A Genetic Algorithm with Clustering for Finding Regulatory Motifs in DNA Sequences

Shripal Vijayvargiya
Department of Computer Science & Engineering,
Birla Institute of Technology Ext. Center, Jaipur
Rajasthan, India

Pratyoosh Shukla
Department of Biotechnology,
Birla Institute of Technology, Mesra, Ranchi,
Jharkhand, India

ABSTRACT

Identification of Transcription Factor Binding Sites (TFBS) also called as motifs, from the promoter region of genes remains a highly important and unsolved problem of computational biology. Motifs are short, recurring patterns in DNA sequences that are presumed to have a biological function. In this paper, we propose an evolutionary approach to identify transcription factor binding sites. This approach is based on the genetic algorithm with population clustering. A simple genetic algorithm favors selection of fittest, and this selective pressure tends to remove the diversity of population. Sometimes promoter sequences of some genes consists multiple motifs that also need to be identified. The proposed algorithm uses clustering scheme to partition population in clusters and the mating is allowed only within cluster. This scheme enables algorithm to retain diversity of population over the generations, against the selection pressure and to find out multiple motifs in promoter sequences of co-regulated genes. We applied this approach on various data sets and the results show that it can find correct results for binding sites.

General Terms: Genetic algorithm, DNA sequences, motif

Keywords: motif, transcription factor, regulatory binding sites, genetic algorithms and clustering

1. INTRODUCTION

Over the years, to understand the biological activities in various organisms many genome-sequencing projects are completed by biologists. These genome sequencing projects provided the full map of gene locations on chromosomes but tell us very little about how, when and why particular genes are expressed, and which interactions of genes are correlated with human disease. To know about this we need to understand the gene expression mechanism. Understanding the process that regulates gene expression and identification of those regulating element is a major challenge of biology. The main idea in gene expression is that every gene contains the information to produce a protein. Gene expression begins with binding of multiple protein factors, known as transcription factors (TF), to enhancer and promoter sequences. Transcription factors regulate the gene expression by activating or inhibiting the transcription machinery. These transcription factors binding sites are called motifs. A motif is very small in length, generally of 4-8 base pair but it may be longer than this. Also there may be multiple motifs present in a promoter sequence. Co – expressed genes are expressed as a group due to the interaction of a TF protein or set of proteins. Thus identification of regulatory regions and binding sites is a prerequisite for understanding gene regulation [1] [2]. Experimental identification and verification of such elements is challenging and costly, so much effort has been put into the development of computational approaches.

Computational discovery of regulatory elements is mainly possible because they occur several times in the same genome, and because they may be evolutionary conserved. This means that searching for over represented motifs across regulatory regions may discover novel regulatory elements. From the computation point of view the motif finding problem can be formulated as follows: given a set of sequences, find an unknown pattern that occurs frequently. If a pattern of *m* letters long appears exactly in every sequence, a simple enumeration of all *m*-letter patterns that appear in the sequences gives the solution. But this simple looking problem is complicated because of evolutionary events like mutations, insertions and deletions.

Motifs or TFBSs are generally represented as consensus IUPAC strings, position frequency matrices (PFMs), position weight matrices (PWMs), or position specific scoring matrices (PSSMs) in databases. Commonly, motifs or TFBSs in non-coding DNA sequences are conserved but still tend to be degenerate, which can influence the interaction between TFs and motifs or TFBSs. Therefore, after motif or TFBS data are collected and aligned from experimental or computational results, relevant consensus IUPAC strings can be constructed by selecting a degeneracy base pair symbol for each position in the alignment. The motif or TFBS data can also be modeled as PFM by aligning identified sites and counting the frequency of each base pair at each position of the alignment. Moreover, by using sequence logos, PWM can be displayed with color and height proportional to the base pair frequency and information content for each position by formulas. Known regulatory motif profiles are cataloged in databases such as TRANSFAC [3] and JASPAR [4].

We used a population clustering genetic algorithm for regulatory motif discovery. The algorithm uses clustering scheme to partition search space thus enabling algorithm to retain diversity of population against the selection pressure and to identify multiple significant motifs. In section 2 we described the computational aspect of problem. Section 3 contains a brief survey of various techniques and algorithms used to solve the motif-finding problem. Section 4 explains the method and it's components like representation, fitness score function, selection, crossover, mutation operators and clustering scheme. Next section contains the simulation results followed by conclusion.

2. PROBLEM STATEMENT

The TFBS identification in unaligned DNA sequences using GAs can be defined as follows [5]:

Given a set of N sequences S ={S_1, S_2, . . ., S_N}, each of which is from the finite alphabet D ={A, T, C, G}, where the length of each sequence is *l*, and the motif width *w* with a valid constraint 0 < *w* << *l*. Find a set of instances M = {m_1, m_2, . . ., m_N} where each m_i is a subsequence with length *w* from sequence S_i, such that the sum of information content IC, as proposed by Stormo, is maximized [6]

$$IC = \sum_{j=1}^{w} \sum_{b} f_b(j) \log \frac{f_b(j)}{p_b} \qquad \text{... (1)}$$

here $f_b(j)$ is the normalized frequency of nucleotide b Є D on the column j of all instances in M and p_b is the background frequency of the same nucleotide from S.

3. EXISTING METHODS

Identification of regulatory motifs in upstream region of co-regulated genes or orthologous genes is an unsolved problem of computational biology. In last few years many algorithms were proposed to find solutions for motif discovery. According to survey [7] two major strategies exist to discover repeating sequence patterns occurring in both DNA and protein sequences: enumeration and probabilistic sequence modeling. Enumeration strategies rely on word counting to find words that are over-represented. Probabilistic model-based methods represent the pattern as a matrix, called a motif, consisting of nucleotide base multinomial probabilities for each position in the pattern and different probabilities for background positions outside the pattern. Among those previous works, most popular one is the Multiple Em for Motif Elicitation (MEME) system [8], Gibbs sampler [9] and CONSENSUS [10]. Even with weak signals, methods such as MEME and Gibbs Motif Sampler effectively find motifs of variable width and occurrences in DNA and protein sequences.

Many other algorithms have been developed to improve these popular motif discovery tools by means of performance, length of motifs or some other considerations. Liu et.al employed genetic algorithm for finding potential motifs in the regions of transcription start site (TSS) [11]. Structured genetic algorithm is used to search and to discover highly conserved motifs amongst upstream sequences of co-regulated genes [12]. Structured genetic algorithm is also used to identify variable length motifs [13].

Recently Algorithms based on promoter sequences of co-regulated genes and phylogenetic footprinting have been suggested. These algorithms integrate two important aspects of a motif's significance, i.e., overrepresentation and cross-species conservation, into one probabilistic score. Based on the consensus algorithm Wang and Stormo developed the motif finding algorithm PhyloCon (Phylogenetic Consensus) [14] that takes into account both conservation among orthologous genes and coregulation of genes within a species. Sinha et al. developed the algorithm PhyME [15] based on a probabilistic approach that handles data from promoters of coregulated genes and orthologous sequences.

4. PROPOSED METHOD

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

4.1 Clustering

The selection procedure in a simple genetic algorithm favors the fittest one. After few generations this selective pressure tends to kill the diversity of population. Due to this the simple GA converges early. To maintain the diversity in GA many schemes are used like crowding factor and fitness sharing. In crowding factor scheme an overlapping population is used where individuals replace existing strings according to their similarity. In fitness sharing scheme, a sharing function is defined to determine the neighborhood and degree of sharing for each string in population. Individuals who are close or similar to each other share their fitness and individuals who are dissimilar share less. Another issue with simple GA is that generally it converges in a single optimal result. Its inability to provide multiple or other sub optimal results, refrain from identifying the multiple or other weak motifs present in the sequence.

To maintain the diversity of population, in addition to genetic algorithm we used clustering scheme. Here we partition the population in multiple clusters and allow only intra-cluster selection and mating. Hence the selective pressure is confined within a cluster and multiple clusters maintain the diversity among population. This scheme help our algorithm to preserve the diversity of population over the generations against the selective pressure and the second advantage of this scheme is its ability to find multiple significant motifs from the given sequence data set, if any present. Our clustering scheme is based on the dissimilarity between the fittest and the rest of the population. To measure the dissimilarity between the fittest and an individual, first we sort the population in descending order to get the fittest member. Then we computed Hamming distance between the fittest candidate motif and other individuals to measure dissimilarity. We clustered the members on the basis of distance from the fittest. All individuals at distance *d* go in same cluster.

4.2 Representation

To represent an individual we used the position based representation approach as used in [5], [16], each individual is represented by a vector P = { p_1, p_2, . . ., p_N} storing the set of possible starting positions for the TFBS instances in each sequence. Here P represents a possible consensus solution set M = {m_1,m_2, . . ., m_N}, where each p_i is uniquely mapped to instance m_i with *w* known. This approach is explained in figure 1.

p_1	p_2	p_3	p_4	···	p_n
39	138	224	71	···	164
m_1	m_2	m_3	m_4	···	m_n

Figure 1. Representation of an individual: p_i is the position of a candidate motif m_i in i^{th} sequence

4.3 Fitness score function

The fitness function is to evaluate how good the individuals are. To compute the fitness of each individual in population we used the fitness score function that computes the similarity score of the consensus string produced by an individual. A pattern of nucleotides that is represented by maximum frequency at a position is called the consensus string. This similarity score is computed using the PWM (position weight matrix) of each individual. This is defined as:

$$Fit\_Score(M) = \sum_{i=1}^{w} f_{\max}(i) \qquad ...(2)$$

here M is a candidate consensus motif, w is the length of motif and $f_{\max}(i)$ is the maximum frequency value in column i. This is explained in the figure 2 given below:

4.4 Selection

Maintaining population diversity and selective pressure is the key issue while using a selection method. We performed the intra cluster selection. We used elitism to retain the best members of a cluster and remaining is selected using the stochastic tournament selection model. Every time, randomly two individuals are selected and the one with higher fitness score is used for mating.

4.5 Crossovers and Mutation

To generate new offspring from their parents we used one point crossover method. In this method a crossover point less than the length of individual, is randomly generated. Then after the crossover point, the substrings representing the parents are swapped.

There may be chances of being trapped in a local optima and getting the false motif. To avoid this we used mutation. Mutation also helps in maintaining population diversity and fast convergence of GA. To produce the mutation effect, first we selected a victim individual randomly and then changed its position value randomly.

4.6 Algorithm

```
//Initialization
n ← Number of individuals in population
import promoter sequences $S_1$ - $S_N$
for $k$ = 0 to $w$ do
    create Cluster( $k$ )
end for
//Fitness Evaluation
for $i$ = 1 to $n$ do
    randomly create candidate chromosomes of $N$ length: $P_1$ - $P_n$
    extract the consensus motifs from chromosomes : $M_1$ - $M_n$
    compute $Fit\_Score(M_i)$ for each candidate motif
end for
// Generation cycle
while stopping criteria is not satisfied
    sort population in descending order on $Fit\_Score(M_i)$
    // Make Clusters
    for $i$ = 1 to $n$ do
        $k$ == HammingDistance($M_1$, $M_i$ )
        put $P_i$ in Cluster( $k$ )
    end for
    //Selection : elitism
    for $k$ = 0 to $w$ do
        insert best individual of the Cluster( $k$ ) in mating pool
        for $j$ = 1 to Cluster( $k$ ).$size$ -1 do
            //Tournament Selection
            get two individual randomly $P_a$ and $P_b$
            if $Fit\_Score(M_a) > Fit\_Score(M_b)$ then
                select $P_a$
            else
                select $P_b$
            end if
        end for
        //One point crossover
        make random pairs of individuals
        perform one point crossover for each pair
        produce two offspring from each pair
        //Mutation
        randomly find the victim individual
        randomly modify the victim position value
    end for
    // Insertion & Evaluation
    for $j$ = 1 to $n$ do
        replace current individuals by newly produced offsprings
        extract candidate motifs from new chromosomes : $M_1$ - $M_n$
        compute $Fit\_Score(M_j)$ for each candidate motif
    end for
end while
```

m_1	**A**	**G**	**T**	**G**	**A**	**C**	**G**	**T**
m_2	**A**	**G**	**T**	**G**	**A**	**C**	**G**	**A**
m_3	**T**	**G**	**A**	**G**	**T**	**C**	**G**	**T**
m_4	**A**	**G**	**T**	**G**	**A**	**C**	**G**	**G**
m_5	**C**	**A**	**G**	**G**	**C**	**T**	**G**	**C**

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
A	0.6	0.2	0.2	0.0	0.6	0.0	0.0	0.2
T	0.2	0.0	0.6	0.0	0.2	0.2	0.0	0.4
C	0.2	0.0	0.0	0.0	0.2	0.8	0.0	0.2
G	0.0	0.8	0.2	1.0	0.0	0.0	1.0	0.2
Consensus String	**A**	**G**	**T**	**G**	**A**	**C**	**G**	**T**

Fitness Score = 5.8 0.6+0.8+0.6+1.0+0.6+0.8+1.0+0.4

Figure 2. A consensus string representation and it's fitness score computation

5. SIMULATION RESULTS

In order to evaluate the performance of our algorithm for motif identification, we used the synthetic datasets comprising various combinations. This includes the number of sequences 8 - 16, sequences of length 200 to 400 bp, size of motifs and conservation of motifs. We embedded each sequence with the instances of a known motif at random positions. We also modified the motif slightly in different sequences to produce the mutation effects.

For each simulated dataset, to evaluate the performance of our algorithm we used the standard information retrieval parameters, precision and recall. Precision *P* is number of predicted motif sites that are true sites divided by number of predicted motif sites and recall *R* is number of predicted motif sites that are true sites divided by number of true sites. These two parameters are combined to compute the standard parameter for comparison F-score, as follows:

$$F = 2 * precision * recall / (precision + recall) \qquad \ldots(3)$$

High values of F occur only when both precision and recall are high. The average of precision, recall and F score was calculated for the discovered motifs for each dataset. Results of various scenarios the number of sequences, length of sequences, average motif width, conservation of motifs, precision, recall and F-score for each simulation condition are shown below in table 1. The F-score for single motif identification is up to 0.90 for long motif lengths with good conservation, however for long motifs with poor conservation this is about 0.79. The results show that with poor conservation it is difficult to identify the correct motifs.

To evaluate the algorithm's ability for identification of multiple motifs we embedded some datasets with multiple known motifs of the same length and carried a fix number of runs. We compared the motifs retrieved by algorithm with original implanted motifs. The cases where we found motif instances of more than 70% similarity with original implanted motifs, we considered this as threshold for successful identification. The results of number of implanted motifs and successful identification of number of motifs are listed in table 2.

We also tested this algorithm with the real biological datasets. We used the promoter sequence data of Saccharomyces cerevisiae. We run this algorithm against ten target genes of transcription factor MIG1, seven target genes of PDR3 transcription factor and six genes of MCB transcription factor. We also executed the algorithm against the target genes of transcription factor SCB and UASCAR. The experimentally reported consensus motif and motif identified by algorithms are shown below in table 3.

Table 1: Results of various scenarios

S.No.	(N)	(L)	(W)	(C)	Precision	Recall	F- Score
1.	08	200	S	G	0.75	0.75	0.750
2.	08	200	M	G	0.75	0.88	0.810
3.	12	300	L	G	0.83	0.91	0.868
4.	12	300	S	G	0.75	0.83	0.788
5.	16	400	M	G	0.81	0.87	0.839
6.	16	400	L	G	0.87	0.94	0.904
7.	08	200	S	P	0.55	0.66	0.600
8.	08	200	M	P	0.66	0.77	0.711
9.	12	300	L	P	0.71	0.78	0.743
10.	12	300	S	P	0.66	0.73	0.693
11.	16	400	M	P	0.7	0.76	0.729
12.	16	400	L	P	0.77	0.83	0.799

N : *number of sequences* ***L:*** *length of sequence* ***W:*** *predicted motif width* ***C:*** *conservation*
S: *short* ***M:*** *medium* ***L:*** *long* ***G:*** *good* ***P:*** *poor*

Table 2: Results of various scenarios

S.No.	(N)	(L)	(w)	(N_M_E)	(N_M_I)
1.	08	200	S	04	03
2.	12	300	S	06	04
3.	16	400	M	04	03
4.	12	300	M	06	05
5.	16	400	L	06	05

N_M_E: *number of motifs embedded* ***N_M_I:*** *number of motifs identified*

Table 3: Results of biological promoter sequences

S.No.	TF Data set	Reported Consensus Motif	Discovered Motif
1.	MIG1	CCCCRNNWWWWW	CCCCACAGTTTT
2.	PDR3	*TCCGYGGA*	*TCCGCGGA*
3.	MCB	*WCGCGW*	*ACGCGT*
4.	SCB	CNCGAAA	CACGAAA
5.	UASCAR	*TTTCCATTAGG*	*TTTCCATTAGG*

6. CONCLUSION

Identification of transcription factor binding sites is an important and difficult problem. Most of the existing methods such as Gibbs sampling algorithm are local search methods, so they may suffer from the problem of local optima. Genetic algorithm provides a good approach to solve this problem. Genetic algorithm solves the optimal problem based on the biological characteristics. In this paper, we used the position based consensus representations for individuals of the population and clustering of population scheme.

Simulation results of the algorithm on synthetic data including various scenario shows that the algorithm is able to predict the motifs with average F-score in the range of 0.75 – 0.90 for good conservation, where as for poor conservation F-score drops to the range of 0.60 – 0.79. The algorithm is also able to detect multiple motifs of same length present in the sequences.

Our algorithm is able to find the correct motifs in the promoter data of Saccharomyces cerevisiae. The performance of this approach can probably be improved using more intelligent operators for selection, crossover and mutation. Currently the algorithm can find multiple motifs of only same length, with suitable encoding techniques this can be made to identify multiple motifs of variable lengths. On the other hand, the fitness evaluation can be improved if we are able to additionally incorporate terms that reflect the biological messages behind the similarities among motifs.

7. REFERENCES

[1] Lockhart D., Winzeler E., 2000. Genomics, Gene Expression and DNA Arrays. Nature, 405, 827-836.

[2] Stormo G.D., 2000. DNA binding sites: representation and discovery. Bioinformatics, vol 16, 16-23.

[3] V. Matys, E. Fricke, R. Geffers, E. Gssling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A.E. Kel, O.V. Kel-Margoulis, D. U. Kloos, S. Land, B. Lewicki-Potapov, H. Michael, R. Munch, I. Reuter, S. Robert, H. Saxel, M. Scheer, S. Thiele and E. Wingender, 2003. TRANSFAC: Transcriptional Regulation, from Patterns to Profiles. Nucleic Acids Research, vol. 31, no. 1, pp. 374-378.

[4] A. Sandelin, W. Alkema, P. Engstrom, W.W. Wasserman, and B. Lenhard, 2004. JASPAR: An Open-Access Database for Eukaryotic Transcription Factor Binding Profiles. Nucleic Acids Research, vol. 32, pp. D91-D94.

[5] Tak Ming Chan, Kwong Sak Leung and Kin Hong Lee, 2008. TFBS identification based on genetic algorithm with combined representations and adaptive post-processing. Bioinformatics, Vol. 24 no. 3, pages 341–349.

[6] Stormo G.D., 1988. Computer methods for analyzing sequence recognition of nucleic acids. Annual Review BioChem, vol 17, 241–263.

[7] Modan K Das and Ho-Kwok Dai, 2007. A survey of DNA motif finding algorithms. BMC Bioinformatics, (Suppl 7), S21.

[8] Bailey T.L. and Elkan C., 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, AAAI Press, Menlo Park, California, pp. 28-36.

[9] Thompson W., Rouchka E.C. and Lawrence C.E., 2003. Gibbs Recursive Sampler: Finding transcription factor binding sites. Nucleic Acids Research, Vol.31, pp. 3580-3585.

[10] Hertz G.Z., Hartzell G.W. and Stormo G.D. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. Bioinformatics, Vol.6, pp. 81-92.

[11] Liu F.F.M et al. 2004. FMGA: Finding Motifs by Genetic Algorithm. Proceedings of the Fourth IEEE Symposium on Bioinformatics and Bioengineering, pp.459-466.

[12] Stine M., Dasgupta D. and Mukatira S., 2003. Motif Discovery in Upstream Sequences of Coordinately Expressed Genes. The 2003 Congress on Evolutionary Computation, pp.1596-1603.

[13] Vijayvargiya S., Shukla P., 2011. A Structured Evolutionary Algorithm for Identification of Transcription Factor Binding Sites in Unaligned DNA Sequences. International Journal of Advancements in Technology, Vol 2: No 1, page no. 100 – 107.

[14] Wang T, Stormo GD. 2003. Combining phylogenetic data with coregulated genes to identify regulatory motifs. Bioinformatics, vol 19, pp. 2369-2380.

[15] Sinha S, Blanchette M, Tompa M., 2004. PhyME: A probabilistic algorithm for finding motifs in sets of orthologous sequences. BMC Bioinformatics, 5:170.

[16] Wei Z. and Jensen S.T., 2006. GAME: detecting cis-regulatory elements using a genetic algorithm. Bioinformatics, vol 22, pp. 1577–1584.