A Genetic Optimization approach for finding common Motif in Biological Sequences

Vishal
Department of Computer Science
PEC University of Technology
Chandigarh, India
Email: vishal_sabi@yahoo.co.in

Shailendra Singh
Department of Computer Science
PEC University of Technology
Chandigarh, India
Email: sscse@in.com

Abstract

The field of Bioinformatics is gaining much attention these days due to advancement of computer programs and molecular biology, but there are many human activities which are lying unknown yet. Finding common motif is also a major topic in the field. As with the similarity in the sequences, their families and activities can be identified easily when one of the sequences is tested. The common motif finding can help finding unknown members of a family and can help in applications like drug design. In this paper, it is proposed to find common motif in biological sequences. Here RNA sequences are used. The paper is divided mainly in to three sections. Firstly some basics of bioinformatics are discussed followed by the proposed approach and finally the result and conclusions are presented.

1. Introduction

In recent days, with the advances in the computer technology and molecular biology, the field of Bioinformatics flourishes so much. The bioinformatics is fast growing field that uses computer algorithms to solve the molecular biology problems. As the enormous amount of data in the field of molecular biology has increased much, there is need of search and processing tools for the information extraction. The technology has also in much boom these days. So there is a need of efficient tools to extract and manipulate the information to make important discoveries and discovery makings. The common motif identification problem is gaining many attentions these days, which gives the secondary structure similarity with other sequences and classify the unknown members of family. Knowing the structural motifs can help us to gain a deeper insight of the regulation activities [1]. The motif finding approaches are divided in to mainly two classes. The first class requires motif specification or motif descriptor and the second class of technique uses sequence alignment and evolutionary techniques to find the motifs. The descriptor based approaches require descriptor as guide to search the motifs. The descriptor based methods require an expert user prediction about the descriptor which can give sufficient results. The commonly descriptor based approaches in literature are RNAMOT [2, 3], PatScan [4], RNAMotif [5]. The quality of the results mainly upon on the quality of the descriptor and an adequate scoring function. The second class of approaches use sequence alignment approaches and evolutionary techniques for finding common motif. These commonly used approaches are ERPIN [6], Infernal [7], RAGA [7]. There are some other approaches which does not use sequence alignment and uses only evolutionary computing. The commonly used techniques in literature are GeRNAMo [9], GPRM [1].

In this paper, it is purposed to use genetic algorithm [10-13] approach to find common motifs in biological sequences.

2. Motif basics and materials used

The section discusses about the motif basics such as its representation, structure, and data set used to find the common motif in the data set.

2.1 Motif representation

The motif is a sequence pattern that has some biological significance. The types of motifs are hairpin loop, stem, internal loop, multi-branch loop, bulge loop. The motifs are as shown in the figure 1. The 5’ and 3’ are start and end points of sequence structure. The curved region is unpaired region and straight lines represent paired one.
Figure 1: The types of motifs

2.2 Motif notations

There are two representations to describe the motif. One is complementary representation which is watson crick pairing that is between AU, CG and GU as wobble pairing. These are represented by h5 and h3. Second is single strand represented by ss. The dot bracket notation is also another notation, where a dot represent unpaired base and brackets are paired base. The compressed form representation is as ss (x:y) h5 (x:y) ss (x:y) h3 (x:y) ss (x:y), where x and y are minimum and maximum lengths of paired or unpaired patterns.

2.3 RNAFOLD

The secondary structure prediction is done using RNAFOLD which gives suboptimal secondary structure of RNA sequence. RNAFOLD is given as a built in function in MATLAB. The output of secondary structure prediction is dot bracket form and minimum energy for that sequence. For example, for the sequence of ‘AUUCGGUUAUAGCCGAAU’ the folding result is ‘((((((...))))))’ and energy value of -7.9. The compressed form representation can be written as ‘h5 (8) ss (3) h3 (8)’.

2.4 Common motif

The common motif can be found in multiple sequences after motifs are identified in multiple sequences. For example, take three sequences as AAGGACUCCUAGUCCCA, GGAGGACCUC-GUCCA and CCGAGACCUCUGUCUCCAAA. Now fold these sequences. The common motif present in the sequences AAGGACUCCU-CAGUCCCA, GGAGGACC-CUCGUCCA and CCGAGACCUCUCUGUCUCCAAA is hairpin motif described as: h5(4:6) ss(4:5) h3(4:6) as shown in figure 2. The number in the brackets is length ranges. The number on left side is minimum length and on right side is maximum length.
The input sequence is folded with the help of RNAFOLD which gives dot bracket notation. For example for the input sequence of ‘AGUUACACUGCCUCUGAGCUG’, the dot bracket notation can be written as ‘……..((……))..’ where a dot denotes unpaired segment and a bracket denotes paired segment. The dot bracket form can also be written in compressed form representation. As in the example the compressed form is ‘ss (9) h5 (2) ss (6) h3 (2) ss (2)’. This is same as motif representation. The number in the bracket represents the number of occurrences of the paired or unpaired segments.

3.2 Encoding the input sequence

The third step is the encoding of sequence. As with the dot bracket notation and compressed notation, these are difficult to make further process, this is encoded in to some suitable form which can be easily compared to find a match.

3.3 Encode the descriptor

The fourth step is the encoding of descriptor. Similarly it is also converted to same encoded form as of input sequence so that both can be compared for match.

3.4 Matching patterns for Pool creation

The fifth, sixth and seventh steps are matching the patterns and making the pool. As the both sequence and descriptor are in same notations, they can be easily matched for finding occurrence of matched patterns in the step 5. The step 6 check for match, when a match is found, the matched substring is copied to pool that serves as initial population in the step 7. The whole sequence is searched for match with the descriptor. Only the pattern is compared and not the length part. The occurrences matched are recorded and stored for further processing. Put the matched sub-sequences in to a pool, and name it as motif pool as step 7. This motif pool will contain all the occurrences of motifs in the whole sequence. The pool in the step 8 serves as initial population for genetic algorithm for further processing.

3.5 Fitness evaluation

The step 9 is to evaluate fitness values for the entire individual in the population. Now Genetic algorithm is applied to find the best fit common motif among the all motifs in the motif pool and also by making some changes to the present motif pool. GAs are adaptive and robust computational research
procedures, modeled on the mechanisms of natural genetic system [14]. The genetic algorithm begins with the population initialization as in step 8 and then finding their fitness for that particular application. Evaluate each individual in population against objective function. Each putative motif in the pool represents an individual in the population. Now each individual is tested and against the fitness function. The fitness function has been taken from GeneXproTools 4.0 a framework [14]. The fitness \( f(ij) \) of an individual program \( i \) for fitness case \( j \) is evaluated by the formula:

\[
\text{If } E_{(ij)} \leq p \text{ then } F_{(ij)} = 1 \text{ else } F_{(ij)} = 0 \quad (1)
\]

In eq (1) Where \( p \) is precision and \( E_{(ij)} \) is relative error of an individual \( I \) for the fitness case \( j \) which is evaluated by:

\[
E_{(ij)} = \left| \left( \frac{P_{(ij)} - T_j}{T_j} \right) \right| \cdot 100 \quad (2)
\]

In eq (2) \( P_{(ij)} \) is value predicted by individual program \( i \) for the text case \( j \) and \( T_j \) is target value for the fitness case \( j \).

The overall fitness \( F_i \) of an individual program \( i \) is expressed as

\[
F_i = \sum_{j=1}^{N} F_{(ij)} \quad (3)
\]

Where \( n \) is the total number of fitness cases as shown in eq (3). Each individual motif in the population of motif pool is tested against the predictor to find the best one.

### 3.6 Selection of individuals

Each individual in the population is given a fitness value which serves as basis for its selection in operator’s decision makings. The best fit candidates are selected using the selection strategies based on their fitness values. The step is as shown in the flow chart as step10
3.7 Generating new offsprings

The selected candidates are termed as parents and these are used to make new offsprings that can solve the problem more efficiently. The steps 11 crossover and step 12 mutations are applied to parents to make new individuals for new population. The crossover is done on two or more individuals, whereas the mutation is done on one individual only. Then add the generated offsprings to new population in the step 13.

3.8 Stopping criterion

The steps 14 and 15 in the flowchart help to make a decision regarding number of generations. The process ends with maximum number of generations are met or the objective value is produced.

3.9 Display best fit motif

In the last step the best fit individual is displayed i.e. common motif with maximum fitness value is displayed along with its fitness value.

4. Result

The motif discovery approach was implemented by using the MATLAB 7.11.0.(R2010b) on windows 7 system with 3 GB of RAM. We experiment by varying several parameters and finally settle down to parameters shown in TABLE-1 shown. Out of 25 sequences, the motifs appear in 20 sequences. The genetic parameters to which we finally settle down are as shown below in Table 1.

<table>
<thead>
<tr>
<th>Number of Generations</th>
<th>51</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Selection Function</td>
<td>Stochastic uniform</td>
</tr>
<tr>
<td>Mutation function</td>
<td>Gaussian</td>
</tr>
<tr>
<td>Mutation scale</td>
<td>1.0</td>
</tr>
<tr>
<td>Crossover function</td>
<td>Scattered</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 1: Genetic Parameters

Figure 3: Proposed Methodology
The common motif predicted by the proposed method is as shown below:-

\[
\begin{align*}
&\text{ss (1:8)} \ h5 (1:16) \ ss (1:6) \ h5 (1:20) \ ss (1:5) \ h5 (1:21) \\
&\text{ss (1:5) \ h5 (1:12) \ ss (3:13) \ h3 (1:12) \\
&\text{ss (1:6) \ h3 (1:22) \ ss (1:3) \ h3 (1:19) \ ss (1:6) \ h3 (1:13) \\
&\text{ss (1:15)}
\end{align*}
\]

5. Conclusions

The proposed methodology was implemented on Windows 7 operating system and microRNA data set was tested for finding best fit motif. The proposed approach is a supervised learning approach. The descriptor is termed as supervised output which gives idea about the number of complementary pairs and non-pairing segments. Then a motif pool was created by comparing input sequences with descriptor. Now genetic programming is used to find the best fit in accordance with fitness function. The new offsprings are made to improve the current population. The system halts when maximum number of generations are exceeded or fitness limit reaches. The best fit motif is very similar to the descriptor. The genetic programming uses idea of natural selection and global optimization which gives good results. Genetic algorithms are very fast and robust for search optimization problems. As number of generations increases, the best fit solution more and more approaches to descriptor. The proposed method is able to discover the common motifs in limited time. As the sequence length increases, the time required to find the common motifs also increase. This is one future direction of the proposed approach.

References


