Motif finding using multi-objective genetic algorithm

Anushree Jagrawal [1], Neha Mathur [2]
[1] Computer Science Engineering, Birla Institute of Technology, Jaipur, Rajasthan, India
[2] Computer Science Engineering, Birla Institute of Technology, Jaipur, Rajasthan, India
anshi31jagrawal@gmail.com[1],nhmathur9@gmail.com[2]

***Abstract*— With the development of gene sequencing technology, almost all the genes in the living creatures of the world have been sequenced and enormous amount of data has been generated. But the identification of Transcription Factor Binding Sites (TFBS) also called as motifs, which help in understanding the function of the proteins coded by a specific genome remains a major challenge. The DNA motif discovery problem abstracts the task of discovering these short, conserved sites in the wealthy data of genomic DNA sequences and forms a crucial topic in computational bio-informatics. Since the size of data search space is very large, genetic algorithm (GA) proved to be an efficient tool. In this paper we present a GA based multi-objective optimization method to find solutions for the above mentioned problem.**

Keywords: motifs, GA, multi-objective optimization
Subject Classification: Genetic Algorithm

I. INTRODUCTION

The complete information of characteristics of a living organism is stored in a simple molecule called as Deoxyribonucleic Acid, in short DNA. The building blocks of DNA are the four nucleotides, namely Adenine, Thymine, Guanine and cytosine, denoted as A,T,G and C. As such, DNA can be seen as a sequence of these four nucleotides. Some portions of the DNA sequences undergo the process of Transcription, forming RNA and consequently Translation, forming proteins which dictate the function of a particular cell. If a protein is being synthesized at a certain state, its coding DNA (called a gene) is termed as "active" or "expressed."

The DNA sequences also consist of promoter regions that activate and deactivate the genes. Transcription factor proteins (TFs) initiate the process of transcription by binding themselves to DNA promoter regions at Transcription Factor Binding sites (TFBSs), also known as motifs. Motif can be identified as an evolutionarily conserved pattern recurring in nature that is common in two or more DNA promoter sequences. The motifs are presumed to have important biological meaning and functions. As such, locating and characterizing motifs forms a crucial task in understanding cell functionalities and finding answers of many unanswered questions in genetics.

This simple to solve looking problem becomes complicated due to the fact that it is quite difficult for all sequences to have a completely matched motif pattern because of the poor conservation and short length of the transcription factor binding sites or motifs in comparison with the length of promoter sequences. Moreover, these motifs are present at different positions within the sequences. This results in mutations which affect gene regulation and contribute to evolution.

Although there are many methods proposed to predict motifs in the past, the motif finding problem is still a major challenge in the field of bio-informatics. The earliest methods include brute force method, greedy approach and branch and bound technique. However the efficiency, run times and the inability to work on large data sets made the researchers find improved methods. The most popular methods include Multiple Em for Motif Elicitation (MEME)[1], Gibbs sampler[2], FMGA[3], GA with clustering[4] , NSGA[5], MOGA[6]. Even with weak signals, MEME effectively find motifs of variable width and occurrences in DNA with high prediction accuracy. Gibbs Sampler is better than other techniques in terms of computation time. NSGA is a non-dominated sorting based multi-objective evolutionary algorithm.

In this paper we compare FMGA[3] and GA with clustering[4] with the results from MEME[1] as ways of solving the motif finding problem. Since both the methodologies involve the concepts of Genetic Algorithm, in Section {III}, we describe GA as an efficient tool for solving the motif finding problem. In sections {IV and V} we discuss two algorithms used to solve the motif-finding problem, FMGA and GA with clustering, in detail. In section {VI}, we present the conclusion obtained by comparing the two algorithms and analyzing the appropriate problem domains for both algorithms.

II. PROBLEM STATEMENT

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 instance M={m_1, m_2,..., m_N} where each m_i is a subsequence with length *w* from sequence Si, such that all m_i's are nearly matched and fulfill the fitness criteria of being an optimum solution.

III. GENETIC ALGORITHM

There are various impressive phenomena going around in the nature which have always attracted the interest of researchers due to their admirable perfection. One such phenomenon, that our world has experienced since ages, is the evolution of human beings which is based on the famous Darwin's theory of the "Survival of the fittest!!" This idea has been modeled into an algorithm known as the Genetic Algorithm (GA).

GA is a popular meta-heuristic algorithm efficient in finding the globally optimal solutions among the potentially huge data search spaces and provides solutions to many Multi-Objective Optimization problems. It operates on a population of candidate solutions to a specific problem domain. Specifically, the structure in the current population is evaluated for its effectiveness as a solution during each generation. Based on this evaluation, a new population of candidate structures is formed using operators like crossover and mutation. This process is iterated until an optimal solution is found or no improvement is achieved after a significant amount of evaluations. The adjacent flowchart describes the algorithm:

As can be seen from the above algorithm, the transition from one generation to the next consists of four basic operators:

1. Selection

Mechanism for selecting individuals (strings) for reproduction according to their fitness (objective function value). It is the component which guides the algorithm to the solution by preferring individuals with high fitness over low-fitted ones. It can be a deterministic operation, but in most implementations it has random components.

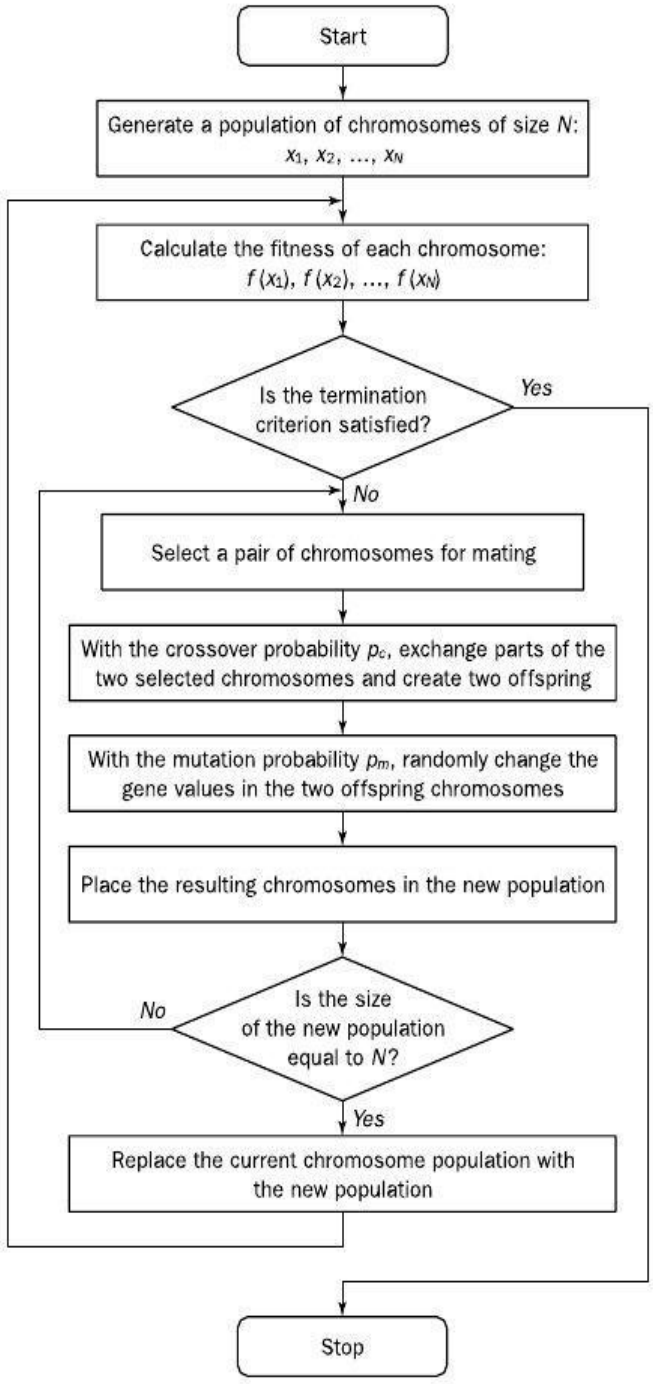

Fig.1. Flowchart of genetic algorithm

2. Crossover

In reproduction, as it appears in the real world, the genetic material of the two parents is mixed to produce offspring gene sets. Usually, chromosomes are randomly split and merged, with the consequence that some genes of a child come from one parent while others come from the other parents. This mechanism is called crossover. It is a very powerful tool not only for introducing new genetic material and maintaining genetic diversity, but with the outstanding property that good parents also produce well-performing children or even better ones.

3. Mutation

In real evolution, the genetic material can by changed randomly by erroneous reproduction or other deformations of genes, e.g. by gamma radiation. In genetic algorithms, mutation can be realized as a random deformation of the strings with a certain probability. In real reproduction, the probability that a certain gene is mutated is almost equal for all genes. The positive effect is preservation of genetic diversity and, as an effect, that local maxima can be avoided.

4. Sampling

Procedure which computes a new generation from the previous one and its offsprings based on their fitness values. On this sampled population next iteration is employed.

Generally speaking, genetic algorithms are simulations of evolution which proves to be an efficient tool for finding solutions from among a very large potential solution space. This fact makes it apt for being used to solve the motif finding problem.

IV. FMGA

FMGA (Finding motif using Genetic Algorithm) is a heuristic approach based algorithm to predict motifs using a total fitness score function and to find the optimal motif using genetic algorithm. It uses the general genetic algorithm framework and operators to serve as its basic architecture. The data is selected in the regions located from the -2000 bp upstream to +1000 bp downstream of transcription start site (TSS).].In case of ambiguity, code penalties are applied to obtain the predicted motif more efficiently.

1. Fitness Function:

Given a motif pattern, there may have several regions in the sequence that match the motif pattern and each has a fitness score according to the fitness score function defined as follows:

$FS(S_M, P_N) = max_j (\sum_{i=1}^{k} match (S_{mji}, P_{ni})/k)$

where,

$match (S_{mji}, P_{ni}) = \{ 1 \text{ if } S_{mji}=P_{ni} , 0 \text{ if } S_{mji} !=P_{ni} \}$,

and *m* is the index of sequences, *i* is the position within the motif , *n* is the index of motif patterns, *k* is the length of motif pattern, *j* is number of matched regions in the sequence.

Algorithm:

Initialization

Setting total number of iterations: M

Creating candidate motifs randomly: $P_1 \sim P_n$

Import promoter sequences $S_1 \sim S_L$

While (iteration number <= M)

{

While (predicted motif is unchanged for more than K generations)

{

Evaluating TFS for each candidate motif

Keeping the candidate motifs with the highest TFS as the new generations, the remaining candidate motifs are created by weighted wheel selection

Mutation using weight matrix to generate two parent patterns

Crossover with ambiguity codes penalties to select the best child pattern for next generation

}

Rearrangement of candidate motifs

Increasing iteration number by 1

}

Output predicted motifs and corresponding TFS

Mutation is performed by first creating a weight matrix from the matched motif patterns in every sequence. The score in weight matrix is calculated as the ratio of occurrences of corresponding base and the numbers of matched motif patterns. FMGA predicts better motif patterns than Gibbs sampler and spends less computation time than MEME. Moreover, FMGA can predict more potential motifs than the other algorithms because the patterns are generated randomly during the operation processes of GA.

V. GA WITH CLUSTERING

Clustering methods divide the dataset into groups called clusters such that the objects in the same cluster are more similar and objects in the different clusters are dissimilar. Candidate motifs are differentiated on basis of some similarity factor and thus formation of clusters takes place. Implementing the simple Genetic Algorithm, we can observe that high degree of elitism arises, that leads to occurrence of a single fittest motif. After few generations this selective pressure tends to kill the diversity of population. Due to this the simple GA converges early. On the other hand, clustering provides alternative solutions and maintains diversity.

Some clustering techniques that are available in the literature are creating clusters on basis of Hamming distance evaluation, *K*-means algorithm, branch and bound procedure, maximum likelihood estimate technique and graph theoretic approaches. To maintain the diversity in GA many schemes are also 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.

In this algorithm, an initial set of chromosomes is initiated as population, and the fitness is evaluated. The population is arranged in descending order to get the fittest member, and then clustering criteria is applied i.e. hamming distance for evaluation of dissimilarity between the fittest member and other individuals.

Here we partition the population in multiple clusters and allow only inter-cluster selection and mating. 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.

Algorithm:

```
//Initialization
n ← Number of individuals in population
import promoter sequences S1 - SN
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: P1 - Pn
        extract the consensus motifs from
        chromosomes : M1 - Mn
        compute Fit_Score ( Mi ) for each candidate
        motif
end for
// Generation cycle
while stopping criteria is not satisfied
sort population in descending order on Fit_Score( Mi
)
// Make Clusters
for i = 1 to n do
        k == HammingDistance(M1, Mi )
        put Pi 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 Pa and
                Pb
                if Fit_Score( Ma ) > Fit_Score( Mb )
                then
                select Pa
                else
                select Pb
                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 : M1 - Mn
        compute Fit_Score( Mj ) for each candidate
        motif
end for
end while
```

VI. CONCLUSIONS

Motif identification is an important problem in bio-informatics that involves the search for approximate matches. Various algorithms have been proposed, including exhaustive searches as well as heuristic searches that involve searching only a subset of all the possible solutions .The search space in the motif discovery problem is extremely large and a good non-exhaustive method must intelligently choose which candidates to examine. The selection of candidate solutions, is thus the key to the solution of the motif discovery problem .In this paper, we have analyzed two algorithms i.e. FMGA and Clustering in terms of various problem domains where they can be implemented to give appropriate results.

FMGA uses a Top down approach. While identifying motifs, it takes motif length and promoter sequence as an input and tries to search for the same sequence or a mutated sequence in the search space. Since it forms a weight matrix for concluding a consensus string in each iteration (which is a kind of statistical methodology), the results found are more accurate .On the other hand, the mechanism of clustering technique is fully random and heuristic in nature. It only takes length of the motif as input, and provides multiple solutions with varying levels of accuracy. Although clustering technique is less accurate than FMGA ,but its advantage over FMGA is the inevitable .The availability of multiple motifs and maintenance of diversity widens the scope of clustering .The performance of this approach can probably be improved using more intelligent operators for selection, crossover and mutation. The problem domain of clustering is very wide. Firstly, it can be used for routing problems where multiple routes are to be detected for same source and destination. The multiple routes can be categorized on the basis of the priority of the factors forming fitness function, for instance cost, time, traffic density. Secondly, it can be used by MNC for forming risk management strategies, which require successful alternative solutions .Depending on which factor has led to the unforeseen risk, the major factor can be prioritized more in the fitness function to obtain required solution.

However, FMGA specifies domain where accuracy is preferred more than alternatives. It can be used in medical sciences for detection of probable diseases that can develop in species or more specifically humans. For example, we know the motif sequence that leads to disease like HIV, diabetes, etc. These sequences can be given as promoter sequence, and its presence in the species can be estimated .Also prediction of genetically inherited diseases that can develop in future can be detected in a new-born, and preventive measures can be taken. Hence, both the algorithms perform best in some specific domains, still they can be interchangeably on the preference of programmer.

VII. REFERENCES

[1] 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.

[2] 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.

[3] Falcon F.M Liu1, Jeffrey J.P. Tsai1, R.M Chen, S.N. Chen and S.H. Shih," *FMGA: Finding Motifs by Genetic Algorithm*"

[4] Shripal Vijayvargiya and Pratyoosh Shukla," *A Genetic Algorithm with Clustering for Finding Regulatory Motifs in DNA Sequences*"

[5] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T Meyarivan," *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II"*

[6] Abdullah Konak, David W. Coit, Alice E. Smith," *Multi-objective optimization using genetic algorithms: A tutorial"*