An Improved Immune Genetic Algorithm for Weak Signal Motif Detecting Problems

Xun Wang
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, 3050006, Ibaraki, Japan

Zhongyu Wang
Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

Tao Song
Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

ABSTRACT
Motif detecting in DNA sequences is one of the most popular tasks in computational biology, which is important for people to understand functions of genes. Recently, the motif detecting problem was abstracted as a planted (l,d)-motif problem and many instances of the problem have been proposed as challenges for motif detecting algorithms. In this work, we propose an improved immune genetic algorithm, called MRPIGA, to solve a class of specific planted (l,d)-motif problems, weak signal motif problems, in which a modified random projection strategy is applied to generate a good initial population of candidate solutions. Experimental results on stimulated data show that MRPIGA performs better than Random Projection, GARPS and MDGA. We also test the MRPIGA on five groups of realistic biological data. It shows that the MRPIGA performs superior to detect motifs.

General Terms
Bioinformatics, Computational Biology, Algorithm, Evolution Optimization

Keywords
Motif detecting, Weak signal motif, Random projection, Immune genetic algorithm

1. INTRODUCTION
Genes are known as the basic fundamental units containing inherited information in deoxyribonucleic acid (DNA), which can be used as templates for the transcription of proteins. The process of protein transcription is called gene expression. It is a basic function of lives and begins with binding the transcription factor proteins to specific locations of genomic sequence. To the knowledge of biologists, the binding locations usually share common patterns of DNA segment with few variable nucleotides. These common DNA segments are known as motifs. Detecting motifs and their binding locations are two major tasks in molecular biology. Several experimental methods, such as DNase foot printing [1] method and gel shift assay method [2], are feasible in laboratories. However, with the development of high-throughput sequencing technology in recent years, huge amount of DNA sequences data can be obtained in once sequencing experiment. Detecting motifs by experimental methods will be labor-intensive, time consuming and expensive. Recently, many computer algorithms have been proposed to detect motifs. For computer scientists, the motif detecting problem was formulated as the planted (l,d)-motif problem [3].

The Planted (l,d)-Motif Problem: Let M be a fixed but unknown nucleotide sequence of length l. Suppose that M occurs once in each of N background sequences of common length L, but that each occurrence of M is corrupted by exactly d substitutions in positions chosen independently at random. Given the N sequences, recover the locations of motif occurrences and the consensus M.

Computationally, the difficulty of the planted (l,d)-motif problem arises from the fact that locations of the motifs can vary significantly on each background sequences. The planted (l,d)-motif problem have been proved to be **NP**-complete. Some instances of such problems are taken as challenges to test motif detecting algorithms. Till now, many algorithms have been proposed to address the problems, and all the algorithms can be divided into two categories. (1) Exact algorithms, such as CENSUS [4], MITRA [5], PMS1 [6], PMSP [7] and SMILE [8], which always come up with the correct answer(s) to the problem. For small scale planted (l,d)-motif problems, exact algorithms are practical and feasible to detect all the potential motifs, but when l and d are assigned with large values, exact algorithms will take an unacceptable time to exhaustively enumerate all the candidate solutions. (It is due to the fact that the solution space of the planted (l,d)-motif problem increases exponentially with the increment of l and d.) (2) Approximate algorithms were also designed to solve the planted (l,d)-motif problem, which can significantly reduce computation time or space for motif detection. The approximate algorithms, such as Random Projection method [3], MEME [9], BioProspector [10] and BioOptimizor [11], give the correct answers to the problem with a certain rate. This rate is called accuracy rate of the algorithm. However, the approximate algorithms will become trapped in a local maximum, and failed in some challenge instances of the problem (associated with low accuracy rate). To avoid the local maximal trap, genetic algorithm (GA) was recently introduced to improve the performance of the approximate algorithms. Several GA based methods, such as GAME [12], MDGA [13] and GARPS [14], performs well to solve the problem, but failed in some weak signal motif detecting problems. Recently, immune genetic algorithm (IGA) was introduced to detect motifs in DNA sequences [15], and achieved superior performances to other algorithms. IGA adopts concentration regulation mechanism to maintain the population diversity and vaccine mechanism to inhibit the degeneracy. The initial population of IGA is randomly chosen as general GA. For large scale planted (l,d)-motif problems, randomly generated initial population will rarely come close

to an optimal solution. It will yield to a long computation time for IGA to detect the motifs.

In this work, we introduce an improved IGA to detect motifs from DNA sequences, where a modified random projection strategy is used to generate a good initial population. The initial population generated by modified random projection strategy is more likely to cover an optimal solution of the problem than the randomly generated population. Our algorithm is called MRPIGA. We test MRPIGA on simulated data to solve the weak signal motif problems. Experimental results show that the performances of our method are better than Random Projection, GARPS and MDGA. Five groups of realistic biological data, including preproinsulin, DHFR, metallothionein, c-fos, and Yeast ECB, which have been successfully detected in laboratories, are chosen to test MRPIGA. The motifs detected by MRPIGA the same as in [17].

2. CLASSIFICATION OF THE PLANTED (l, d)-MOTIF PROBLEMS

As stated in [16], the planted (l, d)-motif problem models the motif detecting problem as a combinational optimization model. In the model, if the consensus is completely conserved in any background sequence, then the motif detecting problem can be reduced to the problem of counting the sequences with the maximal appearance in the background sequences. If there are a sufficient number of substitution points in the motif, then motif detecting problem becomes very hard, since the signal of the conserved positions in the motifs can be covered by the noise from the background sequences. The probability analysis of the planted (l, d)-motif problem has been done in [16], as well as an expectation number $E(l, d)$ of counting motifs with at least one variant in each background sequence is defined as:

$$E(l,d) = 4^l(1-(1-p_d)^{(n-l+1)})^t,$$

where p_d is the probability that a given l-mer occurs with d substitutions as certain position in a random DNA sequence, n is the length of background sequence, and t is the number of background sequences. According the expectation number $E(l, d)$, i.e., signal intensity, the instances of the planted (l, d) -motif problem can be categorized into three classes: tiny signal motifs, faint signal motifs and weak signal motifs [14], as shown in Table 1. In this work, we deal with the class of weak signal motif problems.

Table 1. The classification of instances of the planted (l, d)-motif problem

Tiny Signal Motifs	Faint Signal Motifs	Weak Signal Motifs
(8,2)	(9,2)	(10,2),(11,2)
(10,3)	(11,3)	(12,3), (13,3)
(12,4)	(13,4)	(14,4), (15,4)
(14,5)	(15,5)	(16,5), (17,5)
(16,6)	(17,6)	(18,6), (19,6)

3. METHODS

3.1 Modified random projection strategy

In [3], the random projection strategy was used to generate potential motifs from background sequences. A position set p with k positions was randomly chosen from l positions, and then each l-mer in the background sequences can be hashed into a corresponding bucket based on projection at the k selected positions. The l-mers enriched in certain bucket were taken as potential motifs for further refining.

	1	2	3	4	5	6	7	8	9
r_1	A	T	A			T	C		
r_2	T					T	C	G	A
r_3	A			G	T	T	C		
r_4			T		C	C		T	A
t_1	A	T	T	G	C	T	C	T	A
t_2	A	T	A	G	T	T	C	G	A

Fig. 1: An example of reforming potential motifs from qualified projections

We consider here a modified random projection strategy. Let $p = \{p_1, p_2, \ldots, p_k | 1 \le p_1 < p_2 < \cdots < p_k \le l\}$ be a set of positions with k integers, and $s = s_1 s_2 \ldots s_l$ be an l-mer. We denote the projection of s on p by $Proj(s|p)$, which is the concatenation of the nucleotides at the k positions of s. For a given set of positions p and N background sequences, a number of k-mer projections can be obtained, and each of the projections appear at least once in one of the N background sequences. It is possible that certain projection appears more than one times out of the N background sequences. The projection appearing for significant times is called qualified projection. By repeatedly choosing position set p at random for several times, a set of qualified projections P and the set of their positions Q can be obtained. Such a set of qualified projections P will be likely the enriched conserved patterns in planted motifs. We can refine a set of potential motifs from P and Q. For each potential motif, a profile probability is associated, which is defined as

$$Prob(t) = \prod_{j=1}^{l} prob_j.$$

$prob_j$ is the probability on the jth position being certain nucleotide in these qualified projections. The refined potential motifs associated with significant profile probabilities (greater than a threshold θ) are chosen to make up a qualified motifs set T. An example of refining qualified motifs from qualified projections is given in Fig. 1, in which $r_1 = ATATC$, $r_2 = TTCGA$, $r_3 = AGTTC$, $r_4 = TCCTA$ are qualified projections with $p_1 = \{1,2,3,6,7\}$, $p_2 = \{1,6,7,8,9\}$, $p_3 = \{1,4,5,6,7\}$, $p_4 = \{3,5,6,8,9\}$ and $k = 5, l = 9$. Two qualified motifs $t_1 = ATTGCTCTA$ and $t_2 = ATAGTTCGA$ are obtained according to the threshold $\theta = 1/16$

3.2 Immune genetic algorithm

Immune genetic algorithm (IGA) is a modified genetic algorithm, which is inspired from the biological evolution and immune function. In IGA, by evolution mechanism, the whole population can achieve a significant fitness score, and by immunization mechanism, adaptability of the individuals can be improved through the effect of the vaccine. In immune systems, concentration regulation mechanism is also used to keep the diversity of population. The flowchart of IGA is shown in Fig. 2.

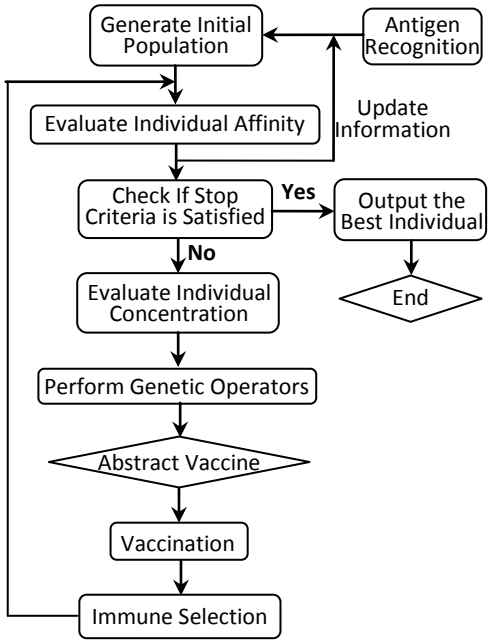

Fig. 2: The flowchart of IGA

3.3 Detecting motifs by MRPIGA

We propose an improved IGA, called MRPIGA, to detect motifs, in which the modified random projection strategy introduced in Section 3.1 are used to generate a good initial population and antigens are taken as target fitness function and antibodies as solutions. As classic IGA, the concentration mechanism is applied to maintain population diversity and vaccine mechanism is used to inhibit degeneracy during the evolution of generations in MRPIGA.

In motif detecting problems, motif can be represented as a Position frequency matrix (PFM) by aligning the motifs. By PFM_p and PMF_b, we denote motif and background PFMs, respectively. The fitness of an antibody a can be defined as:

$$f(a) = (IC_p - IC_b)/l,$$

in which IC_p and IC_b are IC scores on matrixes PFM_p and PMF_b, respectively. The IC score on certain PFM is

$$IC = \sum_{j=1}^{\omega} \sum_{b} f_b(j) log(f_b(j)/p_b),$$

where $f_b(j)$ is the normalized frequency of nucleotide b on column j of a matrix PFM, p_b is the frequency of the same nucleotide in background sequences and l is the length of motifs. The affinity of an antibody a with the antigen is defined as $g(a) = f(a)/\sum_{b \in B} f(b)$, where B is the antibody population. For two PFMs, M and N, we define a scoring function for column-to-column comparisons:

$$score = \sqrt{\sum_{b \in \{A,T,C,G\}} (M_b - N_b)^2/2},$$

where M_b and N_b are the values of nucleotide b in the compared pair of columns in two different PFMs. The affinity of antibodies i and j is $Ab_{ij} = F/l$, where F is the total score of all positional comparisons of i and j, and l is the length of the antibody. The value of Ab_{ij} ranges from 0 to 1, with value 0 indicating total dissimilarity and value 1 indicating complete similarity. The concentration of an antibody is the ratio of similar antibodies to the antibody. With the evolution of individuals, the diversity of the population decreases due to the increasing concentration of antibodies which are associated with higher fitness scores. By genetic operators, we can choose antibodies associated with high concentration in low probability. The concentration of an antibody a is $C_a = 1/m \sum_{j=1}^{m} Ac_{aj}$ with

$$Ac_{aj} = \begin{cases} 1, & Ab_{aj} \geq \lambda; \\ 0, & \text{otherwise} \end{cases},$$

where Ab_{aj} is the similarity between antibody a and j, and λ is a predefined threshold. Based on the concentration of the antibody a, its selection probability is defined as:

$$e_a = \alpha \times Ag_a + (1 - \alpha) \frac{1}{N} e^{-C_a/\beta},$$

where α and β are two predefined thresholds. It is not hard to find that the higher value C_a of the antibody a is, the lower its selection probability is, and also the larger the fitness of the antibody a is, the higher its selection probability is. According to the selection probability, we can select antibodies with high fitness and low concentration to improve the diversity of the population during evolution.

The MRPIGA starts with the initial population generated from the qualified motifs set T, and performs genetic operators, mutation and crossover with probabilities p_c and p_m, on the selected antibodies to generate new population. In MRPIGA, immune operators are also used. When new population is generated, the immune operator, called vaccine mechanism, is used to inhibit the degeneracy of the new population. The immune operation is completed in two steps: a vaccination (to increase the fitness) and an immune selection (to prevent the degeneracy of the population). MRPIGA is capable of fast convergent to the best solution of the motif, and its pseudo-code is described as follows.

Algorithm MRPIGA

Input: $l, d, k, L, S_1, S_2, \ldots, S_N, m, \phi, \theta, P_0, p_m, p_c. \alpha, \beta, \lambda, PFM_p, PFM_b$
Output: The best individual of antibody
1: $P \leftarrow \emptyset$, $Q \leftarrow \emptyset$, T$\leftarrow \emptyset$
2: **for** $h \leftarrow 1$ **to** m **do**
3: $p_h \leftarrow (p_1, p_2, \ldots, p_k)$ generated at random
4: **for** $i \leftarrow 1$ **to** N
5: **for** $j \leftarrow 1$ **to** $L - l + 1$
6: $s \leftarrow j$th subsequence with length l in S_i
7: $r_h = Proj(s|p_h)$ appearing more than ϕ times
8: $P \leftarrow r_h, Q \leftarrow p_h$
9: **end for**
10: **end for**
11: **end for**
12: **for** $i \leftarrow 1$ **to** l **do**
13: Refine a set of potential motifs from P and Q
14: $T \leftarrow$ qualified motifs with profile probability greater than θ
15: **end for**
16: Generate initial population of antibodies P_0 from T
17: Perform antigen recognition in P_0
18: Evaluate individual affinity
19: Update the information to recognize antigen
20: **while** the stop criteria is not satisfied **do**
21: Evaluate the concentration of any antibody a by e_a
22: Evaluate the fitness of any antibody a by $f(a)$

23: Perform tournament selection according to e_a and $f(a)$
24: Perform mutation operators in probability p_m
25: Perform crossover operators in probability p_c
26: Generate a new generation
27: Abstract vaccine from the new generation
28: Do the vaccination operation
29: **end while**
30: **return** the best individual

In algorithm MRPIGA, l is the length of motifs; d is the number substitutions in motifs; k is the length of projection; $L$is the length of background sequences; $S_1, S_2, \ldots, S_N$ are the N background sequences; m is the number of iterating times to generate projections; ϕ is the threshold to select qualified projections; θ is the threshold of profile probability to select qualified motifs; PFM_p and PFM_b are PFMs of motifs and background sequences; p_m and $p_c$are probabilities associated with mutation and crossover operators; α and β are thresholds used to calculate the selection probability e_a of an antibody a. P stores qualified projections, Q stores positions of qualified projections, T stores qualified motifs, and P_0 denotes the initial population generated from T.

Line 1 initializes the set of P, Q and T to be empty set. Lines 2-11 literately choose k positions at random for m times. The obtained qualified projections are stored in P, and their positions are stored in Q. Lines 12-15 refine a set of potential motifs. The ones with significant profile probability (greater than θ) are chosen as qualified motifs and stored in T. Line 16 generates a initial population P_0 from T. Lines 17-19 find antigen in P_0 and evaluate affinity of each individual. Based on the evaluation, the information to recognize antigen is updated. Line 20 checks if the stop criterion is satisfied. If so, then the algorithm stops and returns the best individual. If the stop criterion is not satisfied, then lines 21-28 do the evolution process. It firstly evaluates the concentration and fitness of each antibody, and then performs genetic operators, mutation and crossover, on the significant ones to generate a new population, from which vaccine is abstracted. After doing the vaccination operation, it jumps to line 20 to check if the stop criterion is satisfied to determine whether to do the **while** loop or not.

4. RESULTS

4.1 Experiments results on simulated data

We test MRPIGA on a specific class of the planted (l, d)-motif problems, weak signal motifs problems. 20 background sequences with length 600 are randomly generated as the input instance, as well as an l-mer consensus M is generated at random. The consensus M is randomly planted in each background sequence with d positions of substitutions chosen at random. The length of projection is $k = 7$. Projections appearing greater than $\phi = 30$ times are selected as qualified ones. The potential motifs with profile probabilities bigger than $\theta = 0.6^l$ make up the set of qualified motifs T. The probabilities of genetic operators crossover and mutation are $p_c = 0.03$ and $p_m = 0.01$, respectively. The threshold values are $\alpha = 0.5, \beta = 0.1$ and $\lambda = 0.8$.

For different instances of the weak signal (l, d)-motif problems, we choose distinguish m, which are shown in the last column of Table 2. For small scale problems, m is associated with smaller value. This is due to the fact that if we use large m in small scale problems, the noise from background sequences will cover the genetic information in qualified potential motifs. It can make initial population of the IGA have no difference comparing with randomly generated initial population. But for large scale problems, we should use large m to strengthen the genetic signal in those qualified potential motifs to give a good start for IGA to improve its performance.

Let K be the set of known motif positions in the instances, and W be the set of predicted positions obtained by motif detecting algorithms. By $|K \cap W|/|K \cup W|$, we denote the performance coefficient (PC). We run MRPIGA for 50 times for each weak signal motif problem (WSMP), and take the average value of PC over all 50 times. The average value of PC of Random Projection (RP), GARPS, MDGA and MRPIGA are shown in Table 2, where the corresponding results of Random Projection, GARPS and MDGA can be referred to [3,13,14].

Table 2. Experimental results on simulated data

WSMP	RP (PC)	GARPS (PC)	MDGA (PC)	MRPIGA (PC)	m
(10,2)	0.82	0.906	0.876	0.945	50
(11,2)	0.91	0.992	0.921	0.988	50
(12,3)	0.81	0.805	0.762	0.861	100
(13,3)	0.92	0.963	0.870	0.969	100
(14,4)	0.77	0.873	0.798	0.904	600
(15,4)	0.93	0.947	0.904	0.950	200
(16,5)	0.70	0.725	0.688	0.734	600
(17,5)	0.93	0.947	0.879	0.955	600
(18,6)	0.74	0.814	0.800	0.837	600
(19,6)	0.96	0.994	0.866	0.976	600

As indicted in the column "MRPIGA (PC)" in Table 2, MRPIGA achieves superior performance coefficient (PC) to Random Projection, GARPS and MDGA to solve several instances of weak motif problems, except for (11,2) and (19,6). Although Random Projection and GARPS performs better than MRPIGA to solve (11,2) and (19,6) problems, MRPIGA can also achieve significant accuracy rates.

IGA is known to detect motifs well for its global searching strategy, and can avoid trapping in local optimal solution. By using the modified random projection method, we can generate a better initial population for IGA to improve its performance in detecting motifs.

4.2 Experimental results on real data

We use MRPIGA to detect realistic biological motifs from a test set of DNA sequences, which has been used to test existing algorithms [17]. We detect realistic biological motifs, preproinsulin, DHFR, metallothionein, c-fos, and Yeast ECB by MRPIGA. These motifs have been experimentally detected in laboratories. The problems of detecting realistic biological motifs are firstly reformulated as corresponding (l, d)-motif problems, and then we use MRPIGA to detect these biological motifs. The obtained motifs are the same as the motifs published in [17]. The results are shown in Table 3.

5. FINAL REMARKS

In this work, we introduced an improved immune genetic algorithm to detect motifs, in which to avoid the defect of randomly generating initial population, a modified random projection strategy is proposed to provide a good initial population for further refine. The algorithm is called MRPIGA. In MRPIGA, by evolution mechanism, the whole population can achieve a significant fitness score, and by immunization mechanism, adaptability of the individuals can be improved through the effect of the vaccine. We test MRPIGA on several instances of weak motif detecting problem. MRPIGA achieves superior performance coefficient to Random Projection, GARPS and MDGA, except for instance (11,2) and (19,6). MRPIGA can also achieve significant accuracy rates to solve them. In realistic biological data test, we use MRPIGA to detect biological motifs, including preproinsulin, DHFR, metallothionein, c-fos, and Yeast ECB, which have been successfully determined in laboratories. MRPIGA performs well in detecting realistic motifs, obtains the same motifs as published in previous literatures.

Table 3. Experimental results on realistic biological data

Motifs	Detected Planted Motif	Published Motif	(l,d)-motif problem
preproinsulin	$TTGCCTCAGCCCCTTA$	$CCTCAGCCCC$	(16,3)
DHFR	$ATATCGTGGGCA$	$ATTTCnnGCCA$	(12,2)
metallothionein	$CTCTGC$**ACAC**$CGGCCCATA$	TGC**RCY**CGG	(19,4)
c-fos	$CCATATTGCT$	$CCATATTAGAGACTCT$	(10,2)
Yeast ECB	$CCCATTAAGGAAA$	$TTTCCCnnTnAGGAAA$	(13,3)

6. REFERENCES

[1] Galas D.J., Schmitz A., 1978. A DNA footprinting: a simple method for the detection of protein-dna binding specifility, Nuleic Acids Res, 5, 9, (1978), 3157-3170.

[2] Garner M.M., Revzin A., 1981. A gel electrophoresis method for quantifying the binding of protein to specific DNA regions: application to components of the escherichia coli locates operon regulary system. Nucleic Acids Research, 9, 13, (1981), 3047-3060.

[3] Buhler J., Tompa M., 2001. Finding motifs using random projections. Journal of Computational Biology, 9, (2001), 225-242.

[4] Evans P., Smith A., 2003. Toward optimal motif enumeration. In Proceedings of Algorithms and Data Structures of Eighth International Workshop, WADS2003}, (2003), 47-58.

[5] Eskin E., Pevzner P., 2002. Finding composite regulatory patterns in DNA sequences. Bioinformatics S1, (2002), 354-363.

[6] Rajasekaran S., Balla S, Huang C.H., 2005. Exact algorithms for the planted motif problem. Journal of Computational Biology, 12, 8, (2005), 1117-1128.

[7] Davila J., Balla S., Rajasekaran S., 2006. Space and time efficient algorithms for planted motif search. In Proceedings of the Second International Workshop on Bioinformatics Research and Applications (IWBRA 2006), (2006), 822-829.

[8] Marsan L., Sagot M.F., 2000. Extracting structured motifs using a suffix-tree. Algorithms and Application to Promoter Consensus Identification. In Proceedings of RECOMB2000, Tokyo, (2000), ACM Press.

[9] Bailey T.L., Elkan C., 1994. Fitting a mixture model by expectation maximization to discover motif in biopolymers. In Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, Menlo Park, CA, (1994), 28-36, AAAI Press.

[10] Liu X., Brutlag D.L., Liu J.S., 2001. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes, Pacific Symposium on Biocomputing, 6, (2001), 127-138.

[11] Liu F.F.M., Tsai J.J.P., Chen R.M., Chen S.N., Shih S.H., 2004. FMGA: finding motifs by genetic algorithm, In Proceedings of Fourth IEEE Symposium on Bio-informatics and Bioengineering, BIBE 2004, (2004), 459-466.

[12] Wei Z., Jensen S.T., 2006. GAME: detecting cis-regulatory elements using a genetic algorithm, Bioinform atisc, 22, 13, (2006), 1157-1184.

[13] Che D., Zhao H., Song Y., 2005. MDGA: motif discovery using a genetic algorithm. In Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO 2005), (2005), 447-452.

[14] Huo H., Zhao Z., Stojkovic V., Liu L., 2010. Optimizing genetic algorithm for motif discovery, Mathematical and Computer Modeling, 52, (2010), 2011-2020.

[15] Luo J.W., Wang T., 2010. Motif discovery using an immune genetic algorithm, Journal of Theoretical Biology,(2010), 264, 2, 319-325.

[16] Pevzner P., Sze S.H., 2000. Combinatorial approaches to binding subtle signals in DNA sequences. In Proceedings of Eighth International Conference on Intelligent Systems for Molecular Biology, (2000), 269-278.

[17] Rajasekaran S., Balla, S, Huang C.H., 2005. Exact algorithms for the planted motif problem, Journal of Computational Biology, 12, 8, (2005), 1117-1128.