21BIO112 Intelligence of Biological Systems-2

S2 CSE AIE B BATCH

Project Report

MOTIF DISCOVERY: GENETIC ALGORITHM

GROUP MEMBERS

Aiswarya Shajil Kumar	**AM.EN.U4AIE21107**
Anushka Kaimal	**AM.EN.U4AIE21116**
Navami S K	**AM.EN.U4AIE21146**
Parvathy G Pillai	**AM.EN.U4AIE21150**
Megha Mohan	**AM.EN.U4AIE21176**

Contents

ABSTRACT

During the time period from after the completion of the Human Genome Project to the present days, almost all the genes have been sequenced and enormous amounts of data have been generated. Hence, to separate useful information from these data is a very important topic. This new approach is developed based on the genetic algorithm (GA). The mutation in the GA is performed by using position weight matrices to reserve the completely conserved positions. The crossover is implemented with special-designed gap penalties to produce the optimal child pattern. We also present a rearrangement method based on position weight matrices to avoid the presence of a very stable local minimum, which may make it quite difficult for the other operators to generate the optimal pattern. Motif discovery is a procedure of finding enriched sets of similar short sequences in a large sequence dataset. In our case the large sequence dataset are sequences around ChIP peaks, while the short sequence sets are the transcription factor binding sites. There are two types of motif discovery tools: supervised and unsupervised. Supervised tools require explicit positive and negative sequence sets, and then search for relative enrichment of short motifs in the foreground versus the background. Unsupervised models, on the other hand, require only a set of positive sequences, and then compare motif abundance to a statistically constructed background set. Due to the combinatorial nature of the procedure, motif discovery is computationally expensive. Motif discovery in unaligned DNA sequences is a challenging problem in computer science and molecular biology. Motifs can be used to determine evolutionary and functional relationships. Over the past few years, many motif discovery tools have been designed and make available to public. So in this project we are trying to discover the motifs using Genetic Algorithm.

INTRODUCTION

Motif discovery in unaligned DNA sequences is a challenging problem in computer science and molecular biology. Motifs can be used to determine evolutionary and functional relationships. Over the past few years, many motif discovery tools have been designed and make available to public. They differ each other mostly in their definition of what constitutes a motif, what constitutes statistical overrepresentation of a motif and what method has been used to find statistically overrepresented motifs. All natural, prokaryotic, and eukaryotic creatures must deliver their chemical composition to survive, and gene expression is a crucial step. Gene expression is the process by which the instructions in our DNA are converted into a functional product, such as a protein. As seen in fig 1, this process involves two key steps: transcription and translation. During the meantime, the output and yield of genes to be expressed should be regulated, which is called the regulation of gene expression. Transcription regulation is a crucial stage in the expression of genes. Transcription is when the DNA in a gene is copied to produce an RNA transcript called messenger RNA (mRNA). Translation occurs after the messenger RNA has carried the transcribed 'message' from the DNA to protein-making factories in the cell, called ribosomes. Under certain rules, transcription factors are combined at the target gene promoter sequence designated sites, promoting gene transcription and gene transcription efficiency. Such specific sites are the sites binding transcription factor of varying lengths, ranging from a few to several tens of base pairs. Transcription factor binding sites generally have a fixed pattern — motif, which is the key content of gene transcription regulation with motif identification. Genetic algorithms are stochastic search algorithms, which are used to search large, non-linear spaces where expert knowledge is lacking or difficult to encode and where traditional optimization techniques fall short. The standard genetic algorithm usually starts from an initial population generated at random and attempts to improve on it using genetic operators in the guidance of a fitness function. In this paper, we present a new algorithm **GARPS** that optimizes **Genetic Algorithm via**

Random Projection Strategy to identify (l,d)-motifs.

Fig 1: Gene Expression

LITERATURE REVIEW

Transcription factor (TF) (or sequence-specific DNA-binding factor) is a protein that controls the rate of transcription of genetic information from DNA to messenger RNA, by binding to a specific DNA sequence. The function of TF is to regulate genes to make sure that they are best expressed in the right cell at the right time and in the right amount throughout the life of the cell and the organism. The amount of TF present in a particular cell depends upon the genomic size. A regulatory motif, or transcription factor binding site, is the specific short DNA interval to which a transcription factor binds to regulate a gene. Motif identification in a gene is a difficult task. There are two principal types of motif discovery algorithms; *i.e.* enumeration approach and probabilistic technique. Enumeration approach searches for consensus sequences; motifs are predicted based on the enumeration of words and computing word similarities so this approach is sometimes called the word enumeration approach to solve Panted (l, d) Motif Problem (PMP) with motif length (l) and a maximum number of mismatches (d). The algorithms based on the word enumeration approach exhaustively search the whole search space to determine which ones appear with possible substitutions and therefore it typically locates the global optimum. However, this also means that they are exponential-time algorithms that require a long time to detect the larger l and inefficient for handling dozens of sequences, so they are only suitable for short motifs. Moreover, these algorithms require many parameters determined by the users such as motif length, the number of mismatches allowed etc. The word enumeration approach can be accelerated by using specialized data structures such as suffix trees or parallel processing. A second group is a probabilistic approach. It constructs a probabilistic model called position-Specific Weight Matrix (PSWM) or motif matrix that specifies a distribution of

bases for each position in TFBS to distinguish motifs vs. non-motifs and it requires few search parameters. Recently, new algorithms inspired from nature are presented that solve complex and dynamic problems with appropriate time and optimal cost. These algorithms simulate the behaviour of insects or other animals for problem-solving. Evolutionary algorithms can over-come the disadvantages of local search and synthesize local search and global search. The beauty of nature-inspired algorithms is that they provide flexibility in evaluating the solutions by using fitness functions that score the solutions. Finally, the last category is a combinatorial algorithm that mixes multiple algorithms. The classification of motif discovery algorithms is shown in figure 2. These functions vary from problem to another and evaluate using different information types as biological information, functional information, etc. Moreover, these algorithms provide flexibility in motif representation. Motif identification methods can be classified into two major categories: molecular biology techniques or bioinformatics techniques. Popular methods in molecular biology are DNase foot printing, gel mobility shift method, SELEX, CHIP, etc., which consume a long period and high cost to successfully identify the form of a motif. Gene chip is designed to meet large-scale, high-throughput demand, thus playing an important role in biological technology. The use of particular oligonucleotide fragments or gene fragments has attained gene hybridization using gene chip technology with a labelled test sample operating on the theory of base pairing. Currently, gene chip can be used to express a large number of genes. Chromatin immunoprecipitation (ChIP) is another technique that has been applied to attain a large number of DNA fragments linked to the designated transcription factors. However, the length of the motif is about several bases to several tens of bases, whereas the resolution of the ChIP-chip and ChIP-seq technology is far greater than the mold length. So, these two methods cannot reliably identify the binding sites of motifs, or make the right predictions for repeated genomes.

Traditional bioinformatics studies typically characterize the motif through sequence analysis, using conventional mathematical

methods and high-performance computing technology. Based on various living organisms and motif types, several scholars made a lot of algorithms for motif recognition and depending on this search strategy, motif identification algorithms can be broadly classified into two categories. The first category is the exhaustive search algorithm. This algorithm appears to be simple by attaining the optimal solution theoretically, but because of the large amount of data available, the time complexity of this algorithm is very high. Thus, this method is preferred only for short motifs. The second category is a heuristic, stochastic approximate search algorithm. This method utilizes the location frequency matrix for an estimated description of motif information and adaptive ideas for updating motif forms, thus meeting the end condition of the algorithm or the number of iterations.

At the present day, the Genetic Algorithm is a topic of great importance. This algorithm was proposed by Stine. This algorithm is employed to identify the motifs of different lengths. Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning. An algorithm named Finding Motifs by Genetic Algorithm (FMGA algorithm) was proposed which accelerated the convergence speed and accuracy by initiating genetic operators to conventional genetic algorithms. A scientist named Che suggested Motif Discovery using Genetic Algorithm (MDGA algorithm). He was successful in executing the prediction of the homologous gene motif. Another algorithm, Genetic algorithm approach to Motif Inference (GAMI algorithm), which was successful in recognizing the long sequence motif, was preferred for distinct and large populations. In all the above algorithms, random initialization of the population was performed. The search scope of these algorithms was within the input population. However, the number of motifs in the population was unknown.

This project focuses on the implementation of motif discovery based on Genetic Algorithm.

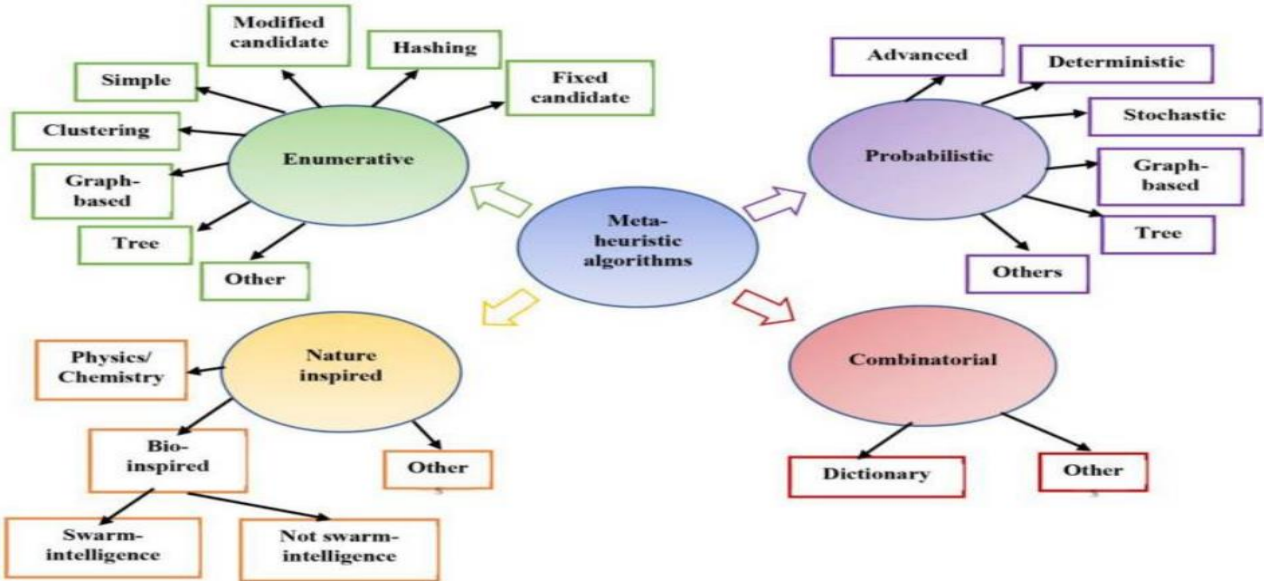


Fig 2: Classification of motif discovery algorithms as enumerative, probabilistic, nature inspired and combinatorial types.

PROBLEM DESCRIPTION

DNA, or deoxyribonucleic acid, is the hereditary material in humans and almost all other organisms. Most DNA is located within the nucleus of the cell (nuclear DNA), and a small amount of DNA is also present within the mitochondria (mitochondrial DNA). The information in DNA is stored by the pairing of chemical bases, adenosine (A) with thymine (T) and guanine (G) with cytosine (C). Each base is connected to a deoxyribose sugar and a phosphate molecule, thus building up the complete structure of DNA. DNA is the carrier of genetic information; thus, it assists the operations of vital functions and biological development within an organism. DNA sequence is present in both prokaryotes and eukaryotes. There are two types of encoded information in the DNA of eukaryotic organisms. The first is the coding region of the DNA/ gene region. It is the part of DNA that codes for a protein. The structure and function of a gene and its products is determined by this region. The second type is the non-coding region/regulation area. These are sequences of DNA that don't code for a protein. Functions of this region include transcriptional and translational regulation of protein-coding sequences i.e. it can recognize the

starting position of gene regulation.

Some of the problems we face in discovering motifs are: with the advances in high-throughput sequencing, motif discovery problems are challenged by both the sequence pattern degeneracy issues and the data-intensive computational scalability issues. Applying motif discovery algorithms to a group of related DNA sequences leads to the identification of putative transcription factor DNA binding sites. These algorithms output a set of DNA motifs, which are frequently redundant. To infer the correct transcription regulation map from the discovered motif set, it is crucial to reduce this redundancy and to relate the discovered motifs to known ones.

Gene expression is defined as the process by which information from a gene is used in the synthesis of a functional gene product. These products include mostly protein but at times, the product includes functional RNA. Transcriptional regulation, a key step in gene regulation, is the process by which the cell controls the transformation of DNA to RNA, thus regulating gene activity. Transcriptional factors (TF) is a protein that controls the rate of transcriptional regulation. Thus, in this process, TF binds to a specific short DNA interval in the target promoter sequence, in order to regulate the gene expression, called motifs. Thus, motif identification has become a crucial step for understanding the mechanism of gene expression.

Description of motif identification problem is: the m known sequences s = (s1,s2,...,sm), where si, i=1,2,...,m are composed of the elements in Δ ={A,C,G,T},with the length of n. si includes subsequence Ti with the length of n; subsequence T = (T1,T2,...,Tm) is generated by a variation of motif P. Namely these sub sequences have the same pattern and high similarity to each other. Referring to the metrics, pattern identification will search for T = (T1, T2..., Tm) from sequence sets to determine motif P. Motif identification has key steps including expression, evaluation and determination. These sequences have a unified mode P. They are from the variation of the same motif, with maximum similarity. Motif identification is to identify these unknown with similarity through some measurement criteria, thereby determining their motif.

CLASSIFICATION OF MOTIF REPRESENTATION METHOD

Motif is the region of protein or DNA sequence that has a specific structure. It is the special binding site for transcriptional factors (TF) in order to initiate the process of transcriptional regulation. It is important to identify the positions of motif in the upstream region of DNA. There are various ways to represent motifs.

Our method to predict motifs is to use a total fitness score function and to find the optimal motif using genetic algorithm. We also use the general genetic algorithm framework and operators to serve as our basic architecture as sequence alignment by genetics algorithm (SAGA) did. Following gives the description about some of the representations.

a) Consensus String:

A consensus string is formed from a collection of strings by taking the most common symbol appearing at each position of the strings. When constructed over a collection of genetic strings, the consensus string represents an average case organism over the collection. For example, if we sequence the same chromosome in a number of different individuals of the same species, then taking the consensus of these chromosomes gives us a notion of an average human chromosome. In fact, this was the idea employed by the researchers who first drafted the human genome in 2001 by taking an average case of 12 individual genomes. This is mostly applied in combinational algorithms and is very easy to understand. A protein binding site, represented by a consensus sequence, may be a short sequence of nucleotides which is found several times in the genome and is thought to play the same role in its different locations. The characters at each position in the consensus string is the base of highest frequency in the corresponding column of each motif sequence.

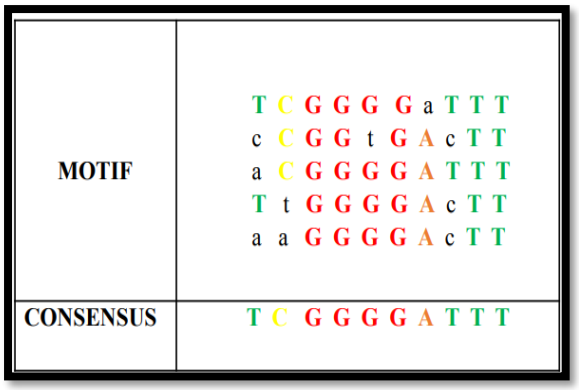


Fig 3: Above shows a Consensus String

b) Motif Pattern:

Motif is a conserved pattern that is common in two or more sequences and can be found in DNA, RNA and protein. The general concept of finding motifs is based on the occurrence frequency of a potential region in every sequence. In fact, it is quite difficult for all sequences to have a completely matched pattern, especially for consensus sequences with the number of base pairs exceeding 20. Hence, we have to allow mismatched nucleotides to some degree of extent. For example, if we allow a 20% mismatched nucleotides for motif pattern [ACCGGC], it is allowed for 1 nucleotide mismatched (6 x 20% = 1.2).

c) Fitness Score Function

First, we consider the computation of fitness score for one single sequence. 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:

$$FS(S_m, P_n) = \max_j \left\{ \sum_{i=1}^{k} match(S_{mji}, P_{ni}) / k \right\}$$

where

$$match(S_{mji}, P_{ni}) = \begin{cases} 1 & if \quad S_{mji} = P_{ni} \\ 0 & if \quad S_{mji} \neq P_{ni} \end{cases}$$

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 the motif pattern, j is the number of matched regions in the sequence. For example, suppose the motif pattern P1 and promoter sequence S1 are as follows:

P_1: *ACCGTA*
Promoter S_1: *A*TCCGGCTA*ACCGTA*CTATATTA

Fitness score: 3/6=0.5 6/6=1
FS(S1,P1) = 1.

d) Total Fitness Score Function:

The total fitness score function of a motif pattern is the summation of fitness score function for all sequences. It represents the score

of a motif pattern in each generation of the genetic algorithm. We define the total fitness score function as follows. 1 (,) (,) L n mn m TFS S P FS S P ¦ where L is the total number of sequences. For example, suppose the motif pattern P1 with 75% nucleotides match (allow 1 base pair unmatched), and promoter sequences S1 ~ S5 are as follows:

P_1: AGGAGGR
S_1: GGAGGAAGGAAGGAAGGAAGGAAGGAA=5.5/7
S_2: ACGAGGGACCAGGATGGCACCGCG=5.5/7
S_3: GAGCTCAAGGAGAAGATCGAGGAGATT=5.5/7
S_4: AAGTGCTATTAGGAGGAGAAAATCAAG=6.5/7
S_5: AGGCTGAGGCAGGAGGATTGCTTAAGG=6.5/7
$TFS(S.P_1)$=4.21

e) Finding Motif by Genetic Algorithm

In this algorithm, initial motif patterns with the same pattern length are created randomly. The users can set the pattern length. All the motif patterns will be different from the initial patterns after N-generation evolution. The purpose of crossover is to select the make child pattern for next generation. The mutation is done randomly to add variety to the motifs. Elitism is used to pass best parent motif to next generation. This is repeated until it reaches saturation point or after M generations.

```
generating 10 random motifs from each sequence.
looping over these random motifs (each motif is Pi) to find Total FItness Score.
  appending Score along with motif to a scoreTable.

looping over a range of N
  sorting the scoreTable based on the score.
  considering the last 50, i.e those with the most score.
  appending table got as output as a result of inputting score table to GeneticAlgorithm to finalTable.
  considering the last element of finalTables as our motif.

func GeneticAlgorithm (parentTable, count) -> childTable
  returns if count exceeds 50.
  generating motif list after crossovering parentTable.
  generating motif list as a mutation of above list.
  looping over the table
    finding the score for each sequence with each motif.
    appending score and motif to the child table.
  sorting child table
    replacing least 10 element of childTable with most 10 of parentTable
  sorting child table
  incrementing the count by 10 if score and motif repeats
  incrementing count by 1
  GeneticAlgorithm(childTable, count)
```

f) Operators:

The operators in FMGA are mutation, crossover and rearrangement that can speed up the FMGA to find the motif patterns:

- Mutation

Mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population. Here Patterns are mutated at random positions for

each sequence. Random position of each sequence is replaced by A, C, G or T.

- CrossOver

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. Crossover is sexual reproduction. Two strings are picked from the mating pool at random to crossover in order to produce superior offspring. The method chosen depends on the Encoding Method. The crossing is done over random parents to give offspring. Here, CrossOver is done by splitting the two parents in random positions and swapping their DNAs.

- Elitism

A strategy in evolutionary algorithms where the best one or more solutions, called the elites, in each generation, are inserted into the next, without undergoing any change. Worst 10 of the child table is replaced by the first 10 of the parent table.

RESULT

The experiment is also done based on Genetic Algorithm. In the implementation, the data is taken from Github.

After crossover, mutation and certain function in Genetic Algorithm, the motifs produced as output may be present at certain positions in the input DNA sequence. The Motif with highest score after M generations or if score repeats for n generations is taken as output.

CONCLUSION

Sequence motifs are short, recurring patterns in DNA that are presumed to have a biological function. Often they indicate sequence-specific binding sites for proteins such as nucleases and transcription factors (TF). Others are involved in important processes at the RNA level, including ribosome binding, mRNA processing (splicing, editing, polyadenylation) and transcription termination.

In the past, binding sites were typically determined through DNase foot printing, and gel-

shift or reporter construct assays, whereas binding affinities to artificial sequences were explored using SELEX. Nowadays, computational methods are generating a flood of putative regulatory sequence motifs by searching for overrepresented (and/or conserved) DNA patterns upstream of functionally related genes (for example, genes with similar expression patterns or similar functional annotation). For a while, it seemed like we had more computationally predicted sequence motifs without a known matching transcription factor, than transcription factors without a known binding sequence, although large-scale efforts to analyze the genome-wide binding of transcription factors using ChIP-chip are rapidly rectifying this situation.

The abundance of both computationally and experimentally derived sequence motifs and their growing usefulness in defining genetic regulatory networks and deciphering the regulatory program of individual genes make them important tools for computational biology in the post-genomic era.

The comprehensive discovery of DNA motifs is fundamental to the global understanding of gene regulation. DNA motifs, often represented as consensus sequences or position weight matrices, are common DNA sequence patterns bound by regulatory proteins (. One major type of regulatory proteins is transcription factors (TFs). Several hundred TFs may be active under an experimental condition. Multiple active TFs often bind short DNA regions of several hundred base pairs (bps) called cis-regulatory modules (CRMs) to control the temporal and spatial expression patterns of target genes. It is thus essential to identify motifs of all active TFs under an experimental condition to gain a global view of gene regulation under this condition.

REFERENCES

- https://examples.yourdictionary.com/examples-of-motif.html/

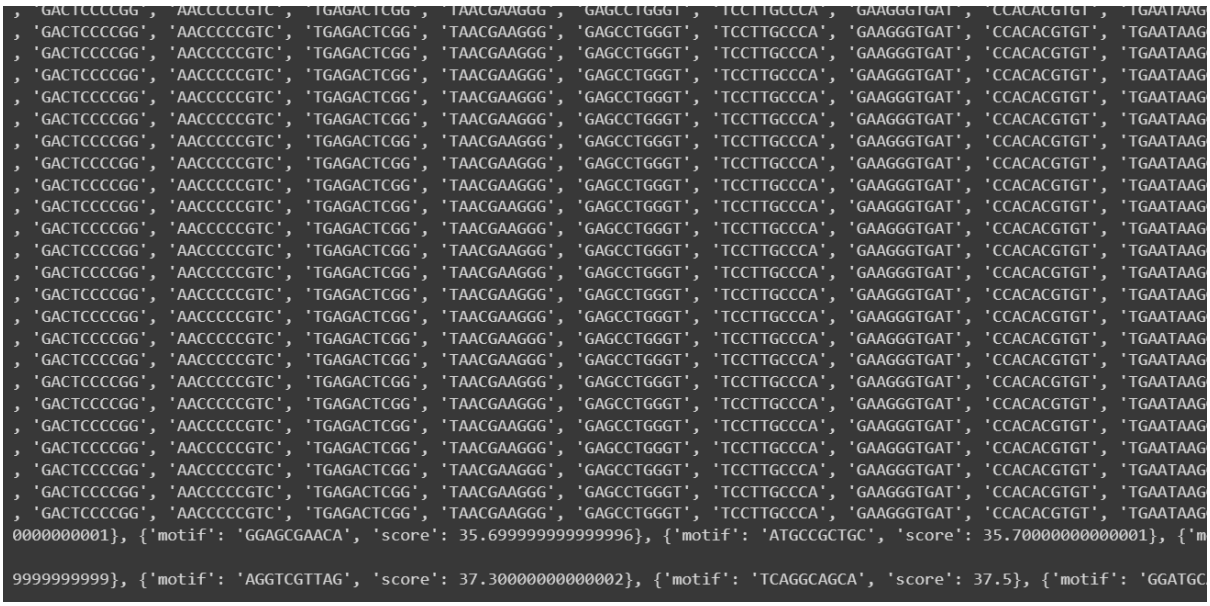
- https://www.sciencedirect.com/topics/engineering/genetic-algorithm/
- https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_fitness_function.htm/
- https://en.wikipedia.org/wiki/Consensus_sequence/

APPENDIXES

```
import random
def randomizedmotifsearch(lines):
    motifs = []
    for n in lines:
        for k in range(10):
            i = random.randint(0, 365)
            lmer=n[i:i+10]
            motifs.append(lmer)
            print(motifs)
    return  motifs
def FitnessScore(Sn, Pn):
    pnl = len(Pn)
    max_score = 0
    for i in range(len(Sn)-pnl+1):
        score = 0
        seq = Sn[i:i+pnl]
        for j in range(pnl):
            if Pn[j] == seq[j]:
                score+=1/pnl
        if score > max_score:
            max_score = score
    return(max_score)
def TotalFitnessScore(sequences, Pn):
    L, Ts = len(sequences), 0
    for i in range(L):
        Ts += FitnessScore(sequences[i], Pn)
    return Ts
def crossover(sTable):
    Table=[]
    for n in range(25):
        k = random.randint(2, 13)
        i = random.randint(1, len(sTable)-1)
        j = random.randint(1, len(sTable)-1)
        p1 = sTable[i]["motif"]
        p2 = sTable[j]["motif"]

        o1=p1[0:k+1]+p2[k+1:10]
        o2=p2[0:k+1]+p1[k+1:10]
        Table.append(o1)
```

```
        Table.append(o2)
    return Table
def GeneticAlgorithm(parentTable, count):
    if(count >= 50):
        return parentTable
    childTable=[]
    tables = crossover(parentTable)
    tables = [mutation(i) for i in tables]
    tables
    for i in tables:
        Pi = i
        score = TotalFitnessScore(sequences, Pi)
        childTable.append({"motif":Pi,"score":score})
    childTable = sorted(childTable, key=lambda x: x['score'])
    childTable[:10] = parentTable[-10:]
    childTable = sorted(childTable, key=lambda x: x['score'])
    if(childTable[-1]['score'] == parentTable[-1]['score']):
        count+=10
    print(childTable)
    print("-------------------------")
    count+=1;
    return (childTable, count)
file = open("sequences.txt")
sequences = file.readlines()
for n,i in enumerate(sequences):
    sequences[n] = i.strip()
sequences = sequences[:-1]
chars = ['A','C','G','T']
scoreTable = []

count = 0
finalTable = []
for i in randomizedmotifsearch(sequences):
    Pi = i
    score = TotalFitnessScore(sequences, Pi)
    scoreTable.append({"motif":Pi,"score":score})

for i in range(2):
    scoreTable = sorted(scoreTable, key=lambda x: x['score'])
    scoreTable = scoreTable[-50:]
    finalTable.append(GeneticAlgorithm(scoreTable, count)[-1])
#print(finalTable)
finalTable = sorted(finalTable, key=lambda x: ['score'])
print("Motif is : ",  finalTable[-1])
```

Output



Github Link

https://github.com/ANUSHKALA/GeneAlgorithm/