of bird flocking or fish schooling. In PSO a number of simple entities, *particles*, are placed in the solution search space of a problem for finding an optimal solution. Each Particle has a coordinates x, which records a potential solution of the problem and a velocity v which determines the direction that the particle will go through the solution search space for finding optimal solution. Similar to the genetic algorithm, PSO is initialized with a population composed of random particles (first generation) and then each particle searches for an optimal solution by updating generations iteratively. In each iteration, each particle is updated based on the following two best values. The first one is the best solution obtained so far by the current particle in the population. Another best is the best solution obtained so far by any particle in the population. Therefore, the direction of each particle in the population is adjusted through the search space based on these best solutions. The process is then iterated for a fixed number of times.

In this paper we present a probabilistic method, PSOMF (Particle Swarm Optimization for Motif Finding), based on the PSO for motif finding problem. The main advantage of our proposed method is that it can identify motif instances by a population size and number of generations much less than that is needed for genetic algorithms. PSOMF usually applies 20 particles in population and reaches to a unique solution in only 5 generations. We perform experiments on the real data sets and compare them with three well-known motif finding tools YMF [11], AlignACE [15] and MEME [3] to demonstrate the effectiveness of our proposed method. It is noted that the reason for choosing these tools for comparison is because of the accuracy of their results comparing with the other existing tools so far.

2 Definitions and Notations

In this section some definitions and notations further used in this paper are introduced. Here, a sequence is a string on a given alphabet Σ thus $\Sigma = \{A, C, G, T\}$

for DNA. Data set $S = \{s_1, \ldots, s_t\}$ consists of t sequences $s_i = s_i[1], \ldots, s_i[n]$ such that $1 \leq i \leq t$, and n denotes the length of sequence s_i. It is desired to find the motif or common pattern δ of length ℓ in $S = \{s_1, \ldots, s_t\}$. Let $s_i[j_1], \ldots, s_i[j_1+\ell-1]$ is a ℓ-mer or subsequence of s_i and $s_k[j_2], \ldots, s_k[j_2+\ell-1]$ is a ℓ-mer of s_k, the number of matches between these two ℓ-mer is called match score of these two subsequences. The number of matches between a subsequence x and y with length ℓ is defined as:

$$M(x, y) = \sum_{j=1}^{\ell} d(x[j], y[j]), \tag{1}$$

where

$$d(a, b) = \begin{cases} 1 & \text{if } a = b, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The pattern instance of pattern e with length ℓ in sequence s_i is a subsequence $s_i[j], \ldots, s_i[j + \ell - 1]$ with the maximum match score with e. The set $U = \{u_1, \ldots, u_t\}$ is called the pattern instance set of pattern e if each u_i is a pattern instance of e in s_i. The summation of match score of pattern e with each element u_i of pattern instance set U of e is called occurrence score of e and is denoted by $OCC(e)$. The set $P = \{p_1, \ldots, p_t\}$ is called the position pattern instance set of pattern e such that p_i is the start position of pattern instance u_i in s_i *i.e.* if $u_i = s_i[j], \ldots, s_i[j + \ell - 1]$ then $p_i = j$. Clearly each u_i has most match (maximum match score) with e between all $n - \ell + 1$ ℓ-mers in s_i. The pattern shared by the pattern instance set $U = \{u_1, \ldots, u_t\}$ which has position pattern instance set $P = \{p_1, \ldots, p_t\}$ is referred to a consensus or a consensus pattern and shown by δ. In fact, in motif finding problem (without having any pre-given pattern) we should find in a given data set $S = \{s_1, \ldots, s_t\}$ a pattern instance set $U = \{u_1, \ldots, u_t\}$ whose all elements have most match together and consensus pattern of U is reported as motif.

To find transcription factor binding sites using PSO, the first problem is to code each particle. In general form, each particle d is composed of four items which are defined as follow:

- The X-array, x_d denotes the current position of the particle in search space.
- The Y-array, y_d records the position of the best solution found so far by the particle.
- The V-array, v_d contains a velocity for each particle.
- The Y-fitness, b_d contains the fitness of the Y-array.

Now, in motif finding problem the above items ares defined as follows. We use binary representation because it is seen that an optimizer which operates on binary value function might be advantageous [24]. Thus to create a binary search space, each X-array of a particle is considered as a $0, 1$ matrix which is created from a position pattern instance set as follows. Let p_i denotes the position of a

pattern instance u_i in the sequence s_i, then we have

$$x[i,j] = \begin{cases} 1 & \text{if } j = p_i, \\ 0 & \text{otherwise.} \end{cases} \quad \text{for} \quad 1 \leq j \leq n$$

In fact all positions are zero except the position of the occurrence of pattern instance. It is clear that a position pattern instance set can be created from the X-array. It should be noted that, for initial population (described in detail in next section) the position pattern instance sets are selected randomly and the pattern instance sets are created based on them. In Fig. 1 a few co-regulated DNA sequences are shown and Fig. 2 shows the X-array of particle corresponding to these sequences. In this figure TFBSs are declared by underlines. To find the consensus pattern δ corresponding to each particle d, we construct the position pattern instance set $P_d = \{p_{d,1}, \ldots, p_{d,t}\}$ based on X-array x_d. Each entry of X-array x_d with value 1 in row i $(1 \leq i \leq t)$, shows the position of unique pattern instance $p_{d,i}$ in the sequence s_i as cited before. By constructing P_d, corresponding pattern instance set U_d is also created, and all the sequences of U_d are aligned and the shared pattern is considered as a consensus pattern δ.

The X-array with best fitness found by the each particle, is stored in Y-array. Thus this array is similar to X-array and is also called best local solution.

A change in particle velocity can be interpreted as a change in the probability of finding the particle in one position or another. Since this change is a stochastic value, it is limited to a range of $[0, 1]$. Thus the V-array is a matrix and each entry contains a change value for each entry X-array.

The Y-fitness is a variable b whose value is the fitness of the best solution which is found by particle, *i.e.* the fitness of best local solution is stored in Y-array.

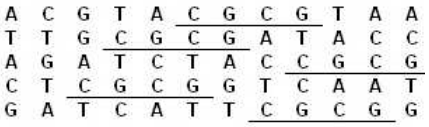

Fig. 1. Co-regulated DNA sequences.

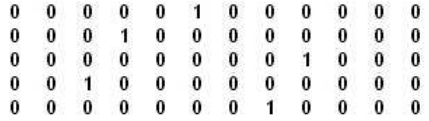

Fig. 2. X-array of particle corresponding to the sequences of Fig. 1.

As mentioned previously, in order to generate new solutions, the best global solution followed by its fitness are stored in each iteration of PSO algorithm.

3 PSO algorithm for Motif Finding

In the proposed algorithm each particle searches for the optimal solution by sharing historical information and social information between the particles [23]. The definition of population, fitness function, evaluation and update procedure are given in below.

3.1 Construction of the Population

As we know, given the set $S = \{s_1, \ldots, s_t\}$ (where the length of s_i is n for $1 \leq i \leq t$), inorder to find a motif of length ℓ in this set, we need to investigate the search space. In each s_i, we have $n-\ell+1$ number of subsequence of length ℓ, so in the set S, we have $(n-\ell+1)^t$ number of potential motif instances of length ℓ. Thus, the search space in this problem is composed of $(n-\ell+1)^t$ elements. Among them, the element with best score is selected as a motif instance. Therefore, by assigning m number of these potential motif instances from the search space in random to the particles, the initial population is constructed. In detail, each particle is generated as follows: For each sequence s_i $(1 \leq i \leq t)$, a random number j is selected from $[1, n]$ and assigned to p_i. Therefore p_i denotes the start position of a pattern in sequences s_i and $P = \{p_1, \ldots, p_t\}$ is the position pattern instance set. As described in Section 2, the X-array is also created from P. If the consensus pattern δ corresponding to this X-array is not a random, we consider it as an X-array of a valid particle in population. For justifying that our obtained X-array is not a random, we obtain the occurrence score of δ (as described in Section 2) and if $OCC(\delta)$ is greater than α (α is a predefined threshold), we can deduce that X-array is random and it could not be an X-array of a particle of initial population. If the X-array is not random, the V-array of a particle is set to 0.2 and Y-array is set to zero and these arrays are added to the initial population as a particle.

The above process is repeated until m particles are obtained for the initial population.

3.2 Fitness Function

The fitness function used for evaluating of a particle is the joint information content (JIC) of the pattern instance corresponding to that particle. First, the position pattern instance set $P = \{p_1, \ldots, p_t\}$ and the pattern instance set $U = \{u_1, \ldots, u_t\}$ of a particle d is created from its X-array. Later, the pattern instance set U is considered as a matrix whose i-th row is u_i. The four dimensional array $F_{4,4,\ell,\ell}$ is created from U such that $F[i, i', j, j']$ is the joint frequency of the pair

nucleotides i and i' ($i, i' \in \{A, C, G, T\}$) in column j and j' ($1 \leq j < j' \leq \ell$) of U. Now, joint position weight array $W_{4,4,\ell,\ell}$ is created as:

$$W[i, i', j, j'] = \frac{F[i, i', j, j']}{t} + \text{pseudocount}, \tag{3}$$

where $i, i' \in \{A, C, G, T\}$ and $1 \leq j, j' \leq \ell$, and pseudocount is an arbitrary small number (*e.g.* 0.0001) for avoiding zero value of W.

Then, JIC fitness of particle d can be computed as:

$$fitness(d) = \sum_{i} \sum_{i'} \sum_{j=1}^{\ell} \sum_{j'=j+1}^{\ell} W[i, i', j, j'] \log \frac{W[i, i', j, j']}{w_0}, \tag{4}$$

where $i, i' \in \{A, C, G, T\}$ and w_0 is the background joint frequency of a pair nucleotides and is considered as $1/16$. Thus the Y-fitness of a particle d is calculated as the sum of the joint information contents of nucleotide pairs of each column of the pattern instance set U.

3.3 Evaluation and Updating

For generating a new population, the particles of the current population are evaluated and updated. Each particle d is evaluated according to the fitness function. Let f_d denotes the fitness value of particle d, and g is denoted the best obtained solution, and *gbest* denotes the fitness of g, then the V-array v_d, X-array x_d, Y-array y_d, and best local fitness b_d are adjusted in the direction of the particle with the best previous local position and the best previous global position between all particles in population.

The velocity array of particle d is updated by the following formula:

$$v_d[i, j] = v_d[i, j] + c_1 r_1 (y_d[i, j] - x_d[i, j]) + c_2 r_2 (g[i, j] - x_d[i, j]), \tag{5}$$

$$\text{for } 1 \leq i \leq t \text{ and } 1 \leq j \leq n,$$

such that factors c_1 and c_2 are constant values and are known as acceleration coefficient (Here $c_1 = c_2 = 2$), r_1 and r_2 are random numbers in the range $[0, 1]$.

The position matrix of d is updated by the following equation:

$$x_d[i, j] = \begin{cases} 1 & \text{if } r < s(v_d[i, j]) \\ 0 & otherwise, \end{cases} \tag{6}$$

where r is a random number selected from a uniform distribution in $[0, 1]$ and the function $s(v)$ is a sigmoid limiting transformation function which is defined as:

$$s(v) = \frac{1}{1 + e^{-v}} \tag{7}$$

Obviously, this method for updating x_d may produce more than one value of 1 in each row of x_d which is against to our assumption. In order to solve this

problem, the following steps can be proceeded. Let $x_d[i, j_1] = 1$ in i-th row and we get another value 1 in the next step for $x_d[i, j_2]$. Now, the position pattern instance set and pattern instance set corresponding to the x_d with $x_d[i, j_1] = 1$ and $x_d[i, j_2] = 0$ is obtained and its corresponding fitness f_1 is computed. Similarly, again the position pattern instance set and pattern instance set corresponding to the x_d with $x_d[i, j_1] = 0$ and $x_d[i, j_2] = 1$ is obtained and its corresponding fitness f_2 is computed. The maximum value of f_1 and f_2 determines which element $x_d[i, j_1]$ or $x_d[i, j_2]$ stays 1 and the other changes to 0, respectively. The above process can be performed for all the other confliction that may occur in each row of x_d and eventually we obtain a matrix with a unique 1 in each row.

The best local position matrix and the best local fitness of d is updated as follows: if $f_d > b_d$ then $y_d[i, j] = x_d[i, j]$ for $1 \leq i \leq t$, $1 \leq j \leq n$, and $b_d = f_d$.

The best global position matrix and the best global fitness is updated as follows: if $f_d > gbest$ then $g[i, j] = x_d[i, j]$ for $1 \leq i \leq t$, $1 \leq j \leq n$, and $gbest = f_d$.

3.4 PSOMF Algorithm

Now with respect to the above discussion, the PSOMF algorithm is demonstrated in Fig. 3. The input of algorithm is the set $S = \{s_1, \ldots, s_t\}$ where length of each s_i is n, and the output of the algorithm is the consensus motif with length ℓ of the best motif instance set. Preliminarily, the initial parameters of algorithm such as population size m, maximum generation N, and factors c_1 and c_2 are set. The initial population with the m particles is constructed. Later, employing the fitness function all the initial particle are evaluated and scored. The evaluation and updating of each particle is performed in each iteration for N times, based on

Algorithm PSOMF
Begin

0 set parameter values.
1 create initial population G_0 with m particles.
2 $k = 0$.
3 while $k < N$ do begin
  3.1 evaluate each particle d of population G_k according to fitness function.
  3.2 update V-array of each particle d of population G_k.
  3.3 update X-array of each particle d of population G_k.
  3.4 update Y-array of each particle d of population G_k.
  3.5 update Y-fitness of each particle d of populationG_k.
  3.6 update global solution and global fitness.
  3.7 $k = k + 1$.
4 end
5 select and report the best solutions.

End

Fig. 3. Particle swarm optimization algorithm PSOMF.

the evaluation process discussed in the previous section. Finally, the 10 particles with the best local fitness are announced as the result of PSOMF (note that in biological sequences there might be more than one TFBS). To find the final motifs the following process are performed. First, pattern instance set $U = \{u_1, \ldots, u_t\}$ of each particle d (from 10 selected particle) is obtained according to the X-array. For each motif instance joint position weight array W is constructed as mentioned before. This array is aligned with each ℓ-mer (subsequence) of s_i $(1 \leq i \leq t)$ of data set $S = \{s_1, \ldots, s_t\}$ and the ℓ-mer with maximum match score is considered as motif. To align array W with each ℓ-mer of s_i, assume $s_i[j], \ldots, s_i[j+\ell-1]$ be a ℓ-mer which is started from position j in s_i, then the score of this ℓ-mer is computed as:

$$score(s_i[j], \ldots, s_i[j+\ell-1]) = \sum_{j_1=j}^{j+\ell-1} \sum_{j_2=j_1+1}^{j+\ell-1} \log(\frac{W[s_i[j_1], s_i[j_2], j_1, j_2]}{w_0}),$$

where w_0 is the background joint frequency and is considered as $1/16$.

Now, the time complexity of the algorithm is discussed. In Step 1, each particle is generated in $O(tn + \ell nt)$, where $O(tn)$ is the time complexity for values assigning of the array X and $O(\ell nt)$ is the computation time of the function OCC. Clearly, the generation of m particles take $O(m\ell nt)$. Evaluation of each particle in step 3.1 takes $O(tn + 16\ell^2)$, where $O(tn)$ is the construction time of the PWM and $O(16\ell^2)$ is the evaluation time of the fitness function. The Steps 3.2, 3.4, and 3.5 each takes $O(tn)$, and Step 3.3 takes $O(\ell^2 tn)$ for each particle, and for all particles take $O(m\ell^2 tn)$. Totally, the time complexity of the algorithm is $O(Nm\ell^2 tn)$, which for small value of N can be considered as $O(m\ell^2 tn)$.

4 Experimental result

We employ data set SCPD to test our algorithm. SCPD is a well-known promoter database of the yeast sacchromyces cerevisiae [25]. We selected 10 transcription factors and its corresponding TFBSs which are kept in their original genomic sequence, and test PSOMF for finding TFBSs with two test method. The specification of these test data sets are given in Fig. 4. The obtained results of PSOMF are compared with the three known programs YMF version 1.0 [11], MEME version 3.5.4 [3] and AlignACE version 3.0 [15].

The comparison of three algorithms are performed based on the following measurements: nucleotide Correlation Coefficient (nCC), nucleotide Sensitivity (nSn), nucleotide Specificity (nSp), nucleotide Performance Coefficient (nPC), and Accuracy (Acc) [9]. The definition of these measurements are given in Fig. 5. In this figure, the variable TP, TN, FP and FN denote the number of correctly predicted positive nucleotides, correctly predicted negative nucleotides, falsely predicted positive nucleotides and falsely classified negative nucleotide, respectively [26]. All of these measurements are previously defined in [9] as suitable measurements for comparison of motif finding tools.

Data set	ℓ	t	n
CPF1	7	3	850
GCR1	5	6	850
MATA1	25	3	850
PHO2	19	3	850
PHO4	6	5	850
RAP1	7	16	850
ROX1	12	3	850
SFF	10	3	850
STE12	7	4	850
UIS	11	3	850

Fig. 4. Specification of the data sets.

Measurement	*Formula*
nCC	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TN+FN)\times(TN+FP)\times(TP+FN)\times(TP+FP)}}$
nSn	$\frac{TP}{(TP+FN)}$
nSp	$\frac{TN}{(TN+FP)}$
nPC	$\frac{TP}{(TP+FP+FN)}$
Acc	$\frac{(TP\times TN)}{(TN+FN+TP+FP)}$

Fig. 5. Comparison measurement formulas.

For performing PSOM we consider $m = 20$ and $N = 5$. In the first test method, the four data sets GCR1, PHO4, RAP1, and STE12 are selected, and tools MEME, AlignACE, and PSOM are performed on them. In Fig. 6, the five measures defined in Fig. 5 are shown for the mentioned 4 regulons predicted by the MEME, AlignACE, and PSOMF. In this test, YMF is not mentioned, because of its similar result on the data sets with MEME and AlignACE. In the second test method, the result obtained by YMF, MEME, and AlignACE on the 6 data sets CPF1, MATA1, PHO2, ROX1, SFF, and UIS are extracted from [11]. In [11], nPC measurement is used for comparsion. PSOM is performed on all of these 6 data sets and the results of nPC measurement are compared with the extracted results from [11]. The comparison results are shown in Fig. 7.

As we can see, our algorithm shows a higher score than the other three algorithms in both test methods.

4.1 Conclusion

In this paper, an algorithm is presented for motif finding in a given set of sequences, based on particle swarm optimization. The binary system is used for motif representation. Particle swarm optimization in binary implementation is

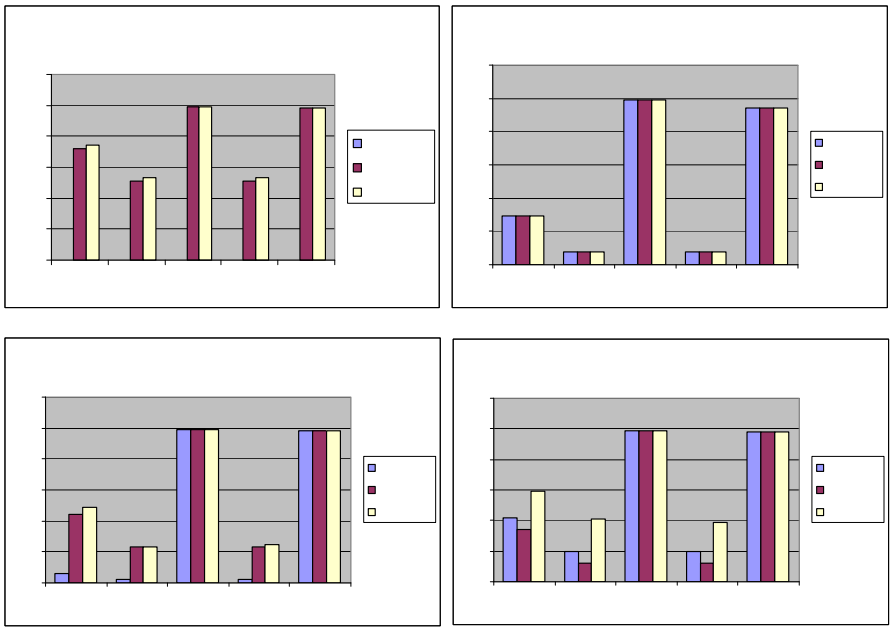

Fig. 6. Results of the first test method.

data set	YMF	MEME	AlignACE	PSOMF
CPF1	0.62	0.49	0.02	0.69
MATA1	0.19	0.20	0.11	0.35
PHO2	0.00	0.00	0.00	0.13
ROX1	0.00	0.03	0.00	0.33
SFF	0.00	0.00	0.05	0.29
UIS	0.01	0.43	0.20	0.44

Fig. 7. Results of the second test method.

capable of solving this problem rapidly. Unlike evolutionary algorithms, PSO has memory of past successes and has a tendency to converge upon regions of the search space that have been successful previously. For this reason it converges so rapidly in comparison with evolutionary algorithm such as genetic algorithms. The algorithm uses joint information content as a fitness function. The results are compared with three algorithms YMF, MEME and AlignACE. The effectiveness of our method is shown by the obtained results.

References

[1] Pavesi, G., Mauri, G., Pesole, G.: An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics* **17** (2001) 207–214

[2] Pevzner, P.: *Computational Molecular Biology: An Algorithmic Approach.* The MIT Press, Massachusetts (2000)

[3] Bailey, T., Elkan, C.: The value of priori knowledge in discovering motifs with MEME. In: *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Menlo Park, CA (1995) 21–29

[4] Eskin, E., Pevzner, P.: Finding composite regulatory patterns in DNA sequences. *Bioinformatics* **18** (2002) 354–363

[5] GuhaThakurta, D., Stormo, G.: Identifying target sites for cooperatively binding factors. *Bioinformatics* **17** (2001) 608–621

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

[7] Lawrence, C., Altschul, S., Bogusky, M., Liu, J., Neuwald, A., Wootton, J.: Detecting subtle sequence signals: Gibbs sampling strategy for multiple alignment. *Science* **262** (1993) 208–214

[8] Stormo, G.: DNA binding sites: representation and discovery. *Bioinformatics* **16** (2000) 16–23

[9] Tompa, M., Li, N., Bailey, T., Church, G., De Moor, B., Eskin, E., Favorov, A., Frith, M., Fu, Y., Kent, W., Makeev, V., Mironov, A., Noble, W., Pavesi, G., Pesole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., van Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., Zhu, Z.: Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* **23** (2005) 137–144

[10] Paul, T., Iba, H.: Identification of weak motifs in multiple biological sequences using genetic algorithm. In: *Proceedings of the 8th annual conference on Genetic and Evolutionary computation*, ACM Press, New York, NY (2006) 271–278

[11] Sinha, S., Tompa, M.: YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res.* **31** (2003) 3586–3588

[12] Jonassen, I., Collins, J., Higgins, D.: Finding flexible patterns in unaligned protein sequences. *Protein Sci.* **4** (1995) 1587–1595

[13] Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm. *Bioinformatics* **14** (1998) 55–67

[14] Thijs, G., Lescot, M., Marchal, K., Rombauts, S., De Moore, B., Rouze, P., Moreau, Y.: A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling. *Bioinformatics* **17** (2001) 1113–1122

[15] Hughes, J., Estep, P., Tavazoie, S., Church, G.: Computational identification of cis-regulatory elements associated with functionally coherent groups of genes in Saccharomyces cerevisiae,. *J. Mol. Biology* **296** (2000) 1205–1214

[16] Liu, X., Brutlag, D., Liu, J.: Bioprospector: Discovering conserved DNA motif in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomput.* **6** (2001) 127–138

[17] Che, D., Song, Y., Rashed, K.: MDGA: Motif discovery using a genetic algorithm. In: *Proceedings of Genetic and Evolutionary Computation*, ACM Press, New York, NY (2005) 447–452

[18] Fogel, G., Weekes, D., Varga, G., Dow, E., Harlow, H., Onyia, J., Su, C.: Discovery of sequence motifs related to co-expression of genes using evolutionary computation. *Nucleic Acids Res.* **32** (2004) 3826–3835

[19] Gertz, J., Riles, L., Turnbaugh, P., Ho, S., Cohen, B.: Discovery, validation, and genetic dissection of transcription factor binding sites by comparative and functional genomics. *Genome Res.* **15** (2005) 1145–1152
[20] Liu, F., Tsai, J., Chen, R., Shih, S.: FMGA: Finding motifs by genetic algorithm. In: *Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*, IEEE Computer Society Press, Los Alamitos, CA (2004) 459–446
[21] Stine, M., Dasgupta, D., Mukatira, S.: Motif discovery in upstream sequences of coordinately expressed genes. *Evol. Comput.* **3** (2003) 1596–1603
[22] Wei, Z., Jensen, S.: GAME: detecting cis-regulatory elements using a genetic algorithm. *Bioinformatics* **22** (2006) 1577–1584
[23] Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: *Proceeding 6th International Symposium on Micro Machine and Human Science*, IEEE Computer Society Press, Los Alamitos, CA (1995) 39–43
[24] Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: *Proceeding of the IEEE International Conference on System, Man, and Cybernetics*, IEEE Computer Society Press, Los Alamitos, CA (1997) 4104–4108
[25] Zhu, J., Zhang, M.: SCPD: a promoter database of yeast saccharomyces cerevisiae. *Bioinformatics* **15** (1999) 563–577
[26] Benitez-Bellon, E., Moreno-Hagelsieb, G., Collado-Vides, J.: Evaluation of thresholds for the detection of binding sites for regulatory proteins in Escherichia coli k12 DNA. *Genome Biology* **3** (2002) 1–16