Finding Motifs from DNA Sequence Using Heuristic Approach

by

Shurid Shahriar Zaman
17101087
Tahseen Tayeb Kabir
17101417
Md. Al-Junaed Islam
17101438
Syed Md. Shamsul Alam
17301133

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shurid Shahriar Zaman

Shurid Shahriar Zaman
17101087

Tahseen Tayeb Kabir
17101417

Md. Al-Junaed Islam
17101438

Syed Md Shamsul Alam

Syed Md. Shamsul Alam
17301133

Approval

The thesis/project titled "Finding Motifs from DNA Sequence Using Heuristic Approach" submitted by

1. Shurid Shahriar Zaman (17101087)
2. Tahseen Tayeb Kabir (17101417)
3. Md. Al-Junaed Islam (17101438)
4. Syed Md. Shamsul Alam (17301133)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering.

Examining Committee:

Supervisor:
(Member)

Faisal

Faisal Bin Ashraf
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr.Md.Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Prof. Mahbub Majumdar
Chairperson
Dept. of Computer Science & Engineering
Brac Universitv

Mahbubul Alam Majumdar
Professor and Dean, School of Data and Sciences
Department of Computer Science and Engineering
Brac University

Ethics Statement

The thesis is carried out in complete compliance with research ethics norms, and the codes and practices set by BRAC University. In our thesis we use the data from primary sources. We are ensuring we use references and in text citations properly. We the four co authors take full responsibility for the thesis code violations. For solving problems we read different websites, youtube tutorials, and Questionnaire Free tools. We also took help from our university faculty members. Finally, we declare that we give credit every people from whom we took help. We did not make any fraud able means for completing the thesis. Our work is in compliance with the ethics standard set by BRAC university

Abstract

Finding patterns of the short sequences in DNA, RNA protein sequence has immense biological significance. The characterization and recognition of motifs is therefore an important method for a more in-depth understanding of genes or proteins in their structure, function and relations of evolution. This is one of the classical problems in the field of computational biology and which is an NP Hard problem. In this paper, we have proposed an evolutionary approach to get the motifs from DNA sequence by searching candidate motifs using heuristic way from the data. We have included various mutation techniques in an evolutionary approach and found an efficient way to calculate the fitness of our candidate motifs. We have evaluated the fitness of found motifs from our approach with benchmark data sets. Our method performs better results in terms of accuracy and specificity.

Keywords: Bio-Informatics; DNA; RNA; Heuristic; Motif; Algorithm

Dedication

We would like to dedicate this research to our parents. Without their support we may not able to do our studies. We also want to dedicate the research to our friends who helped us to do better. Our supervisor helped us throughout the year. We want to dedicate this to him also.

Acknowledgement

All praise to the Great Allah for whom our thesis have been completed without any major interruption.

Firstly, we would like to say thanks to our beloved family members whom we will always be indebted to.

Secondly, we thank our supervisor Faisal Bin Ashraf for all his help and constant guideline

Finally, we would like to thank all the faculty members and stuffs for providing us with such a wonderful study environment where could develop ourselves as well as this research properly. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ANN	Artificial Neural Networks
CS	Cuckoo Search
DNA	Deoxyribonucleic acid
EC	Evolutionary Computation
EMS	Edit Distance Motif Search
FS	Fuzzy Structures
GA	Genetic Algorithm
$mRNA$	messenger RNA
PMS	Plant Motif Search
PSO	Particle Swarm Optimization
RNA	Ribonucleic acid
SI	Swarming Intelligence
SMS	Simple Motif Search
TF	Transcription Factor
YMF	Yeast Motif Finder

Chapter 1

Introduction

1.1 Background

Cells area is the littlest living units of organisms. There are common three things in a cell. Every cell encompasses a membrane that differentiates within cell from the surroundings. Other things are cytoplasm which is a jelly like fluid and last things is DNA which is the genetic component of a cell. Prokaryotic and eukaryotic cells are two broad categories of a cell. Eukaryotic cell has organelles (special parts of a cell) and it can be found in plants and animals. On the other hand, prokaryotic cells have enclosed organelles and have no nucleus and also one celled. Cells have different kinds of functions like providing a shape of a body, intake nutrient from different food and these nutrients are converted into energy. Cells are comprised with different parts like cytoplasm, Golgi body, mitochondria, lysosome etc. Your introduction goes here! Some examples of commonly used commands and features are listed below, to help you get started[1].

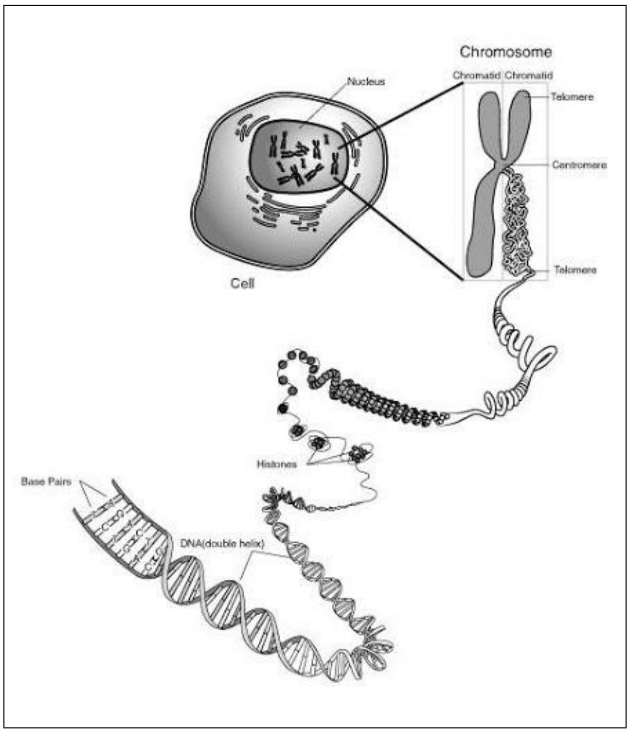


Figure 1.1: Cell of a Human Body

Chromosome is like a chain composed of DNA belonging with the cumulation of reposing protein. Chromosome is like fiber, thread which is extremely long where millions of nucleotides long in addition with hundreds of genes. There are lots of functions of chromosome like the basic genetic materials are comprised in chromosome. In addition, chromosome helps in cell division like mother cell is divided into daughter cells.

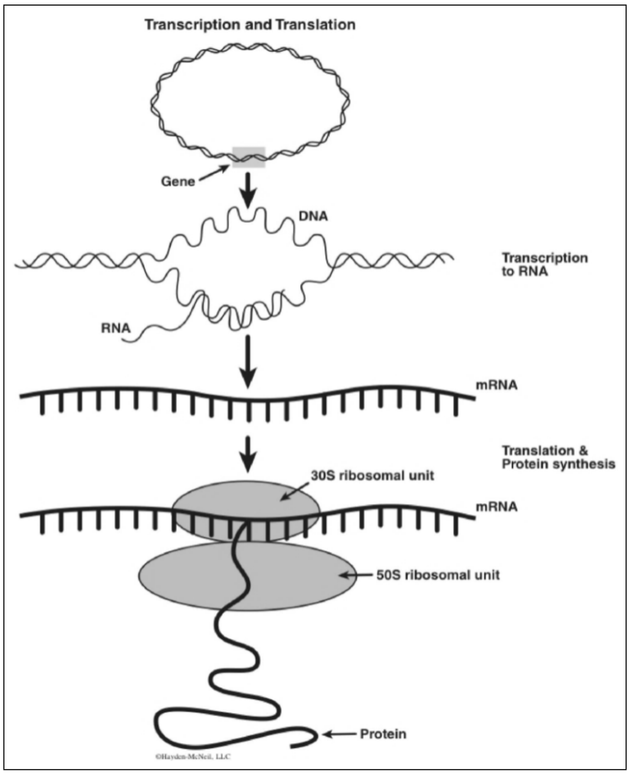


Figure 1.2: Chromosome of Human Body

Furthermore, it also contains histone and non-histone protein that helps wielding gene action in a body. Chromosomes are present in every living body. Every living body has different numbers of chromosomes. Chromosome can be of different types like metacentric, sub-metacentric, acrocentric and telocentric chromosome[2].
DNA refers to Deoxyribonucleic acid that is like a molecule which is present inside the cell nucleus. DNA is the primary biological function in a human body. Storing and coding the genetic information of a body is the main function. There is a very close connection between chromosome and DNA. DNA is allocated in nucleus of a cell in the life cycle of a cell but when DNA starts to replicate, they form a structure like chromosome which helps to stabilize DNA. DNA is actually a polymer with nucleotides. In the basic structure of nucleotide there are three important things which is phosphate, sugar and nitrogenous base. There are four types of nucleotides present which is Adenine, Guanine, Cytosine and Thymine. These four nucleotides are divided into two groups, pyrimidines and purines. Adenine and Guanine are in purines and Thymine and Cytosine are in group of pyrimidines[3].
The process of central dogma can be divided into some processes. Firstly, replication helps to make copies of DNA and then the code of DNA is broken into every cell in post replication which is transferred into a mRNA. After that, reading this mRNA, the code is translated into proteins. This whole process is central dogma. Replica-

tion is basically semiconservative in nature. The catalyst DNA enzyme copies one parental molecule of double stranded DNA into two female offspring molecules of dual stranded DNA. Transcription is rendered from DNA through RNA. The RNA polymerase enzyme produces an RNA molecule complementing a gene encoding portion of DNA. Transmission is produced from mRNA protein[4].

1.2 Problem Statement

DNA motif is basically a sequence pattern with sequence of nucleic acid that has regulatory protein of DNA bonding sites that can be also called Transcription Factor (TF). DNA motifs basically assembles with structural motifs where proteins can be present. Though motifs occur on double stranded DNA also double stranded DNA is bind by TF. Motif sequences can be of different like zero, one or two or many more motifs.

Motif discovery is defined as the question of motifs being found without previous awareness of what the motifs look like. A double helix DNA with a single strand can be mentioned as a string over an alphabet F = A, T, G, C. Arising of motif discovery happens while DNA contains times of binding motifs and they may be unknown. Let us see a short example on finding motif from a DNA sequence.

AGGTACA**CTCA**TGATGCACCTGTA

CTTGATTCACATGA**CTCA**TGACAT

CCGTAACTGCTTGCA**CTCA**AACAT

TGTTAGGA**CTCA**TCACACGACAAT

GAGT**CTCA**CTGATCTGAGTCAGAA

In this above example, there are 5 different DNA sequences. By iterating them we get a common string which is repeated in every sequence i.e. CTCA. Therefore, CTCA is one of the motifs in the above DNA sequences.

Mutation is a frequently occurred phenomenon on DNA sequences. So, some random positions of DNA for a particular species can be different from individual to individual due to mutations over time. Keeping this obvious phenomenon in mind, we try to find motifs by allowing a degree of mutation, i.e, we will overlook some mismatches in the sequences. This problem is called (l, d) motif finding. Let us see an example of finding a (l, d) motifs from given sequences of DNA:

l = length of the motif

d = number of maximum mismatches

TTCGGACGAATGC**CAGTTC** — mismatch with "AAGTTC" = 1

CAGGTCTTGACCGGAATGC — mismatch with "AAGTTC" = 2

AATGGCTCCATGG**CAAGTTC** — mismatch with "AAGTTC" = 0

GTAACCT**TAGCTA**AGGCAAT — mismatch with "AAGTTC" = 3

T**AACTTG**GTTAGGCCATTCG — mismatch with "AAGTTC" = 2

Here, "AAGTTC" can be $(6, 3)$ motif for this above dataset in every DNA sequence where the subsequences of length 6 and the maximum number of mismatches is 3.

1.3 Motivation

Motif Discovery is an NP-Hard topic as it requires infinite time to discover motifs there is space to minimize run-time. Besides this, discovery of large-length motif is still a strong field for study as discovering large-length motif is still a major challenge because it requires up too many days for computation. The recent advent of technologies such as combination of chromatin immunoprecipitation (ChIP[5]) with tiling arrays (ChIP on ChIP[6]) or next-generation sequencing (ChIP-Seq[7]), that possesses the direct genome-wide identification of regions surrounded in vivo with the help of a defined TF. ChIP-Seq becomes the actual norm during this space quickly that rises new issues for the algorithm and device developers, that may be directed within the following. The input is a series of DNA sequences and our aim is identifying one or more motifs where one or many oligos that happens in a wide fraction of the sequence. Oligos belonging to the same pattern should be very similar to one another, probably recognized by the same TF as linking sites. It is generally assumed that the size of the motif is known as a priori. The motif does not seem with the constant frequency and indiscriminately created sequences in any model containing the biologically possible DNA sequences so as to assess the particular that means of the motif and to discriminate against random similarities.

1.4 Research Objective

Motif finding problem takes exponential time to solve. So, our objective is to propose an algorithm for planted motif problem with heuristic approach. This algorithm will help to reduce execution time. Besides, motif finding on different data samples and mismatches from DNA sequences is also our primary concern.

(a) Finding simple motif search from DNA sequences

(b) Using nature inspired methods for optimization of calculation

(c) Use variant data sets for justification

Chapter 2

Related Work

2.1 Literature Review

The detection of these reasons is an essential challenge for molecular biology. The motif finder consists of the class MotifFinder and findMotif method, four separate motif finding algorithms, two heuristic PROJECTION or ePatternBranching algorithms and two similar Algorithms PMS1 and PMSP, gives an in-depth knowledge of structure, role and developmental connections of the genes or proteins.
The vast amount of data from DNA, RNA and the protein sequence has been created by several genome projects. While computational pattern search methods like BLAST have made significant observations such as protein modularity, even more research is available with biological sequence details. However, because of their length and sophistication, we are limited by the computational algorithms used to classify motifs. The three main paradigms historically used for recognizing small, functional patterns of peptides, transcriptional regulatory components, composite regulatory patterns, diagrams of DNA, differences between the protein families etc. are SMS, PMS and EMS. There are two major shortcomings in current pattern-search algorithms for motif-search:

1. Approximate algorithms do not always recognize the correct pattern, but have the advantage that they can be used to search for short and fairly large patterns in large data sets such as genomes.

2. Exact algorithms, on the other hand, often recognize the appropriate pattern, but cannot be used to classify complex data patterns in large datasets. To derive more sophisticated patterns from genomic data we need precise algorithms which can be used to analyze genomes with appropriate computational resources for complex patterns.

We divide the motifs accessible for algorithms into three major groups based on the typology of DNA sequence knowledge used in the template deduction algorithm:

1. those that use single genome coregulated gene sequences,

2. those that use single gene orthological sequences of many species (i.e., phylogenetic foot printing) and

3. those which use prom sequences.

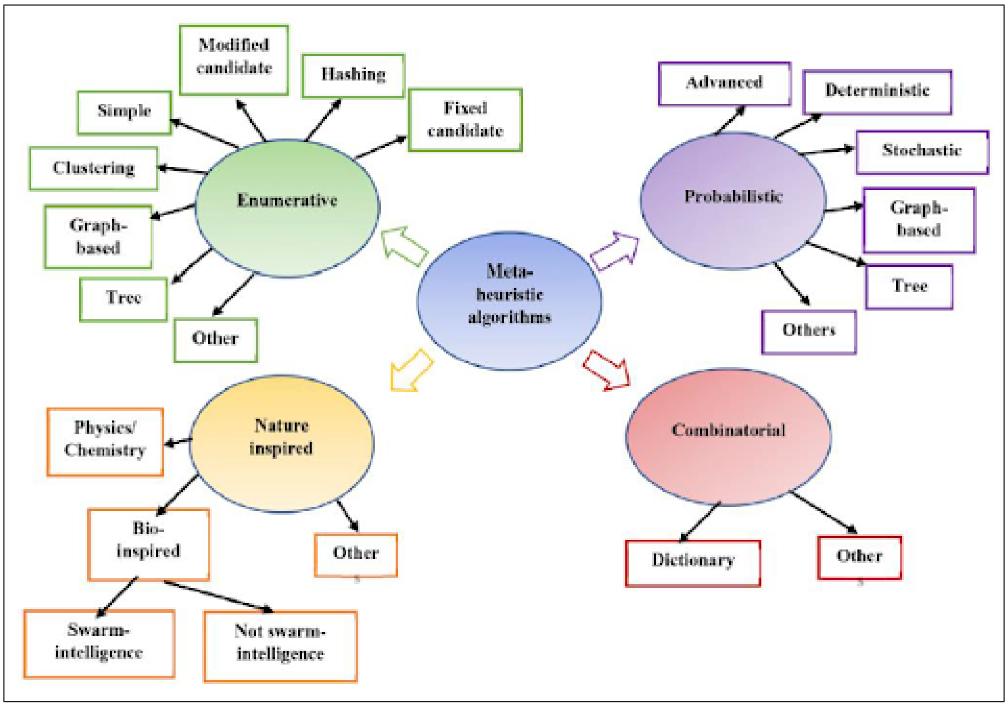


Figure 2.1: Classification of algorithms for motif finding as nature driven, probabilistic, enumerative and combinatorial forms.

Most previous literatures however, classified algorithms of motif searching in two key groups according to the combinatorial methodology employed in their design:

1. word-based methods (string-oriented) mostly focused on full description, i.e. oligonucleotide frequency count and contrast, and
2. probabilistic models with estimates of the function parameters by maximum probability.

2.1.1 Word Based Method

As we know that the word-based methods are mostly enumerative methods so that we can say that they ensure global optimality and by using the eukaryotic genomes they are useful for identifying motifs where motifs are typically shorter than prokaryotes. These methods are also ideal for short motifs. When introduced with structured data structures such as suffix trees, word-based approaches may also be very efficient and are a strong choice in identifying completely restricted motifs, implying that all instances are similar. Nonetheless, for standard transcription factor motifs we can say that word-based approaches may be troublesome. Sometimes they have many weakly restricted locations and the product also needs to be post-processed with any clustering framework. The issue of generating so many incorrect explanations is often exacerbated by word-based methodologies.
Oligo-Analysis is one of the motifs finding algorithm which is based on the word-based methods developed by van Helden[8]. Though conceptually clear, their algorithms have proven successful in eliminating motifs from the majority of regulatory families in yeast(Saccharomyces cerevisiae). In the experimental laboratory research, such motifs had already been discovered. In addition, in the upstream

regions of the coregulated genes, putative new regulatory sites were expected. This oligonucleotide analysis is systematic and detailed in comparison to heuristic approaches. However, its detection spectrum is restricted to basic designs with short motifs with a tightly preserved core.
Sinha and Tompa built an YMF algorithm utilizing the same methodology[9]. The concept was extracted from an analysis of the identified yeast binding sites for the transcription factor. The algorithm inputs consist of a collection of upstream sequences, the number of non- in the motifs to be identified, and the transformation matrix for the order m of a Markov chain formed from the completeness of upstream yeast sequences.

2.1.2 Probabilistic Based Method

The probabilistic method includes a location weight matrix describing the configuration of the motif. The weight matrices of the position are also shown as the pictogram describing increasing position by a stack of letters whose height is commensurate with its knowledge quality. Probabilistic approaches have the benefit of having little criteria for searching but focus on probabilistic approaches of the regulatory regions that are highly responsive to slight input data shifts. Some of the probabilistic algorithms aimed at identifying more or less general explanations that are required for binding sites of transcription. On the other hand, in prokaryotes, motifs are normally lengthier than eukaryotes; they are more suitable for identifying motifs. Such algorithms are not, however, assumed to find ideal solutions internationally, because they use a form of local analysis[10].
One of Hertz's[11] first attempts to find a matrix describing transcription factor connections for the most info-intensive framework was a greedy probabilistic sequence model-based algorithm. This method was used to identify a specific purpose in an increasing series once.
Lawrence and Reilly implemented the EM[12] for motif searching and generalized Hertz greedy algorithm for pattern searching. It was built specifically for protein motives, but may also be used for discovering DNA motives. There is no site synchronization needed, so the basic principle of the model is that at least one specific site will occur in each series.
Gibbs sampling approach[13] has been commonly used for the detection of algorithms among the probabilistic approaches. Below is a brief overview of Lawrence first Gibbs sampler for motif finding. This method was not extended to the DNA sequence but was extended in the original article to the protein series. Provided that at least one case of a topic occurs in any sequence, one of the original principles of this algorithm is the technique often called the “location sampler”.

2.1.3 Natured Inspired Method

Natural algorithms represent a collection of novel techniques and solutions to problem-solving and draw tremendous exposure to their successful results. Nature-inspired algorithms have been used as model models for various real-world problems, which include ANN, FS, EC and SI[14]. While nature-inspired algorithms are so common, there still are many challenges that need to be further investigated. The strength of natural algorithms is the simplicity in determining solutions by the use of exercise functions to offer solutions. Such characteristics vary from problem to problem and determine the usage of various forms of knowledge such as biological, electronic, etc. Moreover, the motif representation of such algorithms is versatile.

Heuristic approach is a way of prioritizing the paths from the initial state of an algorithm to the goal state or final state over other paths in that particular algorithm. This is for finding a solution of a problem and the solution of the problem is computed in the last state or goal state.

Basically, heuristic approach is a process to discover the solution of a problem. It is a shortcut process to produce good enough solutions. In this approach, knowledge is secondary thought. This approach is basically a rational system that does not necessarily need to be accurate or optimal and it is a flexible process for quick decisions, particularly when working with data which are complicated.

Heuristic approach[15] varies quite a lot depending on the problem. So, it is important to first understand the problem and then go for the solution. This approach has four principles that make a structure for solving a problem.

Firstly, try to understand the problem. It really helps to create an image of the given problem and to look at it from different point of view. Questions like “What is the issue” or ”What’s going on” or ”Can the issue be clarified at the end of the day” or ”Is there enough data accessible” and many more questions like these types can help to solve a problem. Second step is to make a plan. There can be different ways to solve a problem. Main thing is to picking up the right process which fits the problem given. When people assume that they have a solution of the problem given then they use “back tracking” as a beginning stage to move in the direction of that specific problem. It can likewise be valuable to cause a diagram of the conceivable outcomes, to erase some of them promptly, work with correlations, or to apply balance. Imagination becomes possibly the most important factor here. Thirdly, carry out the plan. When a system has been picked, the plan can rapidly be executed. Nonetheless, it is important to focus on schedule and show patience because the solution won’t just show up. If the plan doesn’t go anyplace, the guidance is to ignore the plan and go for another plan to solve the problem. Lastly, evaluate and adapt. Set aside the effort to deliberately consider and ponder the work that is now been done. The works that are going according to plan should be maintained, those leading to a lesser solution, should be adjusted. Few things will simply work, while others may not.

Heuristic approach provides a quick solution which is straight forward and easy to implement. As Heuristic approach is practical so it serves as quick and feasible short-term answers to scheduling and planning issues. It can lead ease of use testing to additionally analyze potential issues. This approach can utilize together with other ease of use testing philosophies. Assigning the right Heuristic approach can help propose the best corrective measures[16].

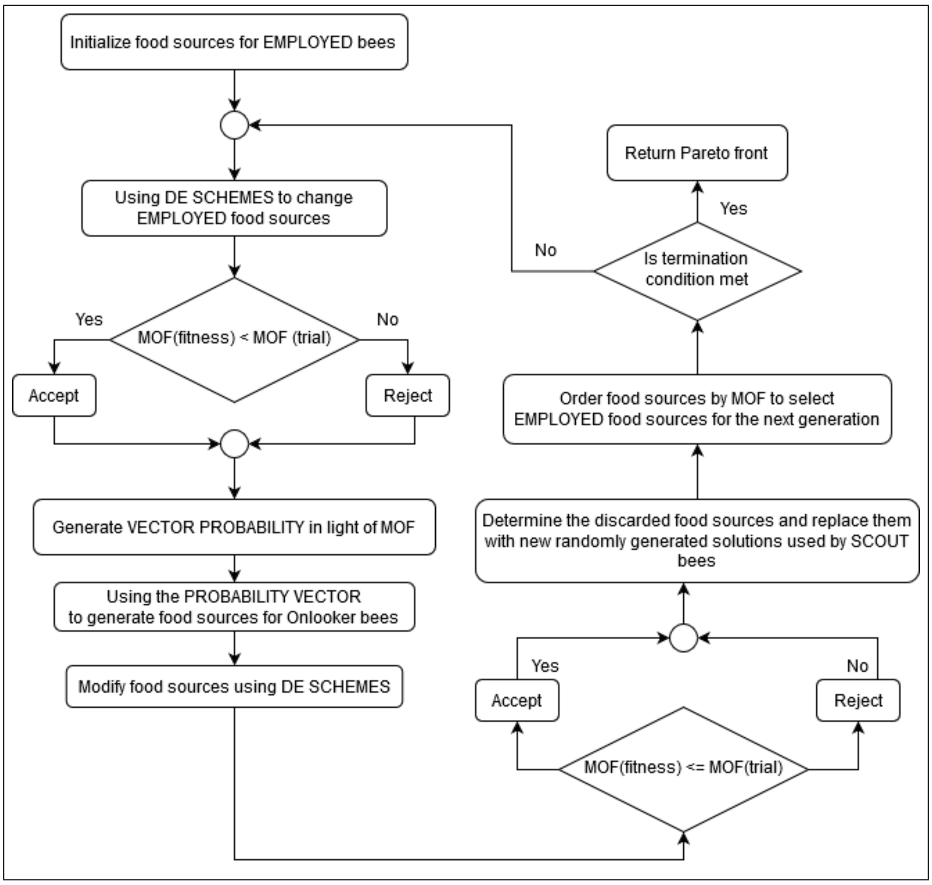
Here is a list of some heuristic algorithms[17] from different types

1. ABC: MOABC
2. ACO: MFACO
3. CS: MACS
4. GA: FMGA, MOGAMOD
5. PSO: PMbPSO, PSO+

2.1.3.1 ABC[18]

Artificial Bee Colony algorithm could be a kind of swarm-based algorithm which Karaboga proposes. To search out a food supply, the action of honey bees simulates by this algorithm. 2 elementary properties in honey bee colonies for getting swarm intelligent behavior are self-organization and division of labor. The bee colony includes 2 categories that are working and the foragers that are unemployed. Employed bees work is move to food sources which was previously visited. They are answerable for providing information about the standard of allotted supply to unemployed foragers. The employed bee that develop into scout bee to seem for a brand-new food source. New food source's fitness is decided and also greedy choice is applied.

2.1.3.1.1 MOABC/DE[19]



Initialize food sources for EMPLOYED bees
Using DE SCHEMES to change EMPLOYED food sources
Yes
MOF(fitness) < MOF (trial)
No
Accept
Reject
Generate VECTOR PROBABILITY in light of MOF
Using the PROBABILITY VECTOR to generate food sources for Onlooker bees
Modify food sources using DE SCHEMES
Return Pareto front
Yes
No
Is termination condition met
Order food sources by MOF to select EMPLOYED food sources for the next generation
Determine the discarded food sources and replace them with new randomly generated solutions used by SCOUT bees
Yes
No
Accept
Reject
MOF(fitness) <= MOF(trial)

Figure 2.2: MOABC/DE Flowchart

González et al proposed MOABC/DE. This algorithm is meant to adapt the Artificial Bee Colony algorithm to a multi-objective context. At the similar time, the multi-objective optimizes over one objective function to induce a collection of best solutions referred to as the Pareto package. This algorithm describes 3 competing objectives as getting used ABC's motif length, help, and similarity and multi-objective adaptations together with multi-term fitness function, ranking, and sorting methodology.

2.1.3.2 ACO[20]

Ant Colony Optimization is a heuristic based algorithm which is based on the real behavior of ants which always try to search for the shortest path from their nest to the source of the food. While moving, the ant always leave a path of chemical factor which is called pheromone on the bottom and it helps the ants to induce to nest. They move through that chemical component with one another. The quantity and quality of the food depends on pheromone as the quantity of it is proportional to both and also this chemical component for the food supply would be directed to different ants. After sometimes the evaporation of pheromone occurs which decreases its attractive strength. Within shorter direction the evaporation takes an extra time than the longest. With high concentrations of pheromones, the ants choose their way.

2.1.3.2.1 MFACO[21]

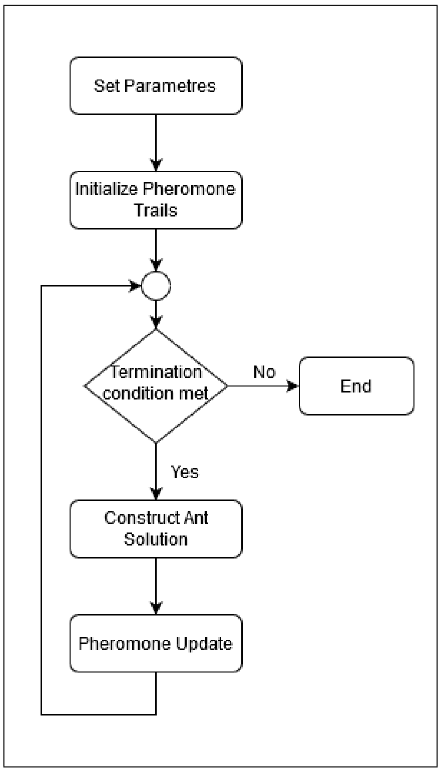


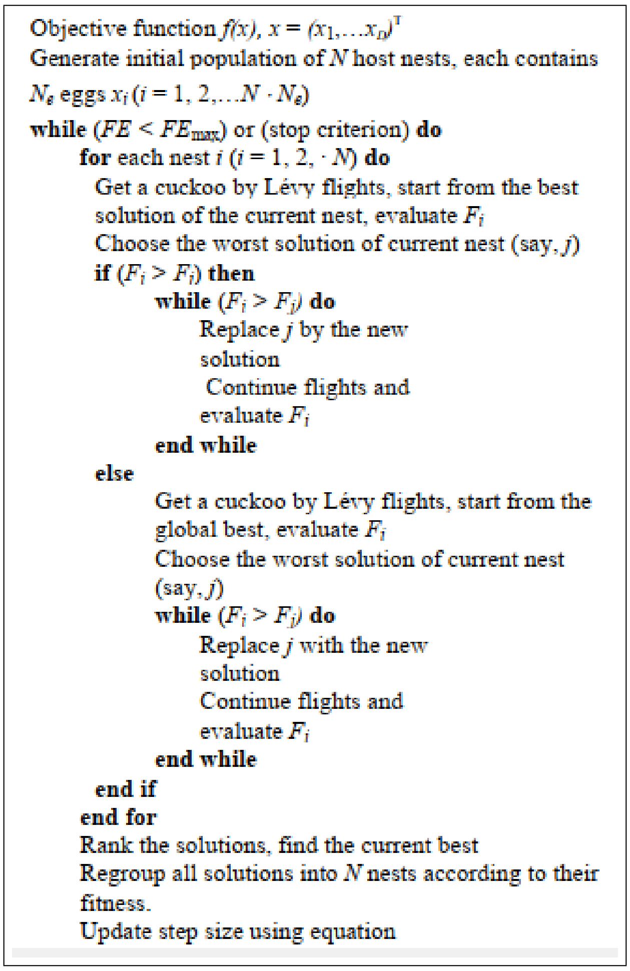
Figure 2.3: Flowchart of MFACO

Ant Colony Optimization finds beginning positions which is better than random initialization of DNA sequences given as beginning position for the GS method. ACO algorithm starts where each and every ant selects the trail to form motif length (m) as sample. This sample depends on the pheromone's probability. After that, to induce the set that represents the most effective matching substrings, comparison between the ants and the chosen sample (m) occurs and each substring within the input sequences. First, it measures the fitness function for each chosen set. The amount of pheromone is then modified and eventually iterated until no adjustment is created.

2.1.3.3 CS[20]

CS could be a heuristic algorithm that is a lot of efficient than GA and PSO. CS is inspired by the breeding mutuality of some species of cuckoo together with the behavior of Lévy flight. Cuckoo lays their eggs in alternative birds' nests with power to choose recent nests and destroy established eggs and also to extend the prospect of their eggs hatching. If such eggs are found by host birds, they throw the eggs away or build another nest to live there. This species follows a rule: One egg at a time laid by each cuckoo and disposes of its egg in a selected nest choosing randomly. The nests containing high-quality eggs which is denoted as solutions are the strongest and can still the generations that follow. Each and every egg could be a solution as it helps to simulate cuckoo reproduction and every egg denoted as new solution.

2.1.3.3.1 MACS[22]



```
Objective function $f(x)$, $x = (x_1,...x_D)^T$
Generate initial population of $N$ host nests, each contains
$N_s$ eggs $x_i$ ($i$ = 1, 2,...$N \cdot N_s$)
while ($FE < FE_{max}$) or (stop criterion) do
    for each nest $i$ ($i$ = 1, 2, · $N$) do
      Get a cuckoo by Lévy flights, start from the best
      solution of the current nest, evaluate $F_i$
      Choose the worst solution of current nest (say, $j$)
      if ($F_i > F_j$) then
            while ($F_i > F_j$) do
                  Replace $j$ by the new
                  solution
                   Continue flights and
                  evaluate $F_i$
            end while
      else
            Get a cuckoo by Lévy flights, start from the
            global best, evaluate $F_i$
            Choose the worst solution of current nest
            (say, $j$)
            while ($F_i > F_j$) do
                  Replace $j$ with the new
                  solution
                  Continue flights and
                  evaluate $F_i$
            end while
      end if
    end for
    Rank the solutions, find the current best
    Regroup all solutions into $N$ nests according to their
    fitness.
    Update step size using equation
```

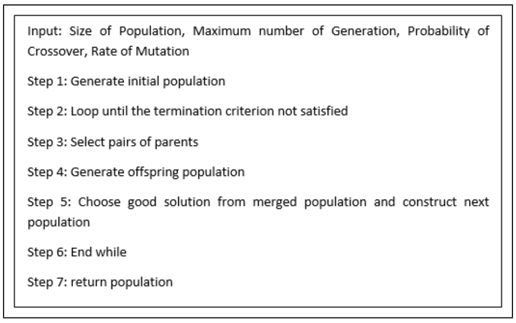
Figure 2.4: Pseudocode of MACS

The MACS algorithm improves the fundamental Cuckoo Search algorithm by combining parallel, incentive, information and adaptive strategies. There are multiple subgroups consisting of the population. Subgroup environments will increase bound lower solutions' survival rate, as weaker solutions will thrive during a better fitness subgroup. to spice up population diversity on the premise of grouping, each generation's cuckoo starts from the subgroup's best solution. When the individual subgroup is worse than the new solution, then they will replace it with new solution. This technique is performed on each subgroup, to make equal with the quantity of cuckoo's activity parallel search. An honest flight (replacement solution's fitness is larger than the worst within the subgroup) wins as a successive flight for each cuckoo in a very subgroup, before the flight fails. This strategy aims at exploiting promising space at a quicker speed. A basic technique is employed to realize high-level sharing of information with low cost computation. Sort the whole population consistent with fitness value at the top of every generation, continuously creating a new subgroup by dividing them. By this, the last generation of the most effective and worst solutions is shared between groups, making new groups with other solutions. This technique preserves the echelon development of fine and poor solutions, such solutions at totally different levels conjoin within the population. It appears a rational option to alter the search phase size slowly with the search method at different stages of the search. The possible access to information of the whole space is required within the early stage of the search. a bigger step-size is needed at this stage. Within the later stage, the search should be allotted during a small neighborhood of individual so as to enhance precision, and a small step size is then needed.

2.1.3.4 GA[23]

In Genetic Algorithm genes, chromosomes, genotype and phenotype this terminology is being used. Firstly, in Genetic algorithm we have initial population. Diversity is the main part of initial population. By using initial population, we calculate the fitness. As the compare of input, the best given output is the fitness function. The next step is selection. By using fitness function, we improve our selection to find out the best parent. The next part is Crossover. By using crossover, we generate a new child and check the fitness function of it. The final part is mutation. In mutation there are different types of parts as like swap and bit flip. We use it because we want to create better population than the previous one. We will follow this sequence until the stopping criteria will fulfill.

2.1.3.4.1 MOGAMOD[24]



Input: Size of Population, Maximum number of Generation, Probability of Crossover, Rate of Mutation

Step 1: Generate initial population

Step 2: Loop until the termination criterion not satisfied

Step 3: Select pairs of parents

Step 4: Generate offspring population

Step 5: Choose good solution from merged population and construct next population

Step 6: End while

Step 7: return population

Figure 2.5: MOGAMOD Steps

In Step 1, generate initial population. In Step 2, there is a loop and this loop will continue until the criterion of termination is not being satisfied. In step 3, the set of parent solution pairs is being selected. After that, step 4 comes where crossover and mutation are applied to provide offspring population. To choose the right solutions from combined population, the next population is made. From step 3, to check every answer within the current population Pareto-dominance and situation is used. In step 5, doing the same thing for combining population and also select the most appropriate solution as representative of subsequent cluster from the consolidated answer to introduce Elitism.

2.1.3.4.2 FMGA[17, 25]

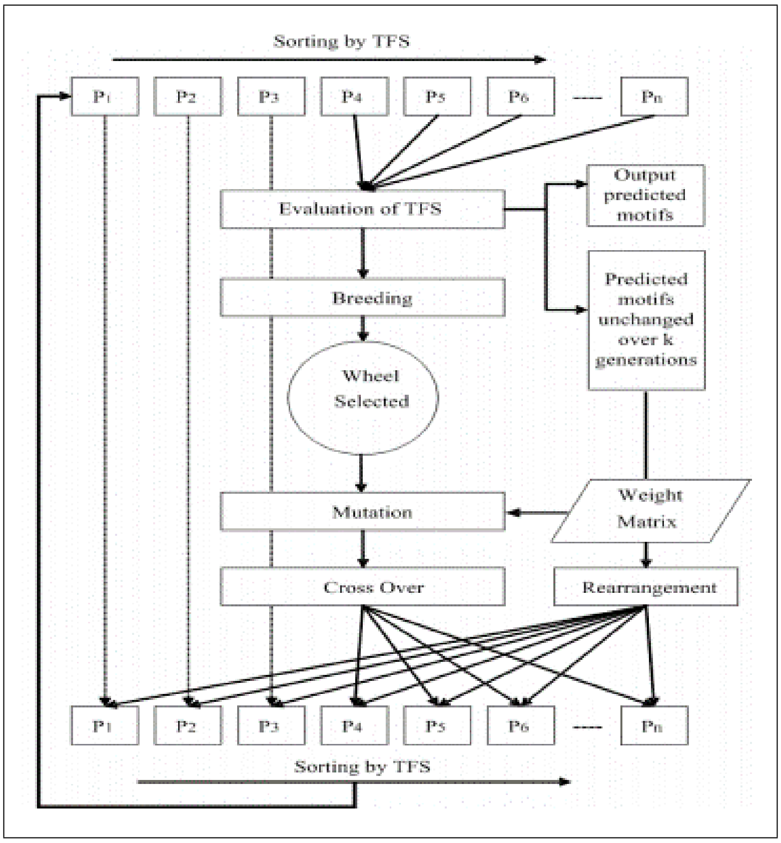


Figure 2.6: FMGA Architecture

FMGA algorithm is used for location motive position within the areas from upstream bp of -2000 to downstream bp of +1000 of the beginning site transcription. The mutation in genetic algorithm takes place with location weight matrices to ensure that all retained positions are not discarded. The crossover is performed to include the desired child pattern of uncommonly designed violation penalties. In fact, to remain far from the vicinity of a fully consistent local minimum, this genetic algorithm uses one of the remodeling methodologies intrigued by place weight matrices that may render the desired design quite complicated for specific administrators. The writers have declared FMGA to be more successful in comparison to GS and MEME.

2.1.3.5 PSO[26]

One of the popular techniques of Swarm Intelligence is Particle Swamp Optimization. Easy computation and the sharing information among the algorithm is distinguished in PSO. It also simulates the behaviors of movements of organisms in bird flocks along with schools of fish to seek out sources of food and defenses against enemies. That particle shares its own flying experience with other particle and vice versa to alter their ”flying” therefore it incorporates self-experience and social experiences. The particle tries to induce the most effective local particle position in self-experiences and this is often done by the stores of the particles. The most effective solution seen in its memory up to now is termed ‘pbest’, and because it navigates through the search space of the answer, it’s an attraction towards this solution.

2.1.3.5.1 PMbPSO[27]

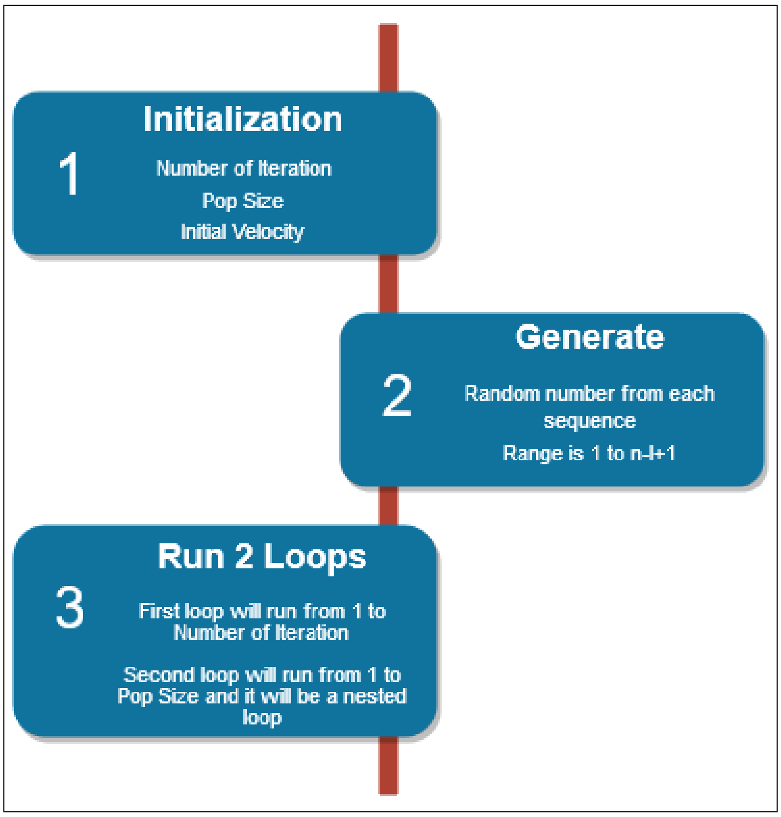
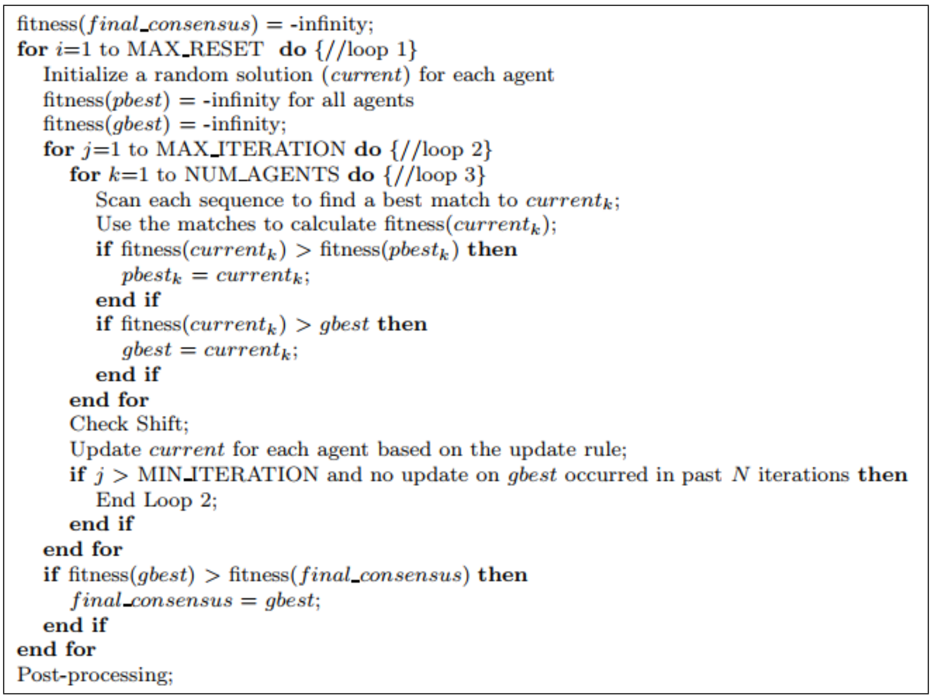

Initialization
1
Number of Iteration
Pop Size
Initial Velocity
Generate
2
Random number from each sequence
Range is 1 to n-l+1
Run 2 Loops
3
First loop will run from 1 to Number of Iteration
Second loop will run from 1 to Pop Size and it will be a nested loop

Figure 2.7: PMbPSO Steps

Reddy et al developed this algorithm. For all motifs, this algorithm selects initial positions and develops 10 children for every motif. After that, evaluation of fitness function has to done for every parent and their children as it can generate the most effective position; at now, the simplest position of all particles is following the rate and site of every single particle for variety of iterations.

2.1.3.5.2 PSO+[28]



```
fitness($final\_consensus$) = -infinity;
for $i$=1 to MAX_RESET do {//loop 1}
  Initialize a random solution ($current$) for each agent
  fitness($pbest$) = -infinity for all agents
  fitness($gbest$) = -infinity;
  for $j$=1 to MAX_ITERATION do {//loop 2}
    for $k$=1 to NUM_AGENTS do {//loop 3}
      Scan each sequence to find a best match to $current_k$;
      Use the matches to calculate fitness($current_k$);
      if fitness($current_k$) > fitness($pbest_k$) then
        $pbest_k = current_k$;
      end if
      if fitness($current_k$) > $gbest$ then
        $gbest = current_k$;
      end if
    end for
    Check Shift;
    Update $current$ for each agent based on the update rule;
    if $j$ > MIN_ITERATION and no update on $gbest$ occurred in past $N$ iterations then
      End Loop 2;
    end if
  end for
  if fitness($gbest$) > fitness($final\_consensus$) then
    $final\_consensus = gbest$;
  end if
end for
Post-processing;
```

Figure 2.8: PSO+ Pseudocode

PSO+ differs from different motifs by directly modeling gaps to seek out algorithms, that offers a straightforward approach to seek out gapped motifs. This algorithm uses each consensus and representations of position-specific weight matrix with the help of advantage of the efficiency of consensus and also the accuracy of PWMs. This technique gives permission of some input sequences which can contain zero or many binding sites, that are popular in real data set, however a number of the algorithms ignore.

Chapter 3

Proposed Method

We are using heuristic approaches, especially genetic algorithms for finding motifs. For finding the best motif we take initial population from the dataset, find the fitness score of that population, access population for finding the best population and after doing mutation we get the final output. We will update the fitness score after getting the better fitness value. We set a limit to run the iteration to get the best motif in the shortest possible time.

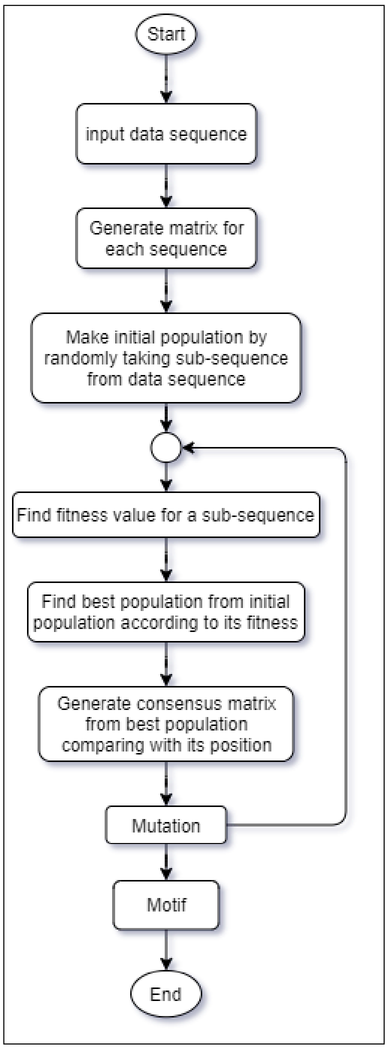

Start
input data sequence
Generate matrix for each sequence
Make initial population by randomly taking sub-sequence from data sequence
Find fitness value for a sub-sequence
Find best population from initial population according to its fitness
Generate consensus matrix from best population comparing with its position
Mutation
Motif
End

Figure 3.1: Flowchart of Our Proposed Model

3.1 Data Representation

In our proposed method, we are trying to calculate the positions of each nucleotide in a sequence of data. We can easily do it by building a matrix. In a matrix, there are 4 rows which are four nucleotides and the columns are the representation of the given sequence. We are using binary formula and if the index of row and column matches then that [row, columns]'s value will be 1 otherwise 0. In this way, we get the position of each nucleotide and this will help us to do our future work like mutation. As an example, we have 10 lengths of subsequences which is "GCTATCGGAT" the matrix representation is looking like:

	G	C	T	A	T	C	G	G	A	T
A	0	0	0	1	0	0	0	0	1	0
C	0	1	0	0	0	1	0	0	0	0
T	0	0	1	0	1	0	0	0	0	1
G	1	0	0	0	0	0	1	1	0	0

Table 3.1: Data Representation

Also, we are trying to represent our candidate motif in the same way. We are doing it to get more motifs easily after doing the crossover and mutation. As an example, we take 4 subsequences of candidate motif which length is 6 then the matrix is looking like:
Subsequences:

ATCGGA
TGCTAT
AGTTAG
CTGCTG

The representation will be:

	0	1	2	3	4	5
A	0.5	0	0	0	0.5	0.25
C	0.25	0	0.5	0.25	0	0
T	0.25	0.5	0.25	0.5	0.25	0.25
G	0	0.5	0.25	0.25	0.25	0.5

Table 3.2: Subsequences Representation

3.2 Build an initial population

The working principle of this method is to generate the initial population from a given sequence. In datasets there will be N number of DNA sequences and this method will find the l length of random sub-sequences. These sub-sequences will fill up the initial population.

For example,

Sequence 1 – AAAATTATTTTTT**TAGAC**TTCCTTC
Sequence 2 – **CAGCT**TTGCGCCTCCACTGTCACCC
Sequence 3 – TTGACAGACAGTGT**GGAGG**GATTA
Sequence 4 – ATACAC**GTACT**ACACATTGGACTCA
Sequence 5 – TGGG**AGGCA**AAGATGGTGGCAGGT
Sequence 6 – TTACAACAATCCATGCATACT**TTTAA**

$\Longrightarrow$

TAGAC
CAGCT
GGAGG
GTACT
AGGCA
TTTAA

Here, our method randomly chooses 5 length subsequences from a given DNA sequence. And add them to the initial population list. As we can see the subsequences of the initial population are already in datasets, so it will be close to an optimal solution.

3.3 Fitness Function

In this step, our main target is to get less number of mismatches. To do this, we have to run two fitness steps. One step is running to get the fitness in sequence and another one is running to get fitness in the whole dataset.
Firstly, to get fitness in sequence we are taking candidate motifs (subsequence) from population and one DNA sequence from the dataset as inputs. It will give numerical value as output. This numerical value is the minimum mismatch value. We get this value after doing comparison between every possible position of DNA sequence and candidate motif (subsequence). Here, from each position we will get one mismatch value. So, the position that returns the minimum mismatch value is the best for that particular candidate motif (subsequence). The output is represented in a matrix form.
As example, if our given subsequence is "TGCA" and DNA sequence is "ATTGCATGCCTT" then the output will be 0. Now we are giving some more examples like this: "ATTGCATGCCTT"

subsequences	ATTG	TTGC	TTGG	TTAA
output	0	0	1	2

Table 3.3: Fitness Function

Here, output is the minimum number of mismatches.

Now, to get fitness in the dataset we are taking one candidate motif (subsequence) and all sequences from the dataset as inputs. It will give numerical value as output that represents fitness in the whole dataset. Firstly, we initialize the fitness variable as zero as we don't get any output. Then, we run a for-loop from which we are getting all the sequences. Now, we are calling fitness functions where candidate motif and sequence from the dataset are given. It returns a numerical value for each sequence. Now, we are adding this value with the fitness variable, basically here the number of mismatch is stored. To get the average fitness value we are dividing the fitness value with the number of sequences. As an example, if our fitness value is

5 and the number of sequences is 20 then the average fitness value is 5/20 = 0.25. But, our target is to get the lowest number of mismatched values. If we subtract the fitness from L then we get the number of match values and to get the percentage we are dividing it from L. Now, the fitness value will return. This is the actual work of fitness value.
As an example, if our candidate motif is “GGTC” and the total dataset is “ATAGTAGCTAGCGGTCACGTATACG” then the output will return 1. As the minimum mismatch value is zero, so it returns the highest probability.

3.4 Assess Population

In this step, our main target is to get the main population between the selected candidate motifs (subsequences). Firstly, we take some subsequences from the population. Then, we calculate the fitness value of these subsequences by using the Fitness in Dataset step. Now we have fitness values of these subsequences and we store these values. Then, we sort these values in a descending order. Now, we know which subsequence has the better fitness value. We take 9 subsequences which have the better fitness values and these are our best population. We will do this step for several times to get the best population again and again. It will help us to get the best motif and also we will use it in our mutation step.
As an example, if our Population is “ATTCGATCGACTAGTACGAA”
Let our best 6 candidate motif is

{“TTCGCT”,“TTCGGT”, “TTCGTT”, “TTCATT”, “TTCCGT”, “TAGCAT”}

Subsequences	**Fitness Value (%)**
TTCGCT	83.3
TTCGGT	83.3
TTCGTT	83.3
TTCATT	50.0
TTCCGT	66.6
TAGCAT	33.3

Table 3.4: Assess Population

We get the fitness values of our 6 subsequences. Now we sort these values in a descending order and then we get these:

Subsequences	**Fitness Value (%)**
TTCGCT	83.3
TTCGGT	83.3
TTCGTT	83.3
TTCCGT	66.6
TTCATT	50.0
TAGCAG	33.3

Table 3.5: Sorted Assess Population

Finally, we take the best 4 subsequences as our best population.

3.5 Consensus Sequence

In this step, we basically count the number of selective nucleotides in a fixed position. We take the best population as inputs which we find from the assess population and then get numerical numbers as outputs. We have to do the whole step for all four nucleotides. Then we divide the output with the number of subsequences which we take as inputs. In the output, the number of columns has to be equal to the length of subsequences and the summation of each column has to be one.
As example, let our length of motif is 10 and we have 4 subsequences which is "CATGAGCTAC", "ACACGTCGAT", "TGCACAGATG", "GTCGTTGACA"
Then the output will look like,

A	0.25	0.25	0.25	0.25	0.25	0.25	0.0	0.5	0.5	0.25
C	0.25	0.25	0.5	0.25	0.25	0.0	0.5	0.0	0.25	0.25
T	0.25	0.25	0.0	0.5	0.25	0.25	0.5	0.25	0.0	0.25
G	0.25	0.25	0.25	0.0	0.25	0.5	0.0	0.25	0.25	0.25

Table 3.6: Consensus Sequence

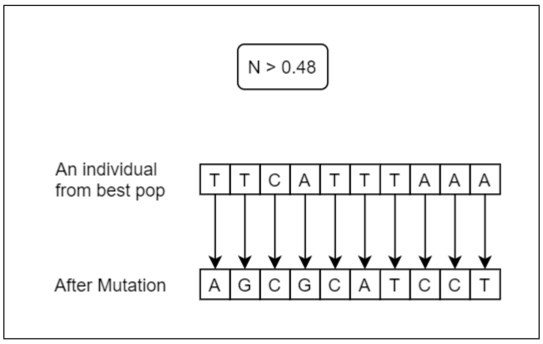
This will help us to get the best motif.

3.6 Mutation

Mutation plays the crucial part to meander around the candidate motif space. In this part, we will retrieve the best nucleotide from the highest frequent nucleotide in each position in the best population. After comparing the best nucleotide with the best population, we get the mismatch position and then we keep up the exchange off in two ways; exploration and exploitation. We have used binomial distribution for maintaining the exchange off between exploration and exploitation. We can control the number between 0 and 1 and it will not give any number that exceeds 1. We have used this technique so that we can easily explore a different part of solution space and can find the existence of the best solution.

3.6.1 Exploration

In this approach, every nucleotide of the individual from the best population gets randomly changed to a new candidate motif.



Here N = Random number from Binomial Distribution

Figure 3.2: Exploration

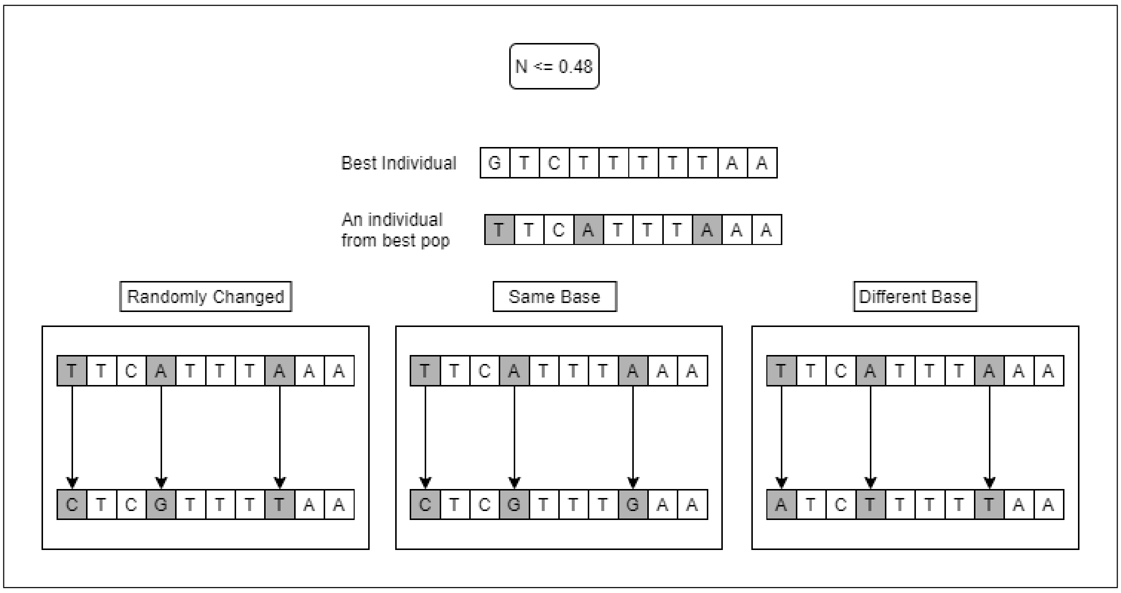
3.6.2 Exploitation

In this approach, the mismatch position of the nucleotide of the individual from the best population gets changed to a new candidate motif. We change the mismatch position in three ways:

1. Randomly change the nucleotide in mismatch position

2. The change of nucleotide in the same base like purine with purine bases and pyrimidine with pyrimidine bases.

3. The change of nucleotide in the different bases like purine with pyrimidine bases and vice versa.



Here N = Random Number from Binomial Distribution

Figure 3.3: Exploitation

Chapter 4

Datasets & Experimentation

4.1 Dataset

We have picked the dataset[29] from where there are 52 datasets. The dataset contains 6 from fly, 26 from human, 12 from mouse, and 8 from yeast. The number of sequences vary from 3 to 35 and the length of each sequence varies from 1000 to 2500. All the sequences in the dataset are in .fasta format.

Dataset	Total Sequence Length	Each Sequence Length
'dm01r'	4	1500
'hm01r'	18	2000
'mus01r'	3	500
'yst01r'	9	1000
'dm02r'	1	2000
'hm02r'	9	1000
'mus02r'	9	1000
'yst02r'	4	500
'dm03r'	3	2000
'hm03r'	10	1500
'mus03r'	5	500
'yst03r'	8	500
'dm04r'	4	2000
'hm04r'	13	2000
'mus04r'	7	1000
'yst04r'	7	1000
'dm05r'	3	2500
'hm05r'	3	1000
'mus05r'	4	500
'yst05r'	3	500

Table 4.1: Datasets

4.2 Experimentation

As an example, we have simulated the ‘hm05r’ dataset which contains 3 sequences with 3000 nucleotides to find the initial population, fitness score and consensus matrix using our proposed method.

Initial Population

The ‘hm05r’ dataset contains 3 sequences. From the sequences, we have got 36 subsequences from the random position of each sequence and we consider the length of each subsequence is 10.

‘TTCATTTAAA’	‘CTAATTTCCC’	‘AAATTAGGAG’	‘AATTTATCCG’
‘TCCTCCTCCT’	‘CTCGAGTTAG’	‘TCCGCGCTTT’	‘TGACTCCGCG’
‘GTTTTTCTGC’	‘ATGTTATTTA’	‘TTTAGGATCT’	‘TTTTCATTCT’
‘AGAATTTTAT’	‘CTAAATACTA’	‘TTTCATTTAA’	‘ACGTTTCATT’
‘GCCCCGCGGG’	‘AATCTAGGTA’	‘GCAGCCAGGG’	‘CTCCTAATTT’
‘AGGCGCCGTC’	‘GGCGACCGCG’	‘ACTCCTCCAA’	‘TGGGAGCTGG’
‘GAAGCGGACT’	‘TATTCTGAGT’	‘GAGTGGAGAA’	‘ACTTCGCCCC’
‘CTCGGGCTCT’	‘TGACGGCTGA’	‘CTCACTGCGG’	‘AGTGGAGAAG’
‘ACTGGGCGCG’	‘CGGGGCGCGG’	‘GGGGGCCCAG’	‘CCGCTAATTC’

Table 4.2: Initial Population

Fitness Score

We have got the fitness score of each subsequence from the initial population in percentage form.

Subsequences	**Fitness (%)**
‘TTCATTTAAA’	0.83
‘CTAATTTCCC’	0.83
‘AAATTAGGAG’	0.80
‘AATTTATCCG’	0.86
‘TCCTCCTCCT’	0.90
‘CTCGAGTTAG’	0.80
‘TCCGCGCTTT’	0.83
‘TGACTCCGCG’	0.76
‘GTTTTTCTGC’	0.83
‘ATGTTATTTA’	0.83
‘TTTAGGATCT’	0.80
‘TTTTCATTCT’	0.80
‘AGAATTTTAT’	0.80
‘CTAAATACTA’	0.76
‘TTTCATTTAA’	0.86
‘ACGTTTCATT’	0.86
‘GCCCCGCGGG’	0.86
‘AATCTAGGTA’	0.76

‘GCAGCCAGGG’	0.83
‘CTCCTAATTT’	0.83
‘AGGCGCCGTC’	0.83
‘GGCGACCGCG’	0.83
‘ACTCCTCCAA’	0.80
‘TGGGAGCTGG’	0.80
‘GAAGCGGACT’	0.83
‘TATTCTGAGT’	0.80
‘GAGTGGAGAA’	0.86
‘ACTTCGCCCC’	0.83
‘CTCGGGCTCT’	0.83
‘TGACGGCTGA’	0.83
‘CTCACTGCGG’	0.83
‘AGTGGAGAAG’	0.83
‘ACTGGGCGCG’	0.80
‘CGGGGCGCGG’	0.80
‘GGGGGCCCAG’	0.80
‘CCGCTAATTC’	0.80

Table 4.3: Fitness Score

Assess Population

From fitness score, we have sorted each subsequence in the population in a descending order according to fitness score. We have selected the top four subsequences and stored them in bestpopulation. From the table, ‘CCCTTTCCCC’ is a best subsequence for the highest fitness score.

Subsequences	**Fitness (%)**
‘TCCTCCTCCT’	0.90
‘TTTCATTTAA’	0.86
‘ACGTTTCATT’	0.86
‘GCCCCGCGGG’	0.86
‘AATTTATCCG’	0.86
‘GAGTGGAGAA’	0.86
‘TTCATTTAAA’	0.83
‘GTTTTTCTGC’	0.83
‘ATGTTATTTA’	0.83

Table 4.4: Assess Population

Consensus Matrix

Consensus matrix of our proposed method calculates the frequency of each nucleotide in each position. It returns a matrix that contains the probability of each nucleotide in each position. Our list of best population contains the candidate solutions - {’TCCTCCTCCT’,‘TTTCATTTAA’,‘ACGTTTCATT’,‘GCCCCGCGGG’, ‘AATTTATCCG’,‘GAGTGGAGAA’,‘TTCATTTAAA’,‘GTTTTTCTGC’, ‘ATGTTATTTA’} then this method will return -

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	0.33	0.22	0.00	0.11	0.11	0.22	0.11	0.22	0.33	0.44
C	0.00	0.33	0.33	0.22	0.22	0.11	0.33	0.22	0.22	0.11
G	0.33	0.00	0.33	0.00	0.11	0.22	0.00	0.22	0.22	0.22
T	0.33	0.44	0.33	0.67	0.56	0.44	0.56	0.33	0.22	0.22

Table 4.5: Consensus Matrix

We have run our algorithm on a data set of “hum05r” for finding motifs of different length. Figure 4.1 shows the found motif of length 10 from the human sequences in the data set.

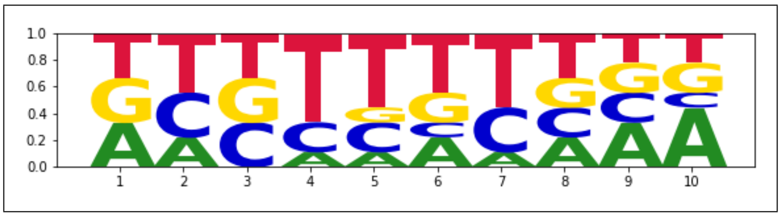


Figure 4.1: Motif Logo of Length 10 for Dataset ”hm05r”

Mutation

Mutation method of our proposed method mutated the nucleotides of mismatch positions. It returns a list of 27 mutated subsequences. We can see that some of the fitness of mutated subsequences are higher than best population subsequences. Table 4.6 contains the top 9 mutated subsequences.

Mutated Subsequences	**Fitness (%)**
‘TTTCATTTAA’	0.92
‘TTTTATTTAA’	0.92
‘TTCATTTAAA’	0.90
‘TTTTATTTAA’	0.89
‘TTCATTTAAA’	0.87
‘TTCTTTTTTT’	0.86
‘ATAAATTTAA’	0.85
‘GACAAAAAAA’	0.83
‘AATTTATTTA’	0.80

Table 4.6: Mutation

Chapter 5

Experimental Result & Analysis

5.1 Experimental Result

We apply our proposed method to different datasets from [29]. Here is the list of datasets with its sequence length and length of each sequence. In this chapter, we have showed the results for each dataset.

Dataset	**Length**	**Accuracy (%)**	**Motif**
‘hm01r’	8	91.00	AGAAAAAA
	13	83.33	AAAAATAAAAAAA
	15	80.00	AAAAAAATAAAAAAA
	23	72.00	AAAAAAAAAAAATAAAAAAAAAA
‘dm01r’	8	96.88	AAATAAAA
	13	90.38	AAAAAAAAATAAA
	15	85.00	AAAATAAAAATAAAA
	23	78.26	ATAAATAAAAAAAAACAAAATAC
‘yst01r’	8	94.44	ATTTTTTT
	13	82.91	AAATAAAAAAAAA
	15	81.48	AAAAAAAAAAAAAAA
	23	72.95	ATATAAAAAAAAAAAAAAAAACT
‘mus01r’	8	91.67	GGGCCACT
	13	84.62	GGGGCCACTGTCT
	15	82.22	GAGCTAAGAATAGCC
	23	73.91	TGTGGACCCTGCGTCGTGTAATA
‘hm02r’	8	93.06	CCCGCCCC
	13	81.20	ACCCACCCCCCTC
	15	81.48	CCCCCGCCTCCCCCT
	23	71.50	CCCCCCCCTCCCCCTCCCAAAGA
‘dm02r’	8	100.00	CATGTTCT
	13	84.62	CAGACGATCCATA
	15	86.67	CTCGCGGGCGGGCGA
	23	82.61	CAACCTGCGGCATGCTATTAAGA

'yst02r'	8	90.63	ATAATAAA
	13	90.38	ATAATAAAAAAAA
	15	85.00	CATAATAAAAAAAAA
	23	77.17	ACATAATAAAAAAAACCAGTTAA
'mus02r'	8	88.89	TAAAAAAA
	13	80.34	AAAAAAAAAAAAA
	15	80.00	AAGAGAAAAAAAAAA
	23	71.01	ATCAAAAAACATAAAAAAAAAAA
'hm03r'	8	91.25	AAAAAAAA
	13	83.85	AAAAAAAATAAAA
	15	81.33	AGAAAAAAAAATAAA
	23	71.30	AGAGCAAACAAAATAAAAAAATG
'dm03r'	8	100.00	TAAATGAA
	13	87.18	GTATTTTTCTTGA
	15	84.44	TTTAACTTTGAATTT
	23	76.81	GAAAATTTTTTTTATTATTAATA
'yst03r'	8	90.63	ATTTTTTT
	13	82.69	AAAGAAAAAAAAA
	15	80.83	AAAAGAAAAAAAAAA
	23	70.65	AAAAAAAAAAAAAAAAAATAAAA
'mus03r'	8	92.50	GGCTTCAG
	13	78.46	CTGCCCTACCCTC
	15	76.00	GGGGGCGGGAGGAGC
	23	69.57	CTTGGAAAACGCAGGAGCAGGCG
'hm04r'	8	93.75	AAAATAAA
	13	83.85	AAAAAAAATAAAA
	15	80.00	AAAAAACAAAATAAA
	23	73.48	AAAAAAAAACAGTGAAAAAAAAA
'dm04r'	8	96.88	AAACAAAA
	13	88.46	AAAAACAAAAAAA
	15	86.67	AAAAACAAAAAATAA
	23	82.61	AAAAAAAAAAAAAAAAAAAATAA
'yst04r'	8	98.21	ATTTTTTT
	13	86.81	AAAAAAAAAAAAA
	15	84.76	TTATTTTTTTTTTTT
	23	76.40	AATTTTTTATTTTTTTTTTATTT
'mus04r'	8	92.86	AATGAAAT
	13	80.22	GGAGAACAAGAGA
	15	75.24	TTTAAAAAAATGAAA
	23	68.94	AATGAAATGGGAGGAAAGTATGG
'hm05r'	8	95.83	GAAAAAAA
	13	82.05	GCCAGGAAGGAGG
	15	80.00	CAGAGGGCACAGTGG
	23	73.91	GATGTTATTTAGTTAAGAAGAAG

‘dm05r’	8	100.00	TTTATTAT
	13	92.31	AATAAATAAAAAA
	15	88.89	AAATAAATAAAAAAT
	23	76.81	TTAATAAAATAAAAAGCATAAAA
‘yst05r’	8	95.83	TTTCTTTT
	13	87.18	TTTGAAATTTTAT
	15	88.89	AATAGTTTCTTATTT
	23	79.71	TTGAAATTTTTTCAACCATGTAA
‘mus05r’	8	87.50	AAAAAAAA
	13	84.62	GGAAAAAAAAGGG
	15	80.00	GAGGTAGAAAAAAAG
	23	72.83	AGAAGAGGAAAAAAAAAGGGAGG

Table 5.1: Mutated Motifs with Accuracy

5.2 Analysis of Experimental Results

5.2.1 Time Analysis

In figure 5.1, it takes more time to find the motif for the longer length of sequence. To illustrate “Human” dataset has more length of sequences and it takes more time to find motif. If we compare “Human” dataset with other species it shows that “Fly”, “Mouse” and “Yeast” takes less time to find the motif.

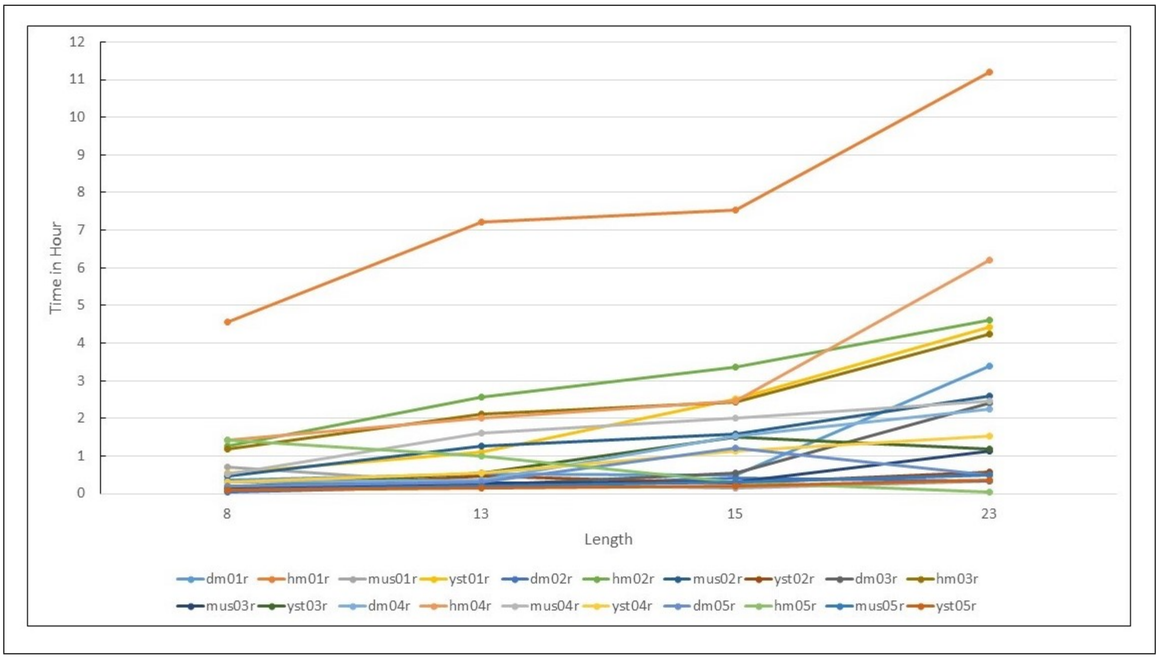
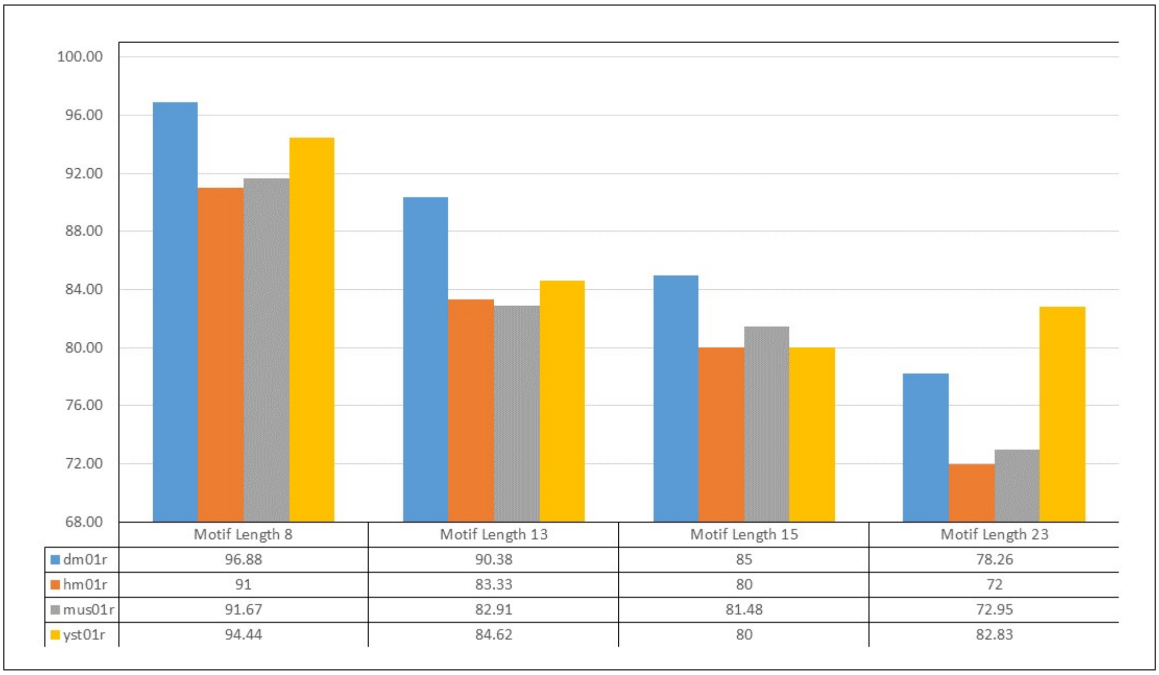


Figure 5.1: Motif Length vs Time graph

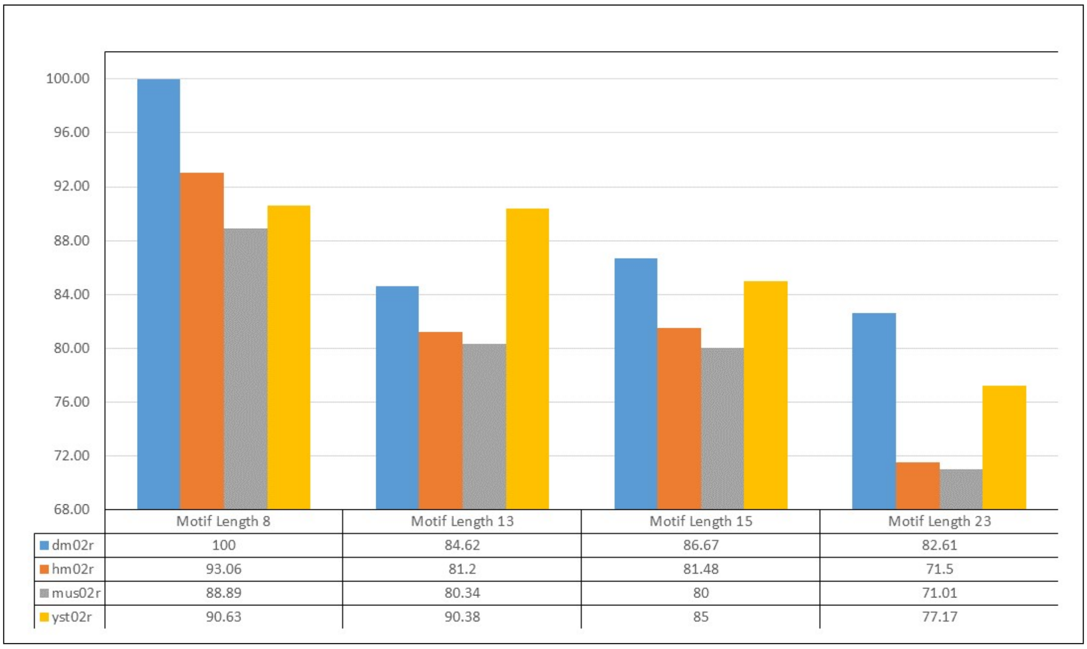
5.2.2 Accuracy of Mutated Motifs

From figure 5.2 to figure 5.6, it is clearly shown that we get higher accuracy for short sequences of motifs. The “dm02r”, “dm03r” and “dm05r” datasets get 100% accuracy for finding 8 lengths of motifs. On the other hand, for 13 and 15 lengths of motifs we get a very close range of accuracy. For 13 lengths of motifs, we get the best accuracy from the “dm05r” dataset. Also, we get good accuracy from “dm01r”, “ys01r” datasets. For 15 length of motifs, we get the best accuracy from the “dm05r” dataset. Also, we get good accuracy from “dm02r” and “dm04r” datasets. As it is shown in the bar graph, for long lengths of motifs we get lower accuracy compared with the short length of motifs. As we understand from our work that whenever the length of motif increases, the chance of finding subsequence of length which has the least number of mismatches increase and for this reason, we get lower accuracy for the long length of motifs. For 23 motif length, we get the best accuracy from “mus05r” and “yst05r” datasets which is less than the best accuracy of 8 length motifs. This whole scenario works for all the datasets that we used to get the accuracy.



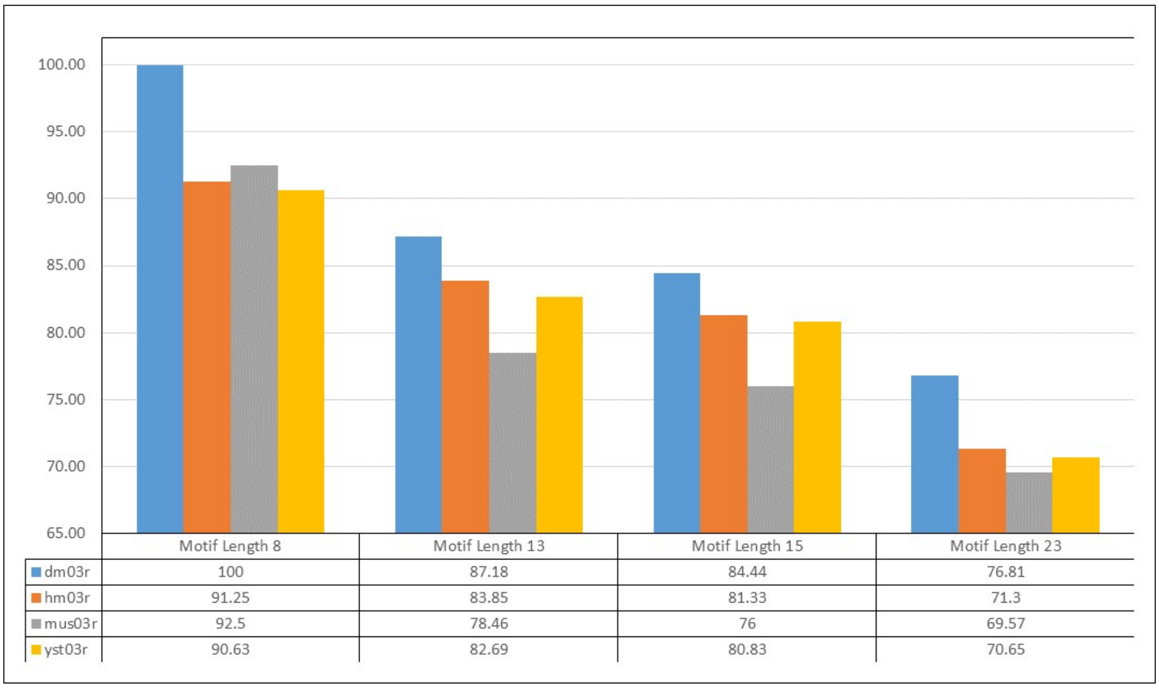
	Motif Length 8	Motif Length 13	Motif Length 15	Motif Length 23
🟦 dm01r	96.88	90.38	85	78.26
🟧 hm01r	91	83.33	80	72
⬜ mus01r	91.67	82.91	81.48	72.95
🟨 yst01r	94.44	84.62	80	82.83

Figure 5.2: Accuracy of the motifs with different lengths



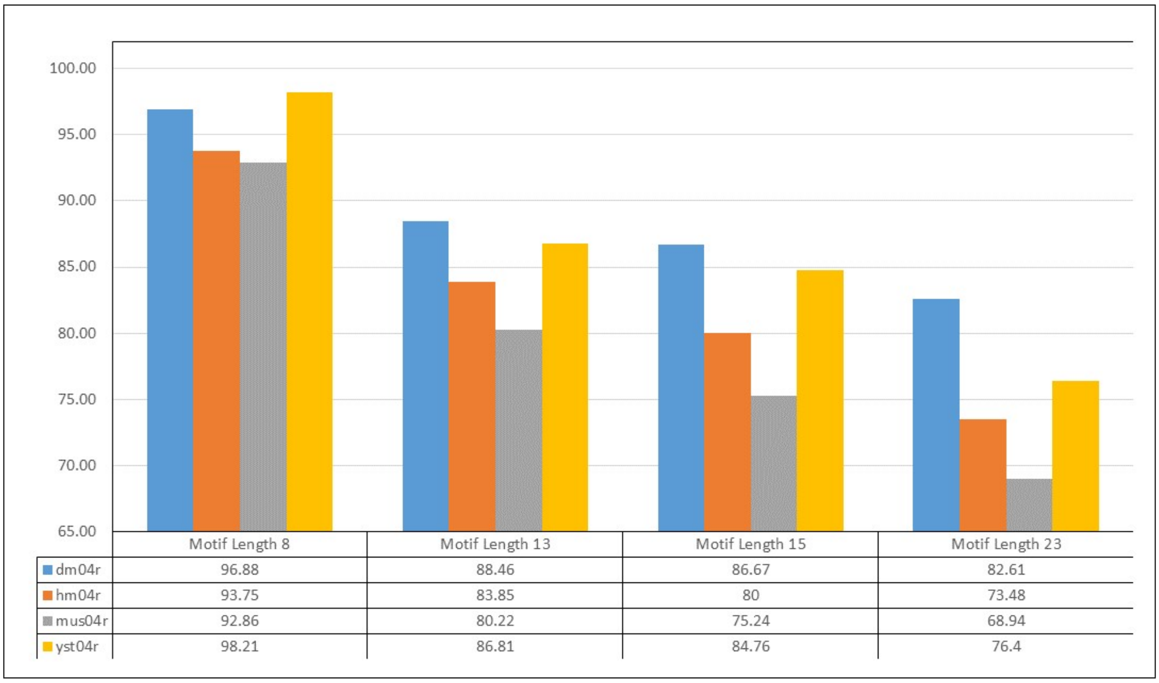
	Motif Length 8	Motif Length 13	Motif Length 15	Motif Length 23
🟦 dm02r	100	84.62	86.67	82.61
🟧 hm02r	93.06	81.2	81.48	71.5
◼ mus02r	88.89	80.34	80	71.01
🟨 yst02r	90.63	90.38	85	77.17

Figure 5.3: Accuracy of the motifs with different lengths



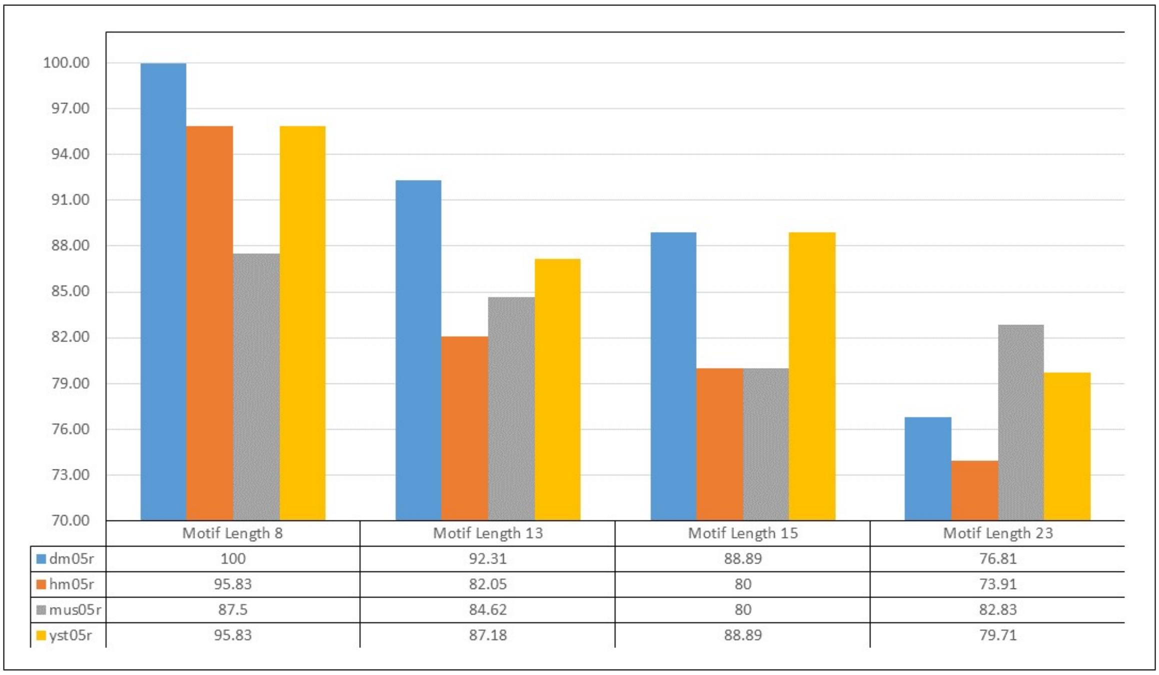
	Motif Length 8	Motif Length 13	Motif Length 15	Motif Length 23
🟦 dm03r	100	87.18	84.44	76.81
🟧 hm03r	91.25	83.85	81.33	71.3
◼ mus03r	92.5	78.46	76	69.57
🟨 yst03r	90.63	82.69	80.83	70.65

Figure 5.4: Accuracy of the motifs with different lengths



	Motif Length 8	Motif Length 13	Motif Length 15	Motif Length 23
dm04r	96.88	88.46	86.67	82.61
hm04r	93.75	83.85	80	73.48
mus04r	92.86	80.22	75.24	68.94
yst04r	98.21	86.81	84.76	76.4

Figure 5.5: Accuracy of the motifs with different lengths



	Motif Length 8	Motif Length 13	Motif Length 15	Motif Length 23
dm05r	100	92.31	88.89	76.81
hm05r	95.83	82.05	80	73.91
mus05r	87.5	84.62	80	82.83
yst05r	95.83	87.18	88.89	79.71

Figure 5.6: Accuracy of the motifs with different lengths

5.2.3 Accuracy of Motifs from Different Species

From figure 5.7 to figure 5.10 we have measured the accuracy of different lengths of the motifs such as “Fly”, “Human”, “Mouse” and “Yeast”. We showed 4 different motif lengths (8,13,15 & 23) of the dataset for each of the species.

In figure 5.7, we have shown the different motif lengths of “Fly” where the dataset of “dm02r”, “dm03r” and “dm05r” gets 100% accuracy for finding 8 lengths of motifs. For the other motifs such as “dm01r” and “dm04r” the accuracy is around 96%. As the motif length increases the accuracy for each dataset’s accuracy gets lesser. For motif length 13, each of the datasets gets accuracy around 85% to above 90%. For motif length 15, each of the datasets gets accuracy around 85% to above 88% and for 23 the data sets gets accuracy around 77% to above 83%.

Furthermore, in figure 5.8, we have shown the 4 motif lengths of “Human” where for each of the datasets(“hm01r”, “hm02r”, “hm03r”, “hm04r”, “hm05r”) the accuracy for the 8 length motif is above 90%, for the 13 length motif, the accuracy is around 83%, for the 15 length motif, the accuracy is around 80% to 82% and for the 23 length motif the accuracy is around 72% to 75%.

Moreover, in figure 5.9, we have shown the 4 motif lengths of “Mouse” where for each of the datasets(“mus01r”, “mus02r”, “mus03r”, “mus04r”, “mus05r”) the accuracy for the 8 length motif is around 88% to 93%, for the 13 length motif, the accuracy is around 82% to 85%, for the 15 length motif the accuracy is around 76% to 82% and for the 23 length motif, the accuracy is around 69% to 83%.

Lastly, in figure 5.7, we have shown the 4 motif lengths of “Yeast” where for each of the datasets(“yst01r”, “yst02r”, “yst03r”, “yst04r”, “yst05r”) the accuracy for the 8 length motif is above 90%, for the 13 length motif, the accuracy is around 83% to 91%, for the 15 length motif, the accuracy is around 80% to 89% and for the 23 length motif, the accuracy is around 71% to 83%.

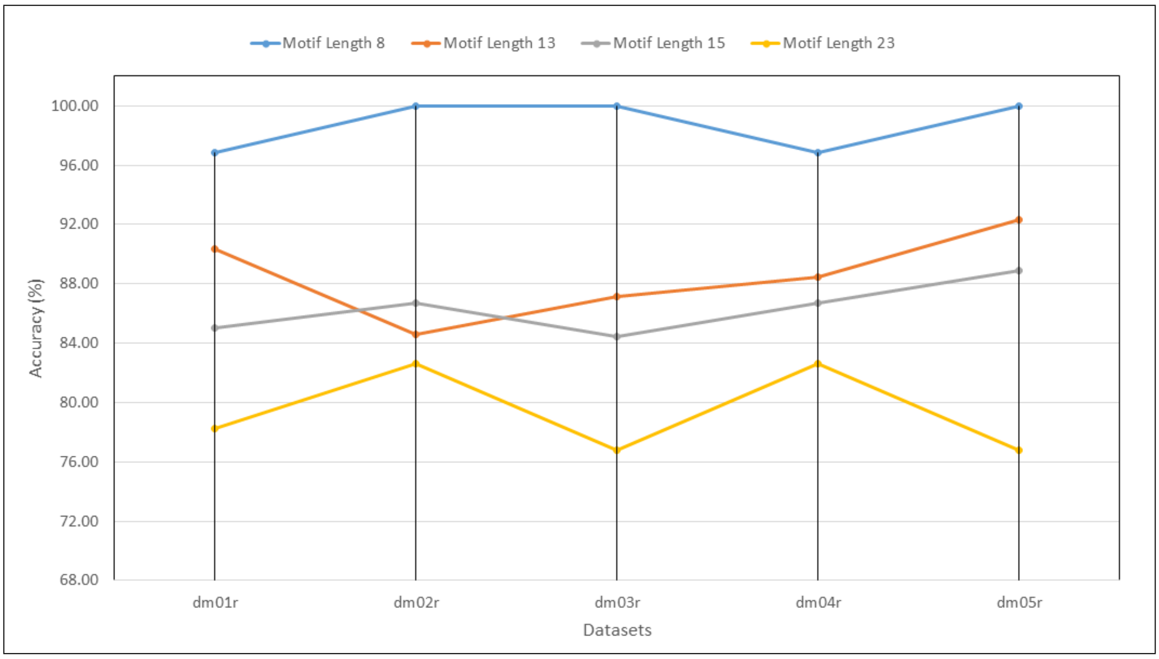


Figure 5.7: Accuracy of different lengths off the motifs of **‘Fly’** dataset

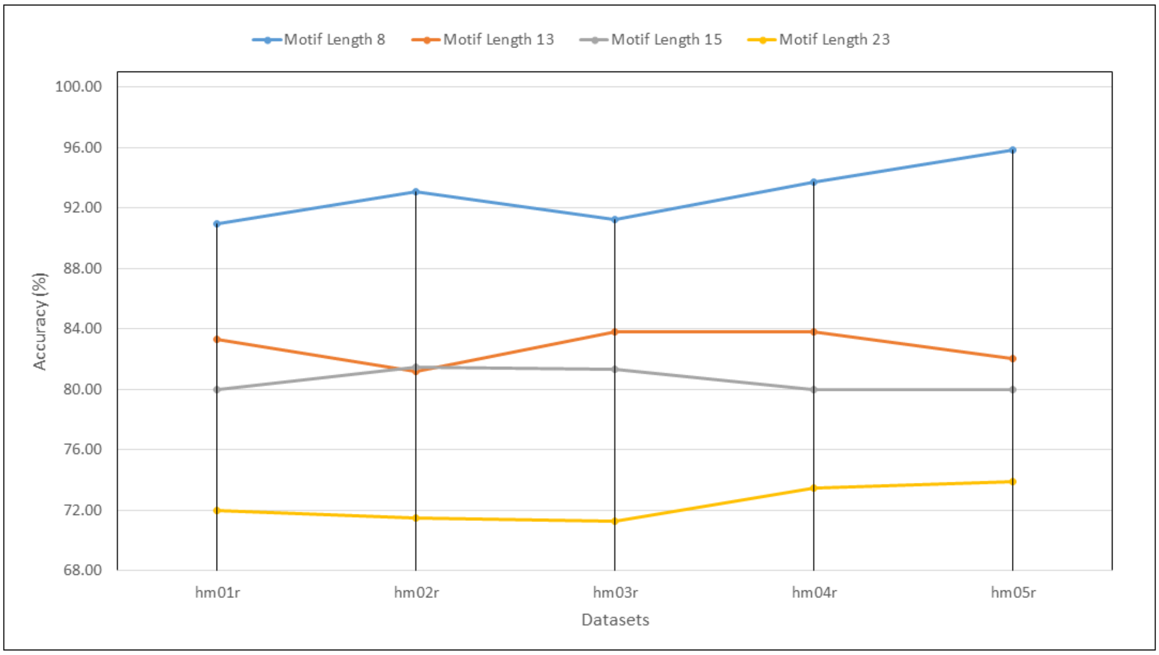


Figure 5.8: Accuracy of different lengths off the motifs of **‘Human’** dataset

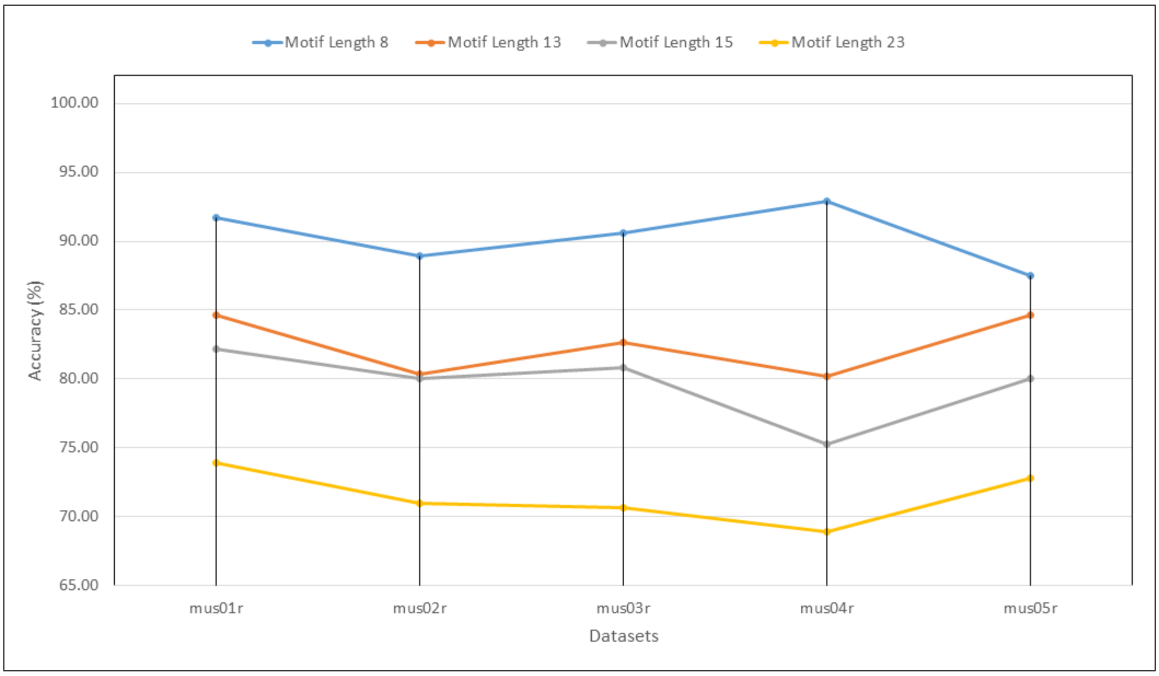


Figure 5.9: Accuracy of different lengths off the motifs of **‘Mouse’** dataset

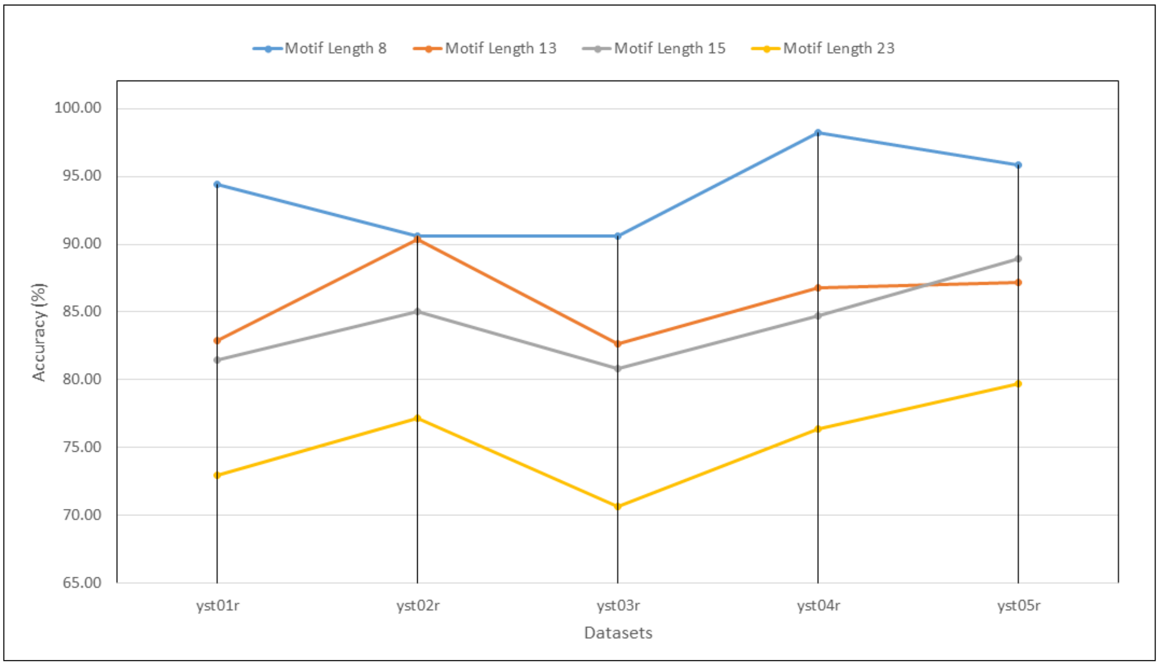


Figure 5.10: Accuracy of different lengths off the motifs of **'Yeast'** dataset

We have compared our proposed algorithm in various dataset with other established motif finding algorithms by calculating specificity. Some established methods are available at and we have compared our method with it. The result of AlignACE[30], ANN-spec[31], Consensus[32], GLAM[33], Improbizer[34], MEME[35], MEME3[35], MITRA[36], MotifSampler[37], Oligo/Dyad-Analysis [38], QuickScore[39], SeSiMCMC[40], Weeder[41] and YMF[9] are contained by it.
Before comparing the results we have to define some definitions.

- True Positives (TP): Number of positions in familiar sites and anticipated sites.
- True Negatives (TN): Number of positions which are neither in familiar sites nor in anticipated sites.
- False Positives (FP): Number of positions in anticipated sites that are not present in familiar sites.
- False Negatives (FN): Number of positions in familiar sites that are not present in familiar sites.

The following equation shows the specificity that calculates found motifs. It denotes how accurately the algorithm performed to get the actual motifs.

$$nSP = \frac{nTN}{nTN + nFP}$$

Data set	**Specificity**
‘dm01r’	0.988766
‘dm02r’	0.9958995
‘dm03r’	0.9959294
‘dm04r’	0.9936427
‘dm05r’	0.9949591
‘hm01r’	0.9984342
‘hm02r’	0.9963399
‘hm03r’	0.9958882
‘hm04r’	0.9984515
‘hm05r’	0.9943157
‘mus01r’	0.9780919
‘mus02r’	0.9922445
‘mus03r’	0.9898219
‘mus04r’	0.9964371
‘mus05r’	0.9952929
‘yst01r’	0.9918901
‘yst02r’	0.993129
‘yst03r’	0.9878017
‘yst04r’	0.9932123
‘yst05r’	0.9691877

Table 5.2: Specificity of Each Dataset

Table 5.3 and table 5.4 shows the presentation of our proposed method with other other established motif finding models with respect to the specificity. We have proposed an evolutionary method. The tables shows that our proposed method performs better than most of the existing methods. To illustrate, the datasets **‘mus01r’**, **‘hm02r’**, **‘mus02r’**, **‘hm03r’**, **‘yst03r’**, **‘yst04r’**, **‘mus05r’** etc. gives a better result than other methods. Besides this, our proposed model also works better for a longer number of sequences as well as a short number of sequences. Furthermore, our method also performs better for longer sequences and also for short sequences.

Data set	**Specificity**	**MFEA**	**AlignACE**	**ANN-Spec**	**Consensus**	**GLAM**	**Improbizer**	**MEME**	**MEME3**	**MITRA**
‘dm01r’	0.988766	-	-	-	-	-	-	-	-	-
‘dm02r’	0.9959	-	1	1	1	0.9794977	0.990774	0.9876986	0.9615582	1
‘dm03r’	0.995929	-	-	-	-	-	-	-	-	-
‘dm04r’	0.993643	-	-	-	-	-	-	-	-	-
‘dm05r’	0.994959	-	-	-	-	-	-	-	-	-
‘hm01r’	0.998434	-	-	-	-	-	-	-	-	-
‘hm02r’	**0.99634**	-	0.978268	0.965344	1	0.9855885	0.9814709	0.9616836	0.9845591	0.9910786
‘hm03r’	**0.995888**	0.991091	0.987527	0.992119	1	0.9869792	0.9882812	0.9853344	0.989926	0.9924616
‘hm04r’	0.998452	-	-	-	-	-	-	-	-	-
‘hm05r’	0.9943	-	1	0.929106	1	0.9839686	0.9804061	0.9946562	1	0.9893124
‘mus01r’	**0.978092**	0.990106	0.944876	0.95265	0.952381	1	0.9809187	0.9724382	0.9540636	0.9526502
‘mus02r’	**0.992245**	0.992587	1	0.967381	1	0.9897354	0.9759352	0.9849453	1	0.9887089
‘mus03r’	0.989822	-	-	-	-	-	-	-	-	-
‘mus04r’	0.996437	-	-	-	-	-	-	-	-	-
‘mus05r’	**0.995293**	-	1	0.971757	1	0.9801255	0.95659	0.9497908	0.9314854	0.9769874
‘yst01r’	0.99189	-	-	-	-	-	-	-	-	-
‘yst02r’	0.993129	-	-	-	-	-	-	-	-	-
‘yst03r’	**0.987802**	0.985329	-	-	-	-	-	-	-	-
‘yst04r’	**0.993212**	-	0.985517	0.980917	0.985347	0.9831317	0.9672857	0.9868802	0.9853467	0.9923326
‘yst05r’	0.969188		1	0.910364	0.97479	0.9810924	0.9453782	0.9831933	0.977591	0.9789916

- indicates that the dataset was not used for that algorithm

Table 5.3: Specificity Comparison with Established Methods of Different Dataset

Data set	**Speficity**	**MotifSampler**	**oligo/ dyad-analysis**	**QuickScore**	**SeSiMCMC**	**Weeder**	**YMF**
‘dm01r’	0.988766	-	-	-	-	-	-
‘dm02r’	0.9959	1	0.9769349	1	0.9077396	0.950795	0.9794977
‘dm03r’	0.995929	-	-	-	-	-	-
‘dm04r’	0.993643	-	-	-	-	-	-
‘dm05r’	0.994959	-	-	-	-	-	-
‘hm01r’	0.998434	-	-	-	-	-	-
‘hm02r’	**0.99634**	0.9833009	1	0.9752945	0.96603	1	0.9903923
‘hm03r’	**0.995888**	0.9908854	1	0.9909539	0.979852	1	0.9882127
‘hm04r’	0.998452	-	-	-	-	-	-
‘hm05r’	0.9943	0.9857499	0.9903812	0.9818311	0.9501247	0.992875	0.9718561
‘mus01r’	**0.978092**	0.9632509	1	0.9745583	0.8869258	0.9717314	0.9773852
‘mus02r’	**0.992245**	0.9828923	0.9944115	0.9852874	0.9671533	1	0.99099
‘mus03r’	0.989822	-	-	-	-	-	-
‘mus04r’	0.996437	-	-	-	-	-	-
‘mus05r’	**0.995293**	0.9722803	1	0.9780335	0.9241632	0.97908	0.9048117
‘yst01r’	0.99189	-	-	-	-	-	-
‘yst02r’	0.993129	-	-	-	-	-	-
‘yst03r’	**0.987802**	-	-	-	-	-	-
‘yst04r’	**0.993212**	0.9844948	0.991651	0.9918214	0.9753755	0.9923326	0.9959107
‘yst05r’	0.969188	0.9978992	0.981933	0.9663866	0.9488796	1	0.9831933

Table 5.4: Specificity Comparison with Established Methods of Different Dataset
- indicates that the dataset was not used for that algorithm

The figure 5.11 shows the specificity comparison between our proposed method with other existence methods on the basis of species like fly, human, mouse and yeast. In this graph, our algorithm gives best results on fly, human and mouse than other existing algorithms and for the yeast our method performs very close to the established algorithms.

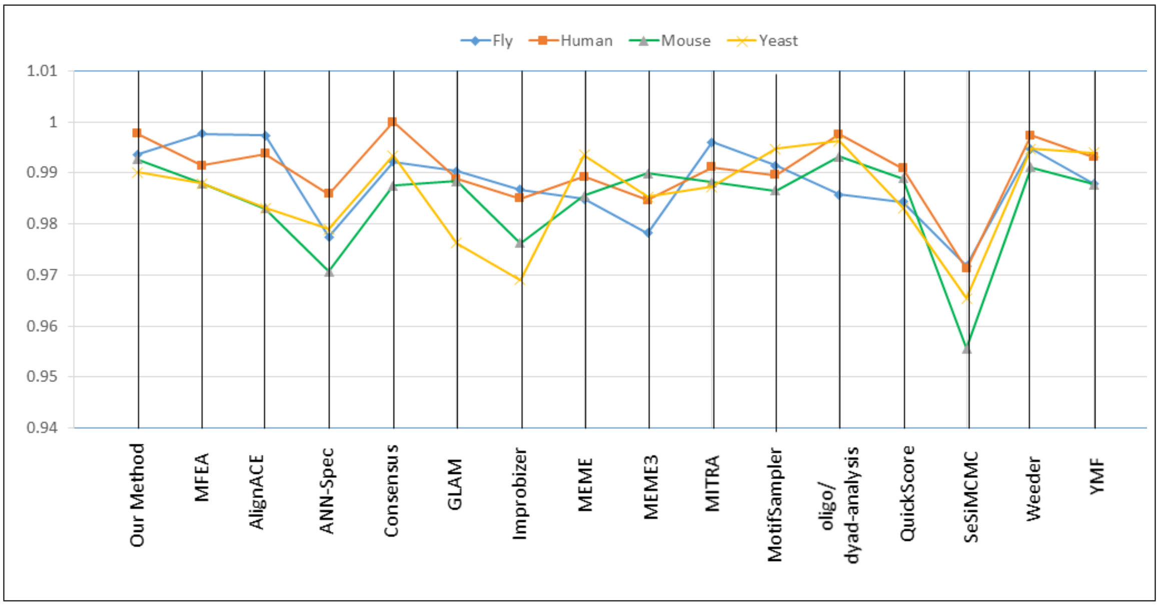


Figure 5.11: Scatter Graph from Specificity of Species of Different Algorithm.

Figure 5.12 shows the comparison of overall specificity of our proposed method with other existing methods. Our algorithm can find longer motifs from the dataset.

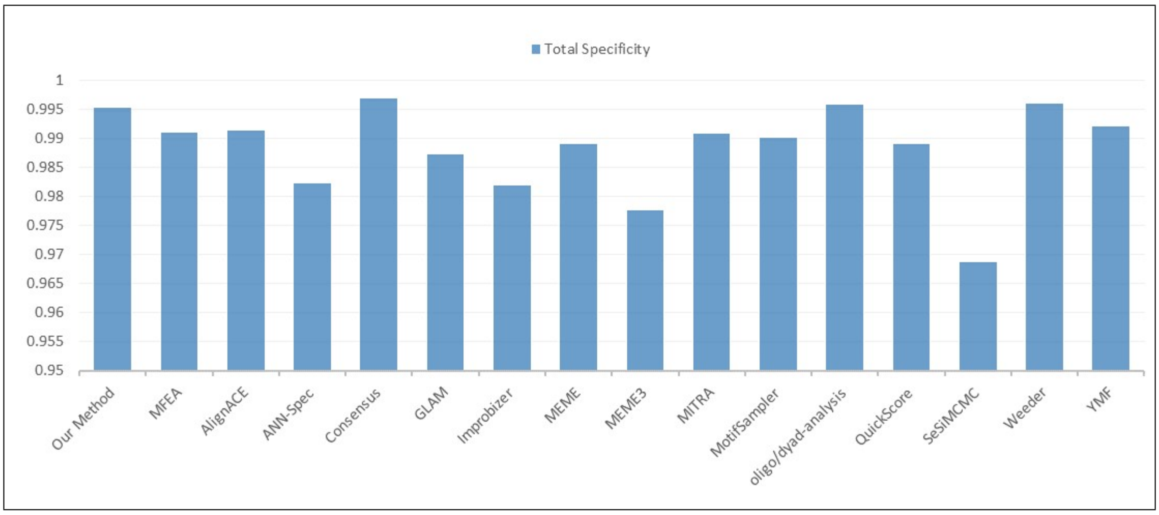


Figure 5.12: Comparison With Other Algorithms Based on Specificity

Chapter 6

Conclusion & Future Works

6.1 Conclusion

To conclude, we have suggested an evolutionary process to find motifs in DNA sequence. We have generated a set of initial candidate motifs and sorted out the best candidates from all of these. Doing mutation in the best candidates appears to produce the dataset's optimum motif. In numerous datasets, our method has performed well and fits the precision with accepted procedures that confirm our method's usefulness. In addition, this approach means that a very large size of motifs can be detected, which is not easy to locate using any other exhaustive process since it requires even days to measure. As in every step of evolution we are using heuristic and picking the best candidates, our approach faces no difficulty in discovering broad length motifs.

6.2 Future Works

Nevertheless, there are some places where modification can be possible in our proposed method. As the accuracy percentage decreases for higher length motifs therefore, we can introduce some new steps to get better accuracy for higher length motifs. We can improve our mutation method to get more accurate motifs. Furthermore, we can work on time duration. Also, we can reduce memory consumption. If we can do this modification, then this approach can be really efficient for finding best fitted motifs of different length with higher accuracy in the near future.

Bibliography

[1] J. A. Cooper and M. R. Bernfield, *Cell — Definition, Types, Functions.* 02 2019.

[2] T. E. of Encyclopaedia Britannica, *Chromosome — Structure Function — Britannica.* 2020.

[3] R. Rettner, "Dna: Definition, structure discovery," 12 2017.

[4] "The genetic code — boundless biology," 2013.

[5] P. Collas, "Chop it, chip it, check it: the current status of chromatin immunoprecipitation," *Frontiers in Bioscience*, vol. 13, p. 929, 2008.

[6] S. Pillai and S. P. Chellappan, "Chip on chip and chip-seq assays: Genome-wide analysis of transcription factor binding and histone modifications," *Methods in Molecular Biology*, vol. 523, pp. 447–472, 2015.

[7] E. R. Mardis, "Chip-seq: welcome to the new frontier," *Nature Methods*, vol. 4, pp. 613–614, 08 2007.

[8] J. van Helden, B. André, and J. Collado-Vides, "A web site for the computational analysis of yeast regulatory sequences," *Yeast*, vol. 16, pp. 177–187, 01 2000.

[9] S. Sinha and M. Tompa, "Ymf: a program for discovery of novel transcription factor binding sites by statistical overrepresentation," *Nucleic Acids Research*, vol. 31, pp. 3586–3588, 07 2003.

[10] K. Leppek, R. Das, and M. Barna, "Functional 5 utr mrna structures in eukaryotic translation regulation and how to find them," *Nature Reviews Molecular Cell Biology*, vol. 19, pp. 158–174, 11 2017.

[11] G. Z. Hertz, G. W. Hartzell, and G. D. Stormo, "Identification of consensus patterns in unaligned dna sequences known to be functionally related," *Bioinformatics*, vol. 6, pp. 81–92, 1990.

[12] C. E. Lawrence and A. A. Reilly, "An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins: Structure, Function, and Genetics*, vol. 7, pp. 41–51, 1990.

[13] M. Nicolae and S. Rajasekaran, "qpms9: An efficient algorithm for quorum planted motif search," *Scientific Reports*, vol. 5, p. 7813, 01 2015.

[14] W. Fang, X. Li, M. Zhang, and M. Hu, “Nature-inspired algorithms for real-world optimization problems,” *Journal of Applied Mathematics*, vol. 2015, pp. 1–2, 2015.

[15] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus, “An efficient heuristic approach for security against multiple adversaries,” *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS ’07*, 2007.

[16] B. Frankovič and I. Budinská, “Advantages and disadvantages of heuristic and multi agents approaches to the solution of scheduling problem,” *IFAC Proceedings Volumes*, vol. 33, pp. 367–372, 06 2000.

[17] F. A.Hashim, M. Mabrouk, and W. Al-Atabany, “Review of different sequence motif finding algorithms,” 04 2019.

[18] *AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION*, 2005.

[19] A. Rubio-Largo, D. L. Gonzalez-Alvarez, M. A. Vega-Rodriguez, J. A. Gomez-Pulido, and J. M. Sanchez-Perez, “Mo-abc/de - multiobjective artificial bee colony with differential evolution for unconstrained multiobjective optimization,” *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, 11 2012.

[20] X.-S. Yang and S. Deb, “Multiobjective cuckoo search for design optimization,” *Computers Operations Research*, vol. 40, pp. 1616–1624, 06 2013.

[21] S. Bouamama, A. Boukerram, and A. F. Al-Badarneh, “Motif finding using ant colony optimization,” *Lecture Notes in Computer Science*, pp. 464–471, 2010.

[22] Y. Zhang, L. Wang, and Q. Wu, “Modified adaptive cuckoo search (macs) algorithm and formal description for global optimisation,” *International Journal of Computer Applications in Technology*, vol. 44, p. 73, 2012.

[23] A. Makolo, “A comparative analysis of motif discovery algorithms,” *Computational Biology and Bioinformatics*, vol. 4, p. 1, 2016.

[24] M. Kaya, “Mogamod: Multi-objective genetic algorithm for motif discovery,” *Expert Systems with Applications*, vol. 36, pp. 1039–1047, 03 2009.

[25] F. Liu, J. Tsai, R. Chen, S. Chen, and S. Shih, “Fmga: finding motifs by genetic algorithm,” *Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering.*

[26] Proceedings of the International Congress on Biological and Medical Engineering, *Human promoter prediction based on sorted consensus sequence patterns by genetic algorithms*, 2002.

[27] U. Reddy, M. Arock, and A. Reddy, “Planted (l, d) - motif finding using particle swarm optimization,” *International Journal of Computer Applications*, vol. ecot, pp. 51–56, 12 2010.

[28] C. Lei and J. Ruan, “A particle swarm optimization-based algorithm for finding gapped motifs,” *BioData Mining*, vol. 3, 12 2010.

[29] “Motif discovery assessment [online]. available: http://bio.cs.washington.edu/assessment/download.html. [accessed: 16-dec-2020].”

[30] X. Ma, A. Kulkarni, Z. Zhang, Z. Xuan, R. Serfling, and M. Q. Zhang, “A highly efficient and effective motif discovery method for chip-seq/chip-chip data using positional information,” *Nucleic Acids Research*, vol. 40, pp. e50–e50, 01 2011.

[31] C. T. WORKMAN and G. D. STORMO, “Ann-spec: A method for discovering transcription factor binding sites with improved specificity,” *Biocomputing 2000*, 12 1999.

[32] J. Buhler and M. Tompa, “Finding motifs using random projections,” *Journal of Computational Biology*, vol. 9, pp. 225–242, 04 2002.

[33] M. C. Frith, “Finding functional sequence elements by multiple local alignment,” *Nucleic Acids Research*, vol. 32, pp. 189–200, 01 2004.

[34] P. PAVLIDIS, T. S. FUREY, M. LIBERTO, D. HAUSSLER, and W. N. GRUNDY, “Promoter region-based classification of genes,” *Biocomputing 2001*, 12 2000.

[35] T. L. Bailey, J. Johnson, C. E. Grant, and W. S. Noble, “The meme suite,” *Nucleic Acids Research*, vol. 43, pp. W39–W49, 05 2015.

[36] E. Eskin and P. A. Pevzner, “Finding composite regulatory patterns in dna sequences,” *Bioinformatics*, vol. 18, pp. S354–S363, 07 2002.

[37] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouze, and Y. Moreau, “A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling,” *Bioinformatics*, vol. 17, pp. 1113–1122, 12 2001.

[38] M. Tompa, N. Li, T. Bailey, G. Church, B. De Moor, E. Eskin, A. Favorov, M. Frith, Y. Fu, W. Kent, V. Makeev, A. Mironov, W. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, and Z. Zhu, “Assessing computational tools for the discovery of transcription factor binding sites,” *Nature biotechnology*, vol. 23, pp. 137–44, 02 2005.

[39] M. Régnier, “Mathematical tools for regulatory signals extraction,” *Bioinformatics of Genome Regulation and Structure*, pp. 61–69, 2004.

[40] A. V. Favorov, M. S. Gelfand, A. V. Gerasimova, D. A. Ravcheev, A. A. Mironov, and V. J. Makeev, “A gibbs sampler for identification of symmetrically structured, spaced dna motifs with improved estimation of the signal length,” *Bioinformatics*, vol. 21, pp. 2240–2245, 02 2005.

[41] G. Pavesi, P. Mereghetti, G. Mauri, and G. Pesole, “Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes,” *Nucleic Acids Research*, vol. 32, pp. W199–W203, 07 2004.

Consensus Matrices from Our Method of Different Dataset

dm01r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.56	0.78	0.33	0.33	0.33	0.33	0.56	0.44
	C	0.22	0.00	0.11	0.33	0.22	0.11	0.33	0.00
	G	0.11	0.11	0.11	0.00	0.11	0.56	0.11	0.33
	T	0.11	0.11	0.44	0.33	0.33	0.00	0.00	0.22
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.44	0.22	0.33	0.56	0.33	0.33	0.22	0.44
	C	0.00	0.33	0.33	0.11	0.22	0.33	0.11	0.33
	G	0.22	0.00	0.00	0.22	0.11	0.00	0.22	0.00
	T	0.33	0.44	0.33	0.11	0.33	0.33	0.44	0.22
		9	**10**	**11**	**12**	**13**			
	A	0.33	0.22	0.33	0.33	0.44			
	C	0.33	0.00	0.44	0.33	0.00			
	G	0.00	0.67	0.00	0.00	0.22			
	T	0.33	0.11	0.22	0.33	0.33			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.56	0.56	0.33	0.44	0.33	0.44	0.33	0.22
	C	0.22	0.33	0.11	0.11	0.11	0.22	0.11	0.11
	G	0.22	0.00	0.22	0.00	0.00	0.11	0.11	0.22
	T	0.00	0.11	0.33	0.44	0.56	0.22	0.44	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.22	0.56	0.33	0.33	0.33	0.33	0.22	
	C	0.22	0.00	0.00	0.33	0.22	0.22	0.22	
	G	0.11	0.11	0.22	0.11	0.11	0.22	0.44	
	T	0.44	0.33	0.44	0.22	0.33	0.22	0.11	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.11	0.33	0.44	0.56	0.44	0.56	0.33	0.78
	C	0.11	0.11	0.11	0.11	0.11	0.11	0.44	0.11
	G	0.11	0.00	0.22	0.00	0.11	0.11	0.11	0.11
	T	0.67	0.56	0.22	0.33	0.33	0.22	0.11	0.00
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.44	0.67	0.22	0.33	0.89	0.22	0.44
	C	0.00	0.11	0.11	0.22	0.00	0.00	0.00	0.11
	G	0.11	0.00	0.00	0.11	0.11	0.00	0.00	0.00
	T	0.67	0.44	0.22	0.44	0.56	0.11	0.78	0.44
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.67	0.33	0.78	0.56	0.33	0.44	0.78	
	C	0.00	0.22	0.11	0.11	0.11	0.11	0.00	
	G	0.11	0.00	0.00	0.22	0.11	0.00	0.00	
	T	0.22	0.44	0.11	0.11	0.44	0.44	0.22	

Table 6.1: Consensus Matrix of ‘dm01r” for various length

hm01r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.11	0.22	0.11	0.22	0.22	0.11	0.33	0.33
	C	0.33	0.22	0.22	0.22	0.44	0.56	0.22	0.22
	G	0.44	0.22	0.44	0.33	0.22	0.00	0.11	0.33
	T	0.11	0.33	0.22	0.22	0.11	0.33	0.33	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.33	0.33	0.44	0.56	0.11	0.22	0.11	0.11
	C	0.11	0.33	0.00	0.22	0.33	0.11	0.11	0.33
	G	0.33	0.22	0.22	0.11	0.33	0.33	0.44	0.22
	T	0.22	0.11	0.33	0.11	0.22	0.33	0.33	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.33	0.22	0.56	0.33	0.44			
	C	0.22	0.33	0.00	0.22	0.22			
	G	0.22	0.22	0.22	0.33	0.11			
	T	0.22	0.22	0.22	0.11	0.22			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.22	0.33	0.56	0.33	0.44	0.56	0.56	0.33
	C	0.33	0.11	0.22	0.33	0.33	0.44	0.22	0.33
	G	0.22	0.22	0.11	0.11	0.00	0.00	0.11	0.22
	T	0.22	0.33	0.11	0.22	0.22	0.00	0.11	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.22	0.33	0.33	0.22	0.22	0.22	
	C	0.33	0.22	0.56	0.33	0.11	0.33	0.56	
	G	0.22	0.44	0.00	0.00	0.33	0.11	0.22	
	T	0.33	0.11	0.11	0.33	0.33	0.33	0.00	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.00	0.00	0.00	0.22	0.33	0.33	0.56	0.11
	C	0.33	0.44	0.33	0.56	0.11	0.22	0.33	0.44
	G	0.44	0.22	0.22	0.22	0.44	0.33	0.11	0.22
	T	0.22	0.33	0.44	0.00	0.11	0.11	0.00	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.11	0.11	0.11	0.11	0.22	0.33	0.56	0.33
	C	0.44	0.22	0.44	0.33	0.22	0.22	0.22	0.56
	G	0.11	0.56	0.22	0.33	0.33	0.33	0.11	0.00
	T	0.33	0.11	0.22	0.22	0.22	0.11	0.11	0.11
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.00	0.11	0.22	0.33	0.33	0.11	0.22	
	C	0.22	0.44	0.56	0.33	0.11	0.11	0.33	
	G	0.33	0.44	0.22	0.11	0.44	0.33	0.33	
	T	0.44	0.00	0.00	0.22	0.11	0.44	0.11	

Table 6.2: Consensus Matrix of “hm01r” for various length

mus01r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.11	0.22	0.33	0.33	0.33	0.22	0.33	0.22
	C	0.44	0.22	0.33	0.22	0.22	0.33	0.22	0.22
	G	0.22	0.11	0.22	0.22	0.44	0.11	0.33	0.00
	T	0.22	0.44	0.11	0.22	0.00	0.33	0.11	0.56
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.22	0.00	0.11	0.11	0.22	0.33	0.33	0.44
	C	0.00	0.22	0.44	0.33	0.44	0.11	0.22	0.22
	G	0.11	0.11	0.22	0.56	0.22	0.22	0.33	0.00
	T	0.67	0.67	0.22	0.00	0.11	0.33	0.11	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.33	0.22	0.33	0.22	0.44			
	C	0.11	0.00	0.22	0.11	0.33			
	G	0.33	0.56	0.11	0.11	0.22			
	T	0.22	0.22	0.33	0.56	0.00			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.33	0.33	0.33	0.44	0.11	0.22	0.44	0.44
	C	0.22	0.33	0.11	0.33	0.33	0.44	0.44	0.00
	G	0.11	0.11	0.22	0.22	0.22	0.11	0.00	0.44
	T	0.33	0.22	0.33	0.00	0.33	0.22	0.11	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.22	0.33	0.44	0.33	0.22	0.22	0.44	
	C	0.22	0.11	0.22	0.22	0.44	0.11	0.22	
	G	0.33	0.11	0.33	0.22	0.22	0.11	0.22	
	T	0.22	0.44	0.00	0.22	0.11	0.56	0.11	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.22	0.22	0.00	0.22	0.22	0.11	0.11	0.33
	C	0.22	0.44	0.33	0.22	0.44	0.44	0.33	0.00
	G	0.44	0.11	0.44	0.33	0.00	0.22	0.22	0.56
	T	0.11	0.22	0.22	0.22	0.33	0.22	0.33	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.11	0.11	0.11	0.22	0.22	0.11	0.22	0.22
	C	0.11	0.22	0.33	0.22	0.33	0.22	0.22	0.22
	G	0.33	0.67	0.44	0.33	0.22	0.22	0.22	0.11
	T	0.44	0.00	0.11	0.22	0.22	0.44	0.33	0.44
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.11	0.22	0.33	0.22	0.44	0.22	0.00	
	C	0.11	0.33	0.22	0.33	0.00	0.33	0.67	
	G	0.22	0.22	0.33	0.33	0.33	0.33	0.22	
	T	0.56	0.22	0.11	0.11	0.22	0.11	0.11	

Table 6.3: Consensus Matrix of “mus01r” for various length

yst01r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.56	0.44	0.33	0.56	0.56	0.56	0.44	0.44
	C	0.11	0.33	0.22	0.11	0.11	0.11	0.11	0.00
	G	0.11	0.11	0.11	0.00	0.11	0.11	0.11	0.00
	T	0.22	0.11	0.33	0.33	0.22	0.22	0.33	0.56
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.33	0.22	0.22	0.44	0.11	0.22	0.56	0.22
	C	0.11	0.22	0.11	0.22	0.00	0.33	0.00	0.11
	G	0.11	0.11	0.00	0.00	0.22	0.11	0.11	0.22
	T	0.44	0.44	0.67	0.33	0.67	0.33	0.33	0.44
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.11	0.33	0.44	0.11			
	C	0.11	0.11	0.11	0.00	0.22			
	G	0.11	0.22	0.11	0.11	0.22			
	T	0.67	0.56	0.44	0.44	0.44			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.22	0.22	0.11	0.11	0.22	0.33	0.00	0.00
	C	0.22	0.00	0.33	0.22	0.22	0.00	0.44	0.22
	G	0.00	0.22	0.22	0.11	0.22	0.22	0.11	0.44
	T	0.56	0.56	0.33	0.56	0.33	0.44	0.44	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.33	0.33	0.00	0.22	0.11	0.22	0.44	
	C	0.22	0.00	0.11	0.00	0.00	0.11	0.00	
	G	0.00	0.11	0.33	0.11	0.11	0.22	0.00	
	T	0.44	0.56	0.56	0.67	0.78	0.44	0.56	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.33	0.22	0.00	0.33	0.33	0.33	0.44	0.56
	C	0.11	0.33	0.00	0.22	0.11	0.11	0.00	0.11
	G	0.22	0.11	0.11	0.00	0.00	0.11	0.22	0.11
	T	0.33	0.33	0.89	0.44	0.56	0.44	0.33	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.44	0.56	0.56	0.11	0.22	0.44	0.33
	C	0.11	0.00	0.11	0.22	0.22	0.22	0.00	0.00
	G	0.11	0.11	0.11	0.00	0.11	0.22	0.11	0.00
	T	0.56	0.44	0.22	0.22	0.56	0.33	0.44	0.67
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.22	0.22	0.56	0.33	0.33	0.56	0.56	
	C	0.33	0.44	0.33	0.00	0.33	0.11	0.11	
	G	0.11	0.11	0.11	0.11	0.00	0.00	0.11	
	T	0.33	0.22	0.00	0.56	0.33	0.33	0.22	

Table 6.4: Consensus Matrix of “yst01r” for different length

dm02r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.56	0.00	0.22	0.33	0.44	0.44	0.56	0.11
	C	0.11	0.44	0.22	0.44	0.22	0.22	0.22	0.33
	G	0.22	0.22	0.56	0.22	0.22	0.22	0.22	0.11
	T	0.11	0.33	0.00	0.00	0.11	0.11	0.00	0.44
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.00	0.22	0.56	0.11	0.11	0.11	0.22	0.22
	C	0.33	0.11	0.11	0.11	0.33	0.44	0.22	0.33
	G	0.22	0.33	0.22	0.56	0.00	0.11	0.22	0.11
	T	0.44	0.33	0.11	0.22	0.56	0.33	0.33	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.11	0.33	0.33	0.33			
	C	0.33	0.56	0.33	0.33	0.22			
	G	0.33	0.22	0.33	0.11	0.22			
	T	0.22	0.11	0.00	0.22	0.22			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.22	0.11	0.33	0.11	0.22	0.11	0.44	0.44
	C	0.33	0.11	0.22	0.22	0.33	0.11	0.11	0.22
	G	0.33	0.33	0.00	0.11	0.11	0.44	0.22	0.22
	T	0.11	0.44	0.44	0.56	0.33	0.33	0.22	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.00	0.22	0.33	0.22	0.22	0.33	
	C	0.44	0.11	0.22	0.22	0.22	0.56	0.22	
	G	0.22	0.11	0.44	0.44	0.33	0.11	0.22	
	T	0.22	0.78	0.11	0.00	0.22	0.11	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.11	0.22	0.11	0.33	0.33	0.33	0.33	0.56
	C	0.33	0.33	0.33	0.11	0.22	0.33	0.33	0.11
	G	0.11	0.00	0.33	0.33	0.22	0.11	0.11	0.11
	T	0.44	0.44	0.22	0.22	0.22	0.22	0.22	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.11	0.11	0.11	0.22	0.22	0.33	0.22	0.22
	C	0.44	0.11	0.44	0.33	0.44	0.22	0.67	0.11
	G	0.11	0.33	0.11	0.22	0.22	0.22	0.00	0.44
	T	0.33	0.44	0.33	0.22	0.11	0.22	0.11	0.22
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.11	0.33	0.33	0.22	0.33	0.33	0.11	
	C	0.22	0.22	0.33	0.22	0.56	0.11	0.33	
	G	0.22	0.00	0.11	0.33	0.11	0.22	0.22	
	T	0.44	0.44	0.22	0.22	0.00	0.33	0.33	

Table 6.5: Consensus Matrix of ‘dm02r” for various length

hm02r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.11	0.11	0.11	0.33	0.22	0.22	0.11	0.11
	C	0.33	0.22	0.56	0.44	0.67	0.11	0.22	0.56
	G	0.22	0.44	0.22	0.00	0.11	0.11	0.44	0.33
	T	0.33	0.22	0.11	0.22	0.00	0.56	0.22	0.00
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.22	0.33	0.11	0.00	0.22	0.00	0.00	0.00
	C	0.22	0.22	0.56	0.44	0.33	0.67	0.33	0.33
	G	0.33	0.33	0.11	0.22	0.11	0.22	0.44	0.56
	T	0.22	0.11	0.22	0.33	0.33	0.11	0.22	0.11
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.22	0.44	0.11	0.11			
	C	0.33	0.67	0.11	0.33	0.44			
	G	0.44	0.11	0.22	0.44	0.44			
	T	0.11	0.00	0.22	0.11	0.00			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.33	0.22	0.22	0.22	0.11	0.22	0.00	0.11
	C	0.22	0.33	0.22	0.33	0.56	0.22	0.11	0.44
	G	0.22	0.22	0.44	0.33	0.22	0.33	0.67	0.33
	T	0.22	0.22	0.11	0.11	0.11	0.22	0.22	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.11	0.22	0.00	0.00	0.33	0.11	
	C	0.44	0.56	0.33	0.22	0.33	0.33	0.22	
	G	0.33	0.11	0.22	0.44	0.33	0.11	0.44	
	T	0.11	0.22	0.22	0.33	0.33	0.22	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.11	0.00	0.00	0.22	0.11	0.00	0.11	0.11
	C	0.22	0.44	0.56	0.33	0.33	0.44	0.33	0.44
	G	0.22	0.44	0.44	0.33	0.22	0.44	0.33	0.33
	T	0.44	0.11	0.00	0.11	0.33	0.11	0.22	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.00	0.11	0.11	0.22	0.11	0.22	0.33	0.11
	C	0.44	0.44	0.33	0.67	0.33	0.44	0.33	0.33
	G	0.22	0.22	0.44	0.11	0.22	0.22	0.33	0.44
	T	0.33	0.22	0.11	0.00	0.33	0.11	0.00	0.11
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.11	0.22	0.00	0.11	0.00	0.00	
	C	0.00	0.22	0.33	0.33	0.22	0.33	0.22	
	G	0.22	0.44	0.33	0.22	0.11	0.56	0.44	
	T	0.33	0.22	0.11	0.44	0.56	0.11	0.33	

Table 6.6: Consensus Matrix of “hm02r” for different length

mus02r

Length		1	2	3	4	5	6	7	8
8	A	0.44	0.11	0.33	0.33	0.56	0.11	0.22	0.33
	C	0.00	0.22	0.22	0.11	0.22	0.56	0.33	0.33
	G	0.22	0.11	0.00	0.22	0.00	0.00	0.11	0.00
	T	0.33	0.56	0.44	0.33	0.22	0.33	0.33	0.33
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	A	0.67	0.33	0.44	0.44	0.22	0.56	0.33	0.67
	C	0.00	0.33	0.11	0.11	0.33	0.00	0.22	0.00
	G	0.22	0.11	0.11	0.11	0.22	0.11	0.11	0.11
	T	0.11	0.22	0.33	0.33	0.22	0.33	0.33	0.22
		9	**10**	**11**	**12**	**13**			
	A	0.33	0.33	0.11	0.22	0.22			
	C	0.22	0.33	0.33	0.11	0.11			
	G	0.44	0.00	0.22	0.11	0.11			
	T	0.00	0.33	0.33	0.56	0.56			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	A	0.11	0.22	0.33	0.44	0.67	0.44	0.33	0.33
	C	0.22	0.11	0.33	0.22	0.00	0.11	0.11	0.22
	G	0.11	0.11	0.00	0.11	0.11	0.11	0.22	0.11
	T	0.56	0.56	0.33	0.22	0.22	0.33	0.33	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.33	0.33	0.44	0.11	0.22	0.33	0.33	
	C	0.33	0.22	0.11	0.44	0.44	0.22	0.11	
	G	0.22	0.00	0.22	0.22	0.11	0.00	0.00	
	T	0.11	0.44	0.22	0.22	0.22	0.44	0.56	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	A	0.11	0.44	0.33	0.67	0.44	0.56	0.33	0.44
	C	0.22	0.22	0.22	0.33	0.11	0.00	0.22	0.11
	G	0.33	0.00	0.33	0.00	0.22	0.11	0.22	0.33
	T	0.33	0.33	0.11	0.00	0.22	0.33	0.22	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.33	0.67	0.11	0.56	0.22	0.44	0.33	0.56
	C	0.33	0.11	0.44	0.11	0.11	0.33	0.44	0.22
	G	0.33	0.11	0.11	0.11	0.22	0.11	0.11	0.00
	T	0.00	0.11	0.33	0.22	0.44	0.11	0.11	0.22
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.33	0.44	0.22	0.33	0.33	0.22	
	C	0.33	0.00	0.11	0.22	0.00	0.00	0.44	
	G	0.00	0.11	0.22	0.33	0.11	0.11	0.11	
	T	0.22	0.56	0.22	0.22	0.56	0.56	0.22	

Table 6.7: Consensus Matrix of “mus02r” for various length

yst02r

Length		1	2	3	4	5	6	7	8
8	A	0.78	0.78	0.44	0.44	0.44	0.56	0.44	0.67
	C	0.00	0.11	0.11	0.33	0.22	0.11	0.33	0.11
	G	0.00	0.00	0.11	0.11	0.00	0.22	0.00	0.11
	T	0.22	0.11	0.33	0.11	0.33	0.11	0.22	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	A	0.22	0.33	0.22	0.22	0.33	0.33	0.22	0.67
	C	0.22	0.11	0.22	0.33	0.11	0.11	0.11	0.11
	G	0.11	0.22	0.11	0.11	0.11	0.22	0.56	0.11
	T	0.44	0.33	0.44	0.33	0.44	0.33	0.11	0.11
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.44	0.33	0.33	0.11			
	C	0.11	0.11	0.00	0.11	0.11			
	G	0.22	0.22	0.33	0.00	0.00			
	T	0.56	0.22	0.33	0.56	0.78			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	A	0.22	0.33	0.33	0.33	0.33	0.33	0.22	0.33
	C	0.22	0.33	0.22	0.33	0.11	0.11	0.44	0.22
	G	0.11	0.11	0.11	0.22	0.00	0.11	0.11	0.11
	T	0.44	0.22	0.33	0.11	0.56	0.44	0.22	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.33	0.33	0.33	0.44	0.56	0.67	
	C	0.11	0.11	0.11	0.22	0.22	0.00	0.00	
	G	0.33	0.22	0.33	0.11	0.22	0.22	0.22	
	T	0.44	0.33	0.22	0.33	0.11	0.22	0.11	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	A	0.22	0.56	0.44	0.44	0.22	0.22	0.56	0.33
	C	0.33	0.22	0.00	0.00	0.33	0.00	0.11	0.11
	G	0.22	0.00	0.11	0.22	0.11	0.22	0.11	0.11
	T	0.22	0.22	0.44	0.33	0.33	0.56	0.22	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.56	0.11	0.22	0.56	0.33	0.11	0.33	0.33
	C	0.11	0.00	0.44	0.00	0.22	0.00	0.22	0.11
	G	0.11	0.44	0.00	0.11	0.22	0.00	0.33	0.22
	T	0.22	0.44	0.33	0.33	0.22	0.89	0.11	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.56	0.44	0.33	0.11	0.22	0.56	0.33	
	C	0.22	0.00	0.11	0.33	0.22	0.00	0.00	
	G	0.00	0.22	0.22	0.11	0.00	0.33	0.11	
	T	0.22	0.33	0.33	0.44	0.56	0.11	0.56	

Table 6.8: Consensus Matrix of “yst02r” for different length

dm03r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.56	0.33	0.33	0.33	0.22	0.22	0.22	0.22
	C	0.00	0.22	0.00	0.00	0.00	0.22	0.11	0.00
	G	0.22	0.11	0.22	0.00	0.22	0.11	0.11	0.33
	T	0.22	0.33	0.44	0.67	0.56	0.44	0.56	0.44
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.22	0.22	0.33	0.33	0.33	0.22	0.22	0.22
	C	0.00	0.22	0.22	0.11	0.11	0.11	0.22	0.11
	G	0.22	0.00	0.00	0.11	0.00	0.11	0.11	0.11
	T	0.56	0.56	0.44	0.44	0.56	0.56	0.44	0.56
		9	**10**	**11**	**12**	**13**			
	A	0.56	0.22	0.33	0.33	0.11			
	C	0.00	0.00	0.00	0.11	0.22			
	G	0.11	0.00	0.11	0.00	0.44			
	T	0.33	0.78	0.56	0.56	0.22			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.11	0.33	0.44	0.33	0.56	0.33	0.33	0.22
	C	0.22	0.11	0.33	0.11	0.00	0.11	0.56	0.11
	G	0.11	0.11	0.00	0.11	0.11	0.11	0.11	0.11
	T	0.56	0.44	0.22	0.44	0.33	0.44	0.00	0.56
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.22	0.44	0.33	0.22	0.11	0.11	0.22	
	C	0.00	0.11	0.22	0.22	0.11	0.11	0.11	
	G	0.33	0.00	0.11	0.00	0.11	0.00	0.00	
	T	0.44	0.44	0.33	0.56	0.67	0.78	0.67	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.33	0.33	0.11	0.56	0.56	0.11	0.44	0.44
	C	0.11	0.44	0.33	0.22	0.11	0.11	0.22	0.22
	G	0.00	0.00	0.11	0.00	0.00	0.22	0.11	0.11
	T	0.56	0.22	0.44	0.22	0.33	0.56	0.22	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.44	0.22	0.33	0.67	0.22	0.44	0.56
	C	0.11	0.11	0.11	0.00	0.00	0.11	0.11	0.00
	G	0.22	0.11	0.22	0.22	0.00	0.22	0.00	0.11
	T	0.44	0.33	0.44	0.44	0.33	0.44	0.44	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.56	0.22	0.33	0.44	0.33	0.44	0.44	
	C	0.22	0.33	0.11	0.11	0.00	0.00	0.11	
	G	0.00	0.00	0.22	0.22	0.11	0.11	0.11	
	T	0.22	0.44	0.33	0.22	0.56	0.44	0.33	

Table 6.9: Consensus Matrix of ‘dm03r” for various length

hm03r

Length		1	2	3	4	5	6	7	8
8	A	0.33	0.56	0.33	0.33	0.33	0.33	0.22	0.22
	C	0.22	0.22	0.22	0.44	0.22	0.22	0.22	0.00
	G	0.00	0.00	0.22	0.11	0.33	0.00	0.33	0.22
	T	0.44	0.22	0.22	0.11	0.11	0.44	0.22	0.56
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	A	0.44	0.44	0.22	0.33	0.56	0.33	0.44	0.33
	C	0.22	0.22	0.11	0.11	0.22	0.33	0.22	0.33
	G	0.22	0.22	0.33	0.11	0.11	0.11	0.00	0.11
	T	0.11	0.11	0.33	0.44	0.11	0.22	0.33	0.22
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.11	0.22	0.33	0.00			
	C	0.00	0.11	0.11	0.33	0.33			
	G	0.44	0.67	0.22	0.33	0.33			
	T	0.44	0.11	0.44	0.00	0.33			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	A	0.44	0.33	0.56	0.22	0.22	0.22	0.33	0.11
	C	0.22	0.22	0.22	0.11	0.33	0.22	0.11	0.22
	G	0.11	0.22	0.22	0.22	0.11	0.11	0.11	0.44
	T	0.22	0.22	0.00	0.44	0.33	0.44	0.44	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.44	0.22	0.67	0.56	0.22	0.11	0.22	
	C	0.22	0.11	0.22	0.33	0.22	0.22	0.22	
	G	0.11	0.33	0.00	0.00	0.33	0.22	0.33	
	T	0.22	0.33	0.11	0.11	0.22	0.44	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	A	0.56	0.44	0.56	0.56	0.56	0.33	0.56	0.56
	C	0.00	0.00	0.00	0.11	0.11	0.11	0.22	0.11
	G	0.00	0.44	0.11	0.11	0.11	0.22	0.22	0.00
	T	0.44	0.11	0.33	0.22	0.22	0.33	0.00	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.44	0.22	0.33	0.67	0.22	0.44	0.56
	C	0.11	0.11	0.11	0.00	0.00	0.11	0.11	0.00
	G	0.22	0.11	0.22	0.22	0.00	0.22	0.00	0.11
	T	0.44	0.33	0.44	0.44	0.33	0.44	0.44	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.11	0.22	0.22	0.33	0.33	0.11	
	C	0.11	0.11	0.11	0.22	0.11	0.33	0.11	
	G	0.11	0.11	0.11	0.11	0.44	0.11	0.33	
	T	0.33	0.67	0.56	0.44	0.11	0.22	0.44	

Table 6.10: Consensus Matrix of “hm03r” for different length

mus03r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.22	0.33	0.11	0.11	0.33	0.00	0.11	0.00
	C	0.44	0.22	0.44	0.22	0.56	0.11	0.44	0.44
	G	0.22	0.22	0.11	0.44	0.11	0.56	0.44	0.33
	T	0.11	0.22	0.33	0.22	0.00	0.33	0.00	0.22
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.44	0.00	0.22	0.22	0.11	0.22	0.22	0.11
	C	0.22	0.33	0.11	0.22	0.11	0.22	0.22	0.22
	G	0.33	0.44	0.33	0.22	0.44	0.56	0.44	0.44
	T	0.00	0.22	0.33	0.33	0.33	0.00	0.11	0.22
		9	**10**	**11**	**12**	**13**			
	A	0.22	0.11	0.11	0.22	0.22			
	C	0.11	0.44	0.11	0.00	0.11			
	G	0.44	0.33	0.56	0.78	0.44			
	T	0.22	0.11	0.22	0.00	0.22			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.11	0.33	0.33	0.11	0.22	0.11	0.11	0.22
	C	0.22	0.33	0.33	0.33	0.44	0.11	0.33	0.56
	G	0.67	0.11	0.22	0.22	0.22	0.56	0.33	0.11
	T	0.00	0.22	0.11	0.33	0.11	0.22	0.22	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.33	0.22	0.44	0.11	0.22	0.11	0.00	
	C	0.67	0.22	0.33	0.00	0.33	0.44	0.44	
	G	0.00	0.11	0.22	0.11	0.33	0.22	0.22	
	T	0.00	0.44	0.00	0.78	0.11	0.22	0.33	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.22	0.11	0.33	0.00	0.11	0.11	0.22	0.22
	C	0.33	0.22	0.33	0.33	0.33	0.22	0.00	0.22
	G	0.11	0.33	0.11	0.33	0.33	0.44	0.44	0.44
	T	0.33	0.33	0.22	0.33	0.22	0.22	0.33	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.11	0.22	0.11	0.00	0.11	0.11	0.33	0.22
	C	0.44	0.44	0.22	0.11	0.11	0.33	0.22	0.33
	G	0.33	0.22	0.56	0.67	0.44	0.44	0.33	0.22
	T	0.11	0.11	0.11	0.22	0.33	0.11	0.11	0.22
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.22	0.00	0.11	0.56	0.22	0.00	0.00	
	C	0.11	0.22	0.33	0.00	0.11	0.33	0.33	
	G	0.44	0.56	0.33	0.22	0.44	0.33	0.56	
	T	0.22	0.22	0.22	0.22	0.22	0.33	0.11	

Table 6.11: Consensus Matrix of “mus03r” for various length

yst03r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.33	0.11	0.22	0.33	0.33	0.33	0.56	0.56
	C	0.22	0.33	0.11	0.00	0.22	0.11	0.00	0.11
	G	0.00	0.11	0.00	0.22	0.11	0.22	0.00	0.11
	T	0.44	0.44	0.67	0.44	0.33	0.33	0.44	0.22
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.44	0.33	0.00	0.33	0.33	0.56	0.44	0.44
	C	0.22	0.11	0.00	0.22	0.33	0.22	0.00	0.11
	G	0.00	0.11	0.11	0.00	0.00	0.00	0.44	0.00
	T	0.33	0.44	0.89	0.44	0.33	0.22	0.11	0.44
		9	**10**	**11**	**12**	**13**			
	A	0.22	0.56	0.44	0.56	0.11			
	C	0.00	0.11	0.22	0.11	0.33			
	G	0.00	0.11	0.00	0.00	0.11			
	T	0.78	0.22	0.33	0.33	0.44			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.44	0.11	0.33	0.22	0.44	0.56	0.44	0.33
	C	0.11	0.44	0.00	0.33	0.22	0.22	0.11	0.00
	G	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.11
	T	0.44	0.44	0.67	0.44	0.11	0.22	0.44	0.56
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.22	0.11	0.44	0.33	0.33	0.22	0.56	
	C	0.22	0.22	0.00	0.11	0.11	0.22	0.11	
	G	0.00	0.11	0.22	0.11	0.33	0.11	0.11	
	T	0.56	0.56	0.33	0.44	0.22	0.44	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.44	0.22	0.11	0.33	0.33	0.00	0.00	0.11
	C	0.00	0.11	0.00	0.22	0.22	0.33	0.11	0.33
	G	0.11	0.00	0.22	0.11	0.11	0.33	0.11	0.11
	T	0.44	0.67	0.67	0.33	0.33	0.33	0.78	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.44	0.33	0.22	0.56	0.44	0.22	0.33	0.33
	C	0.00	0.22	0.33	0.11	0.00	0.11	0.22	0.11
	G	0.11	0.00	0.22	0.00	0.11	0.33	0.00	0.11
	T	0.44	0.44	0.22	0.33	0.44	0.33	0.44	0.44
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.33	0.56	0.22	0.44	0.33	0.56	0.56	
	C	0.11	0.22	0.33	0.22	0.00	0.11	0.11	
	G	0.11	0.00	0.00	0.11	0.11	0.00	0.22	
	T	0.44	0.22	0.44	0.22	0.56	0.33	0.11	

Table 6.12: Consensus Matrix of “yst03r” for different length

dm04r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.67	0.44	0.44	0.44	0.44	0.44	0.78	0.22
	C	0.22	0.11	0.11	0.22	0.33	0.33	0.11	0.33
	G	0.00	0.22	0.22	0.11	0.11	0.11	0.11	0.33
	T	0.11	0.22	0.22	0.22	0.11	0.11	0.00	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.56	0.33	0.44	0.33	0.56	0.67	0.56	0.44
	C	0.11	0.22	0.11	0.33	0.00	0.00	0.22	0.11
	G	0.00	0.00	0.22	0.11	0.11	0.11	0.11	0.11
	T	0.33	0.44	0.22	0.22	0.33	0.22	0.11	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.56	0.33	0.56	0.33	0.22			
	C	0.22	0.11	0.00	0.11	0.00			
	G	0.00	0.11	0.11	0.22	0.22			
	T	0.22	0.44	0.33	0.33	0.56			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.67	0.44	0.56	0.56	0.56	0.33	0.67	0.56
	C	0.00	0.00	0.00	0.11	0.11	0.11	0.00	0.00
	G	0.22	0.11	0.00	0.00	0.11	0.11	0.11	0.00
	T	0.11	0.44	0.44	0.33	0.22	0.44	0.22	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.67	0.44	0.44	0.22	0.44	0.33	0.22	
	C	0.11	0.11	0.00	0.00	0.11	0.33	0.44	
	G	0.00	0.11	0.22	0.44	0.11	0.22	0.11	
	T	0.22	0.33	0.33	0.33	0.33	0.11	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.33	0.67	0.22	0.44	0.33	0.33	0.56	0.56
	C	0.22	0.11	0.33	0.22	0.56	0.11	0.11	0.11
	G	0.33	0.22	0.11	0.11	0.00	0.44	0.33	0.22
	T	0.11	0.00	0.33	0.22	0.11	0.11	0.00	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.44	0.33	0.11	0.00	0.33	0.33	0.33	0.33
	C	0.00	0.00	0.22	0.44	0.22	0.00	0.00	0.00
	G	0.22	0.44	0.33	0.00	0.22	0.33	0.44	0.33
	T	0.33	0.22	0.33	0.56	0.22	0.33	0.22	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.33	0.44	0.22	0.22	0.33	0.44	0.22	
	C	0.22	0.11	0.22	0.33	0.22	0.11	0.00	
	G	0.33	0.22	0.44	0.22	0.22	0.22	0.44	
	T	0.11	0.22	0.11	0.22	0.22	0.22	0.33	

Table 6.13: Consensus Matrix of ‘dm04r” for various length

hm04r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.33	0.22	0.11	0.11	0.11	0.33	0.33	0.44
	C	0.11	0.33	0.00	0.56	0.33	0.00	0.11	0.22
	G	0.22	0.44	0.44	0.22	0.33	0.33	0.22	0.22
	T	0.33	0.00	0.44	0.11	0.22	0.33	0.33	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.33	0.44	0.33	0.33	0.11	0.22	0.22	0.44
	C	0.33	0.33	0.33	0.11	0.44	0.44	0.44	0.33
	G	0.11	0.11	0.00	0.22	0.22	0.11	0.11	0.11
	T	0.22	0.11	0.33	0.33	0.22	0.22	0.22	0.11
		9	**10**	**11**	**12**	**13**			
	A	0.11	0.22	0.11	0.33	0.33			
	C	0.33	0.11	0.33	0.33	0.00			
	G	0.44	0.22	0.44	0.33	0.33			
	T	0.11	0.44	0.11	0.00	0.33			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.22	0.22	0.11	0.11	0.56	0.22	0.44	0.22
	C	0.22	0.00	0.56	0.22	0.00	0.22	0.33	0.56
	G	0.22	0.22	0.22	0.00	0.44	0.33	0.11	0.11
	T	0.33	0.56	0.11	0.67	0.00	0.22	0.11	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.44	0.22	0.33	0.44	0.22	0.22	
	C	0.22	0.22	0.33	0.44	0.00	0.11	0.33	
	G	0.11	0.11	0.33	0.22	0.33	0.22	0.00	
	T	0.56	0.22	0.11	0.00	0.22	0.44	0.44	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.22	0.33	0.22	0.22	0.44	0.33	0.22	0.56
	C	0.22	0.22	0.33	0.22	0.22	0.00	0.44	0.22
	G	0.33	0.33	0.44	0.11	0.22	0.44	0.22	0.11
	T	0.22	0.11	0.00	0.44	0.11	0.22	0.11	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.33	0.11	0.44	0.33	0.33	0.33	0.22	0.22
	C	0.33	0.44	0.11	0.11	0.44	0.33	0.33	0.44
	G	0.00	0.22	0.22	0.22	0.11	0.11	0.33	0.00
	T	0.33	0.22	0.22	0.33	0.11	0.22	0.11	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.33	0.11	0.22	0.33	0.22	0.22	0.22	
	C	0.44	0.11	0.22	0.33	0.33	0.33	0.44	
	G	0.22	0.56	0.56	0.33	0.22	0.33	0.11	
	T	0.00	0.22	0.00	0.00	0.22	0.11	0.22	

Table 6.14: Consensus Matrix of “hm04r” for different length

mus04r

Length		1	2	3	4	5	6	7	8
8	**A**	0.33	0.33	0.22	0.33	0.56	0.22	0.33	0.22
	C	0.11	0.22	0.22	0.22	0.00	0.22	0.22	0.22
	G	0.33	0.22	0.00	0.33	0.00	0.22	0.22	0.11
	T	0.22	0.22	0.56	0.11	0.44	0.33	0.22	0.44
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.00	0.33	0.33	0.22	0.44	0.56	0.33	0.33
	C	0.33	0.00	0.22	0.22	0.00	0.11	0.00	0.22
	G	0.22	0.11	0.00	0.11	0.11	0.11	0.11	0.11
	T	0.44	0.56	0.44	0.44	0.44	0.22	0.56	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.56	0.44	0.33	0.56	0.44			
	C	0.22	0.11	0.11	0.00	0.00			
	G	0.00	0.22	0.33	0.11	0.22			
	T	0.22	0.22	0.22	0.33	0.33			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.33	0.22	0.67	0.56	0.67	0.11	0.56	0.56
	C	0.11	0.22	0.00	0.11	0.22	0.22	0.00	0.11
	G	0.33	0.56	0.00	0.11	0.11	0.11	0.33	0.22
	T	0.22	0.00	0.33	0.22	0.00	0.56	0.11	0.11
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.67	0.44	0.33	0.56	0.22	0.22	0.33	
	C	0.11	0.22	0.22	0.22	0.11	0.11	0.00	
	G	0.11	0.22	0.22	0.00	0.22	0.11	0.11	
	T	0.11	0.11	0.22	0.22	0.44	0.56	0.56	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.56	0.44	0.33	0.44	0.44	0.44	0.44	0.33
	C	0.00	0.11	0.22	0.00	0.11	0.11	0.00	0.00
	G	0.22	0.11	0.11	0.33	0.22	0.22	0.11	0.22
	T	0.22	0.33	0.33	0.22	0.22	0.22	0.44	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.56	0.11	0.22	0.44	0.22	0.44	0.11
	C	0.11	0.11	0.22	0.00	0.22	0.22	0.00	0.33
	G	0.33	0.11	0.22	0.33	0.22	0.22	0.44	0.22
	T	0.33	0.22	0.44	0.44	0.11	0.33	0.11	0.33
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.44	0.11	0.56	0.33	0.44	0.00	
	C	0.11	0.00	0.33	0.11	0.11	0.11	0.22	
	G	0.22	0.22	0.22	0.00	0.33	0.33	0.22	
	T	0.22	0.33	0.33	0.33	0.22	0.11	0.56	

Table 6.15: Consensus Matrix of “mus04r” for various length

yst04r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.44	0.44	0.44	0.33	0.11	0.33	0.11	0.33
	C	0.22	0.22	0.00	0.22	0.22	0.11	0.22	0.33
	G	0.11	0.22	0.22	0.11	0.11	0.11	0.22	0.00
	T	0.22	0.11	0.33	0.33	0.56	0.44	0.44	0.33
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.22	0.44	0.22	0.56	0.33	0.56	0.33	0.33
	C	0.22	0.00	0.33	0.11	0.22	0.00	0.11	0.33
	G	0.11	0.11	0.00	0.00	0.22	0.00	0.22	0.00
	T	0.44	0.44	0.44	0.33	0.22	0.44	0.33	0.33
		9	**10**	**11**	**12**	**13**			
	A	0.44	0.33	0.56	0.33	0.22			
	C	0.11	0.33	0.11	0.11	0.22			
	G	0.11	0.11	0.00	0.22	0.11			
	T	0.33	0.22	0.33	0.33	0.44			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.33	0.11	0.44	0.00	0.22	0.11	0.33	0.22
	C	0.11	0.56	0.11	0.00	0.11	0.22	0.56	0.11
	G	0.11	0.00	0.00	0.22	0.11	0.11	0.00	0.11
	T	0.44	0.33	0.44	0.78	0.56	0.56	0.11	0.56
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.44	0.22	0.33	0.22	0.56	0.22	0.44	
	C	0.11	0.11	0.22	0.22	0.00	0.11	0.00	
	G	0.00	0.22	0.00	0.00	0.11	0.00	0.11	
	T	0.44	0.44	0.44	0.56	0.33	0.67	0.44	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.33	0.00	0.22	0.22	0.33	0.33	0.00	0.33
	C	0.00	0.11	0.11	0.56	0.11	0.11	0.11	0.22
	G	0.33	0.22	0.00	0.11	0.00	0.11	0.11	0.00
	T	0.33	0.67	0.67	0.11	0.56	0.44	0.78	0.44
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.33	0.44	0.33	0.44	0.11	0.44	0.44
	C	0.22	0.22	0.33	0.11	0.00	0.11	0.00	0.22
	G	0.11	0.22	0.00	0.00	0.22	0.11	0.22	0.22
	T	0.44	0.22	0.22	0.56	0.33	0.67	0.33	0.11
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.56	0.33	0.33	0.33	0.56	0.22	0.11	
	C	0.00	0.33	0.33	0.22	0.11	0.44	0.22	
	G	0.11	0.00	0.00	0.22	0.00	0.00	0.11	
	T	0.33	0.33	0.33	0.22	0.33	0.33	0.56	

Table 6.16: Consensus Matrix of “yst04r” for different length

dm05r

Length		1	2	3	4	5	6	7	8
8	A	0.33	0.22	0.22	0.11	0.33	0.22	0.11	0.33
	C	0.22	0.11	0.11	0.22	0.33	0.33	0.11	0.11
	G	0.00	0.22	0.22	0.22	0.11	0.00	0.33	0.11
	T	0.44	0.44	0.44	0.44	0.22	0.44	0.44	0.44
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	A	0.33	0.11	0.56	0.56	0.56	0.11	0.44	0.11
	C	0.11	0.22	0.11	0.11	0.11	0.22	0.11	0.11
	G	0.22	0.22	0.00	0.22	0.22	0.33	0.11	0.22
	T	0.33	0.44	0.33	0.11	0.11	0.33	0.33	0.56
		9	**10**	**11**	**12**	**13**			
	A	0.44	0.33	0.33	0.22	0.44			
	C	0.11	0.22	0.00	0.11	0.00			
	G	0.11	0.22	0.11	0.33	0.22			
	T	0.33	0.22	0.56	0.33	0.33			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	A	0.22	0.22	0.11	0.22	0.11	0.11	0.33	0.00
	C	0.22	0.22	0.44	0.33	0.33	0.33	0.22	0.44
	G	0.11	0.33	0.22	0.33	0.33	0.11	0.11	0.22
	T	0.44	0.22	0.22	0.11	0.22	0.44	0.33	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.11	0.22	0.11	0.44	0.11	0.22	
	C	0.11	0.33	0.00	0.11	0.11	0.22	0.22	
	G	0.33	0.22	0.56	0.33	0.11	0.44	0.33	
	T	0.44	0.33	0.22	0.44	0.33	0.22	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	A	0.67	0.44	0.56	0.67	0.78	0.56	0.44	0.56
	C	0.00	0.44	0.22	0.22	0.11	0.33	0.22	0.00
	G	0.22	0.11	0.00	0.11	0.11	0.00	0.00	0.11
	T	0.11	0.00	0.22	0.00	0.00	0.11	0.33	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.22	0.11	0.22	0.33	0.67	0.44	0.33	0.56
	C	0.11	0.11	0.11	0.00	0.00	0.22	0.11	0.22
	G	0.33	0.11	0.22	0.11	0.11	0.22	0.22	0.22
	T	0.33	0.67	0.44	0.56	0.22	0.11	0.33	0.00
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.44	0.44	0.44	0.33	0.11	0.67	
	C	0.11	0.11	0.11	0.22	0.11	0.67	0.00	
	G	0.00	0.22	0.00	0.11	0.11	0.00	0.00	
	T	0.44	0.22	0.44	0.22	0.44	0.22	0.33	

Table 6.17: Consensus Matrix of ‘dm05r” for various length

hm05r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.11	0.33	0.33	0.33	0.11	0.11	0.11	0.22
	C	0.11	0.00	0.44	0.33	0.33	0.33	0.22	0.33
	G	0.56	0.33	0.11	0.00	0.56	0.33	0.44	0.33
	T	0.22	0.33	0.11	0.33	0.00	0.22	0.22	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.33	0.22	0.11	0.22	0.33	0.33	0.22	0.22
	C	0.22	0.22	0.33	0.33	0.11	0.11	0.00	0.22
	G	0.22	0.44	0.22	0.33	0.44	0.44	0.44	0.33
	T	0.22	0.11	0.33	0.11	0.11	0.11	0.33	0.22
		9	**10**	**11**	**12**	**13**			
	A	0.22	0.33	0.44	0.44	0.22			
	C	0.22	0.22	0.11	0.11	0.33			
	G	0.44	0.44	0.33	0.44	0.44			
	T	0.11	0.00	0.11	0.00	0.00			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.11	0.11	0.22	0.33	0.56	0.22	0.33	0.22
	C	0.11	0.22	0.22	0.33	0.22	0.33	0.11	0.11
	G	0.44	0.56	0.56	0.33	0.22	0.44	0.44	0.44
	T	0.33	0.11	0.00	0.00	0.00	0.00	0.11	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.11	0.00	0.33	0.11	0.11	0.11	0.22	
	C	0.11	0.44	0.11	0.33	0.33	0.33	0.44	
	G	0.67	0.33	0.22	0.33	0.44	0.33	0.11	
	T	0.11	0.22	0.33	0.22	0.11	0.22	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.33	0.33	0.22	0.33	0.33	0.22	0.11	0.22
	C	0.11	0.22	0.11	0.11	0.11	0.33	0.00	0.22
	G	0.44	0.33	0.44	0.44	0.11	0.33	0.56	0.56
	T	0.11	0.11	0.22	0.11	0.44	0.11	0.33	0.00
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.00	0.00	0.22	0.44	0.33	0.44	0.22	0.22
	C	0.22	0.22	0.33	0.11	0.22	0.22	0.00	0.44
	G	0.56	0.56	0.44	0.33	0.33	0.22	0.56	0.33
	T	0.22	0.22	0.00	0.11	0.11	0.11	0.22	0.00
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.33	0.22	0.22	0.11	0.22	0.11	0.22	
	C	0.00	0.22	0.22	0.56	0.22	0.22	0.22	
	G	0.44	0.33	0.22	0.11	0.33	0.67	0.44	
	T	0.22	0.22	0.33	0.22	0.22	0.00	0.11	

Table 6.18: Consensus Matrix of “hm05r” for different length

mus05r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.67	0.22	0.44	0.22	0.33	0.22	0.44	0.33
	C	0.11	0.11	0.22	0.33	0.33	0.33	0.33	0.33
	G	0.22	0.44	0.11	0.22	0.33	0.22	0.11	0.22
	T	0.00	0.22	0.22	0.22	0.00	0.22	0.11	0.11
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.22	0.22	0.44	0.67	0.33	0.44	0.44	0.33
	C	0.11	0.33	0.22	0.11	0.22	0.11	0.11	0.22
	G	0.22	0.11	0.22	0.00	0.22	0.22	0.22	0.33
	T	0.44	0.33	0.11	0.22	0.22	0.22	0.22	0.11
		9	**10**	**11**	**12**	**13**			
	A	1.00	0.33	0.22	0.33	0.78			
	C	0.00	0.00	0.11	0.22	0.11			
	G	0.00	0.56	0.22	0.33	0.00			
	T	0.00	0.11	0.44	0.11	0.11			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.33	0.11	0.22	0.33	0.22	0.33	0.33	0.33
	C	0.11	0.56	0.22	0.44	0.56	0.22	0.22	0.33
	G	0.11	0.22	0.11	0.11	0.11	0.11	0.44	0.11
	T	0.44	0.11	0.44	0.11	0.11	0.33	0.00	0.22
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.44	0.33	0.44	0.44	0.22	0.44	0.22	
	C	0.00	0.00	0.11	0.11	0.33	0.11	0.33	
	G	0.22	0.56	0.33	0.33	0.00	0.33	0.00	
	T	0.33	0.11	0.11	0.11	0.44	0.11	0.44	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.44	0.22	0.44	0.33	0.11	0.22	0.22	0.22
	C	0.00	0.22	0.11	0.11	0.00	0.33	0.22	0.33
	G	0.22	0.22	0.00	0.33	0.56	0.11	0.44	0.11
	T	0.33	0.33	0.44	0.22	0.33	0.33	0.11	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.56	0.22	0.11	0.56	0.44	0.44	0.56	0.67
	C	0.11	0.44	0.44	0.11	0.11	0.22	0.22	0.11
	G	0.22	0.33	0.11	0.11	0.22	0.22	0.22	0.00
	T	0.11	0.00	0.33	0.22	0.22	0.11	0.00	0.22
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.56	0.33	0.33	0.11	0.33	0.22	0.67	
	C	0.22	0.22	0.00	0.22	0.22	0.22	0.11	
	G	0.11	0.33	0.44	0.56	0.33	0.33	0.11	
	T	0.11	0.11	0.22	0.11	0.11	0.22	0.11	

Table 6.19: Consensus Matrix of “mus05r” for various length

yst05r

Length		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
8	**A**	0.33	0.22	0.33	0.22	0.33	0.22	0.44	0.33
	C	0.22	0.33	0.11	0.00	0.11	0.33	0.22	0.22
	G	0.00	0.22	0.22	0.22	0.22	0.00	0.00	0.00
	T	0.44	0.22	0.33	0.56	0.33	0.44	0.33	0.44
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
13	**A**	0.33	0.67	0.33	0.56	0.44	0.33	0.44	0.33
	C	0.22	0.11	0.22	0.11	0.11	0.00	0.11	0.11
	G	0.22	0.00	0.22	0.22	0.00	0.33	0.11	0.00
	T	0.22	0.22	0.22	0.11	0.44	0.33	0.33	0.56
		9	**10**	**11**	**12**	**13**			
	A	0.44	0.33	0.22	0.22	0.33			
	C	0.11	0.00	0.11	0.22	0.00			
	G	0.11	0.11	0.11	0.00	0.22			
	T	0.33	0.56	0.56	0.56	0.44			
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
15	**A**	0.11	0.22	0.00	0.33	0.33	0.56	0.33	0.44
	C	0.22	0.00	0.44	0.00	0.33	0.11	0.22	0.11
	G	0.00	0.22	0.11	0.22	0.22	0.22	0.11	0.11
	T	0.67	0.56	0.44	0.44	0.11	0.11	0.33	0.33
		9	**10**	**11**	**12**	**13**	**14**	**15**	
	A	0.44	0.67	0.33	0.44	0.33	0.22	0.67	
	C	0.33	0.11	0.11	0.00	0.11	0.11	0.11	
	G	0.00	0.00	0.11	0.33	0.11	0.00	0.00	
	T	0.22	0.22	0.44	0.22	0.44	0.67	0.22	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
23	**A**	0.56	0.33	0.22	0.22	0.22	0.33	0.33	0.33
	C	0.11	0.11	0.11	0.22	0.11	0.22	0.00	0.11
	G	0.11	0.00	0.11	0.11	0.22	0.00	0.11	0.00
	T	0.22	0.56	0.56	0.44	0.44	0.44	0.56	0.56
		9	**10**	**11**	**12**	**13**	**14**	**15**	**16**
	A	0.33	0.00	0.11	0.22	0.11	0.00	0.11	0.22
	C	0.11	0.11	0.22	0.00	0.11	0.33	0.22	0.22
	G	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.11
	T	0.56	0.56	0.67	0.78	0.78	0.67	0.67	0.44
		17	**18**	**19**	**20**	**21**	**22**	**23**	
	A	0.44	0.22	0.11	0.00	0.22	0.11	0.11	
	C	0.22	0.11	0.22	0.00	0.22	0.22	0.33	
	G	0.00	0.11	0.00	0.33	0.33	0.22	0.00	
	T	0.33	0.56	0.67	0.67	0.22	0.44	0.56	

Table 6.20: Consensus Matrix of “yst05r” for different length

Motif Logo from Our Method of Different Dataset

dm01r

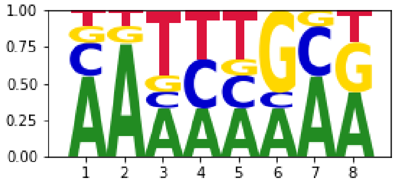
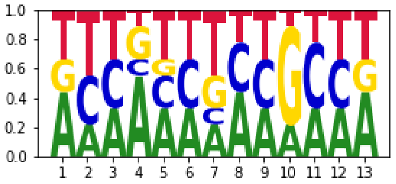
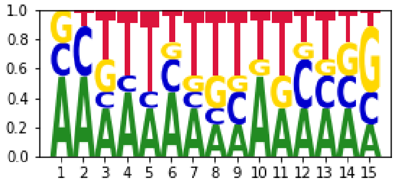
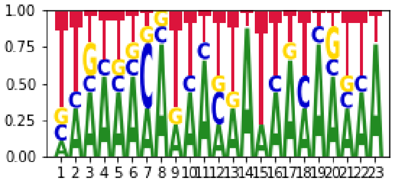
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.21: Motif Logo of “dm01r” for different length

hm01r

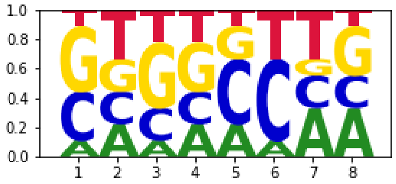
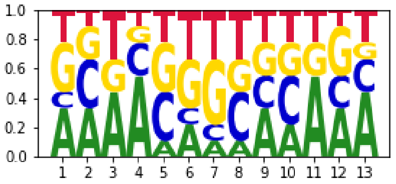
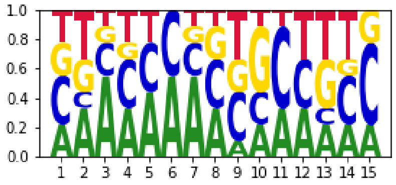
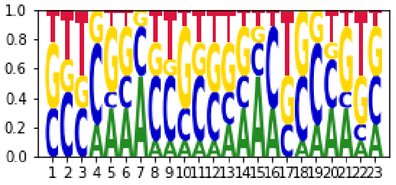
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.22: Motif Logo of “hm01r” for different length

mus01r

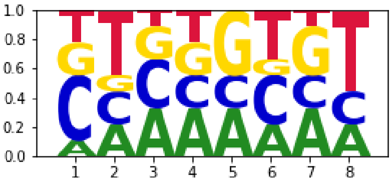
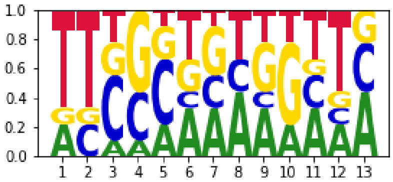
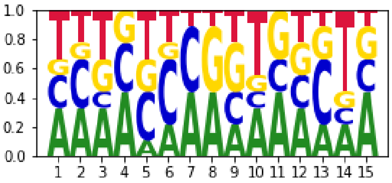
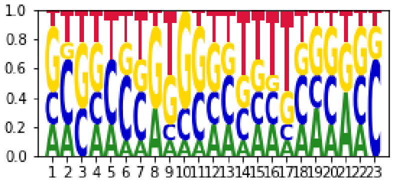
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.23: Motif Logo of “mus01r” for different length

yst01r

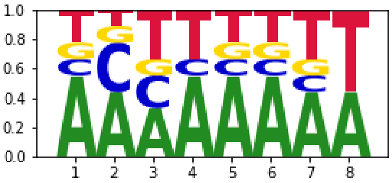
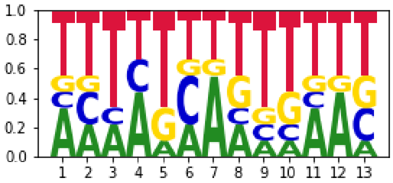
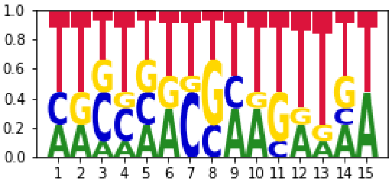
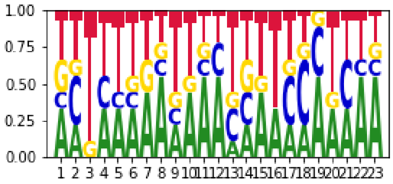
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.24: Motif Logo of “yst01r” for different length

dm02r

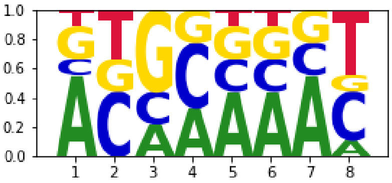
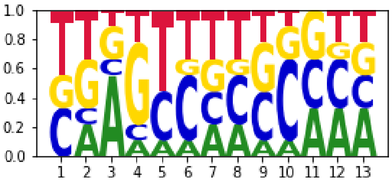
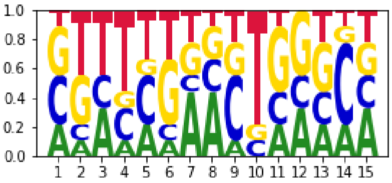
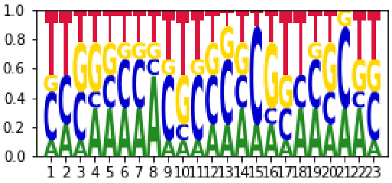
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.25: Motif Logo of “dm02r” for different length

hm02r

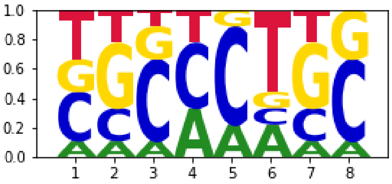
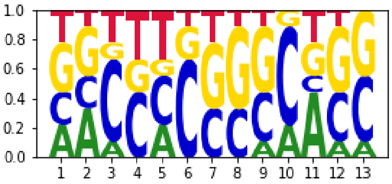
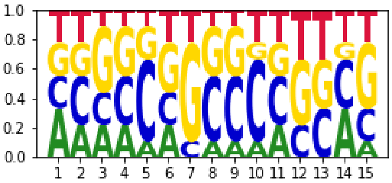
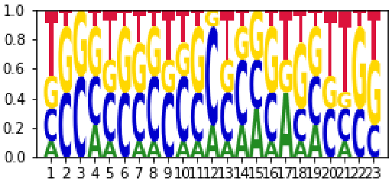
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.26: Motif Logo of “hm02r” for different length

mus02r

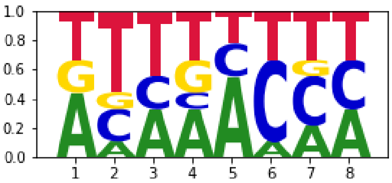
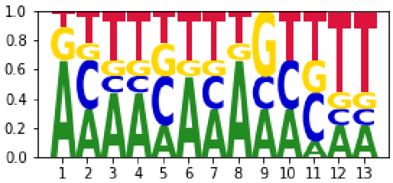
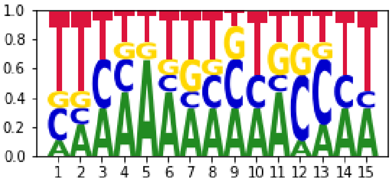
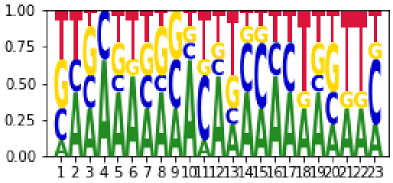
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.27: Motif Logo of “mus02r” for different length

yst02r

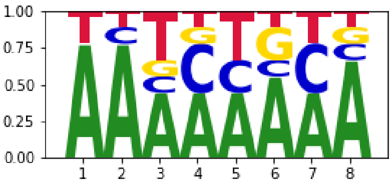
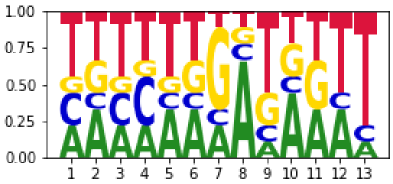
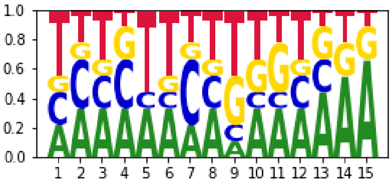
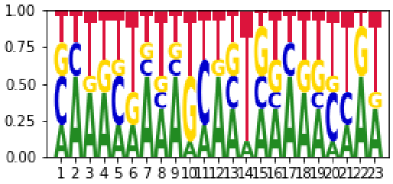
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.28: Motif Logo of “yst02r” for different length

dm03r

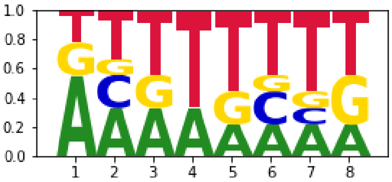
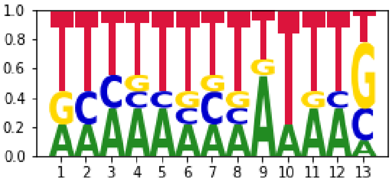
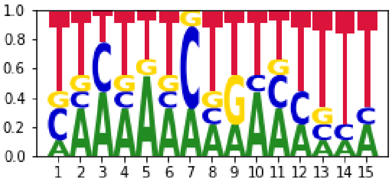
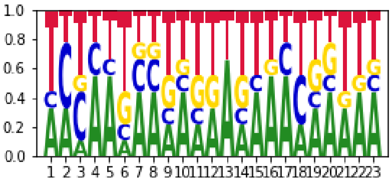
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.29: Motif Logo of “dm03r” for different length

hm03r

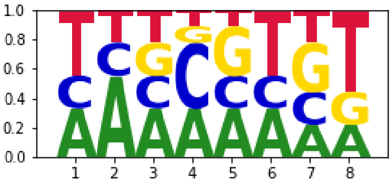
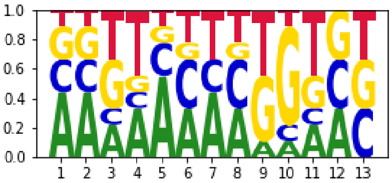
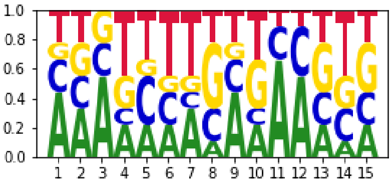
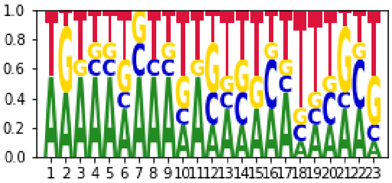
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.30: Motif Logo of “hm03r” for different length

mus03r

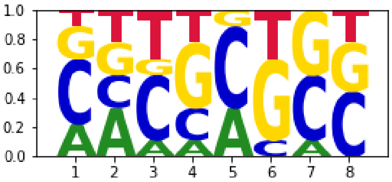
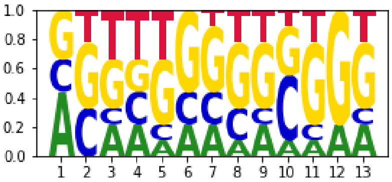
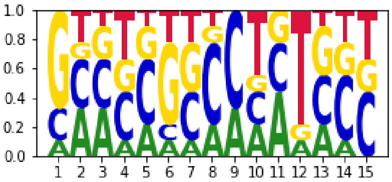
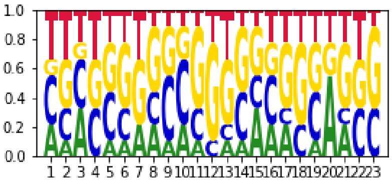
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.31: Motif Logo of “mus03r” for different length

yst03r

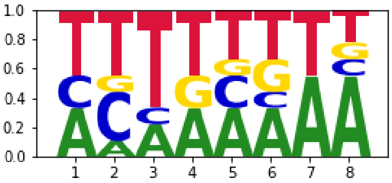
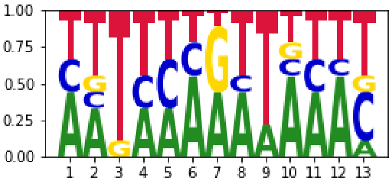
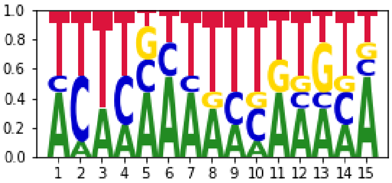
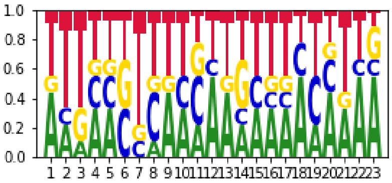
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.32: Motif Logo of “yst03r” for different length

dm04r

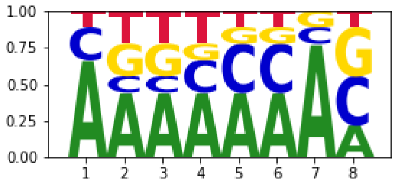
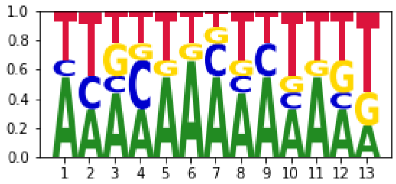
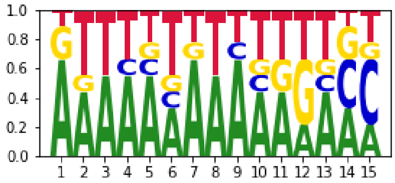
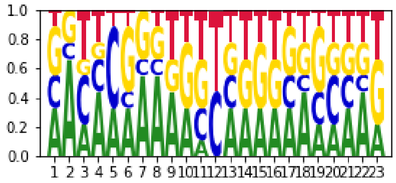
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.33: Motif Logo of “dm04r” for different length

hm04r

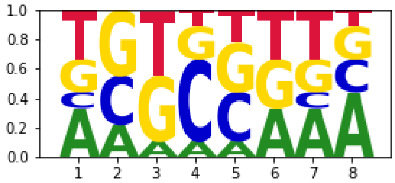
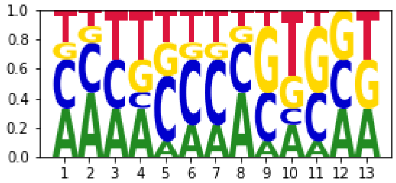
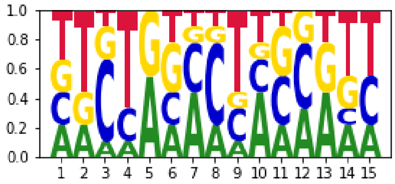
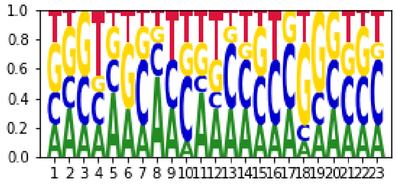
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.34: Motif Logo of “hm04r” for different length

mus04r

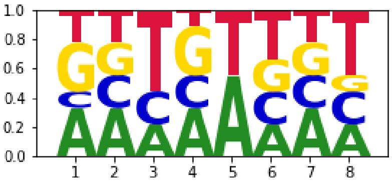
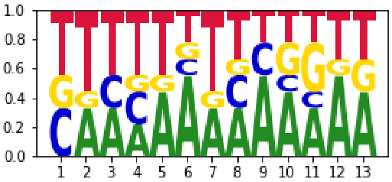
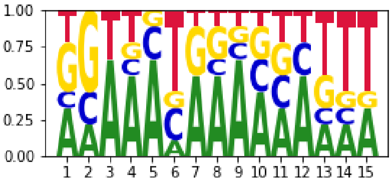
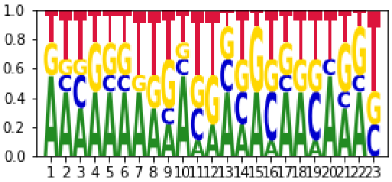
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.35: Motif Logo of “mus04r” for different length

yst04r

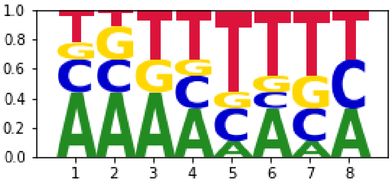
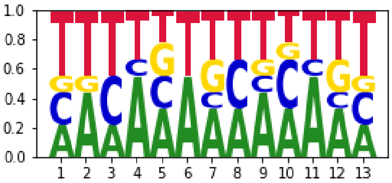
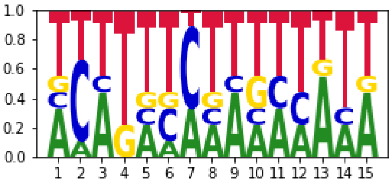
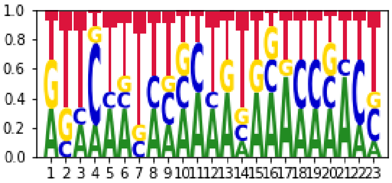
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.36: Motif Logo of “yst04r” for different length

dm05r

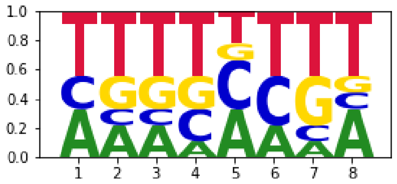
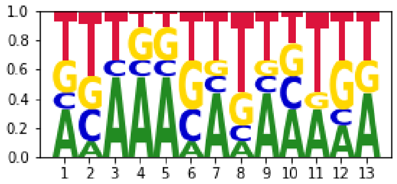
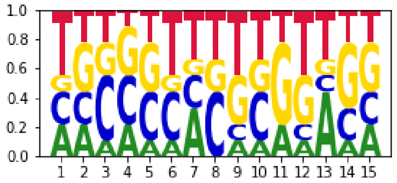
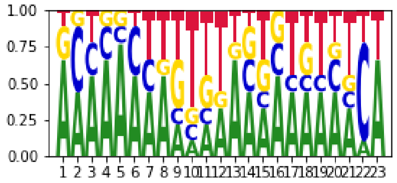
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.37: Motif Logo of “dm05r” for different length

hm05r

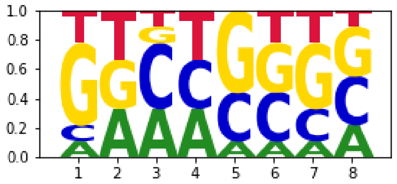
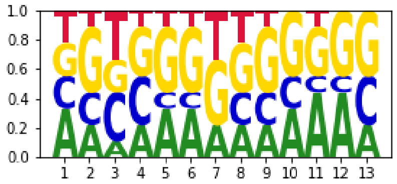
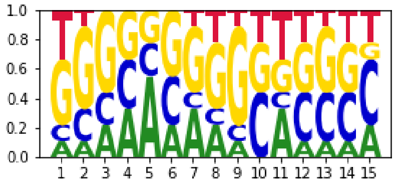
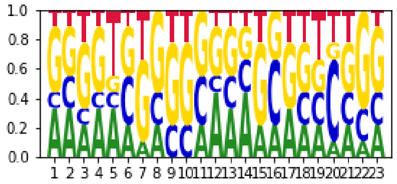
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.38: Motif Logo of “hm05r” for different length

mus05r

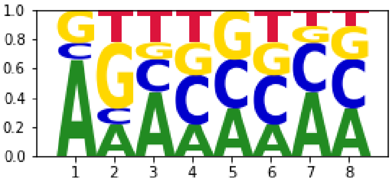
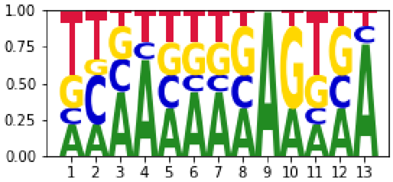
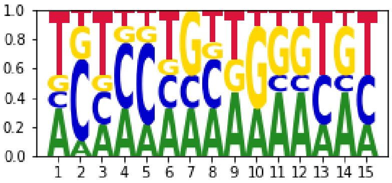
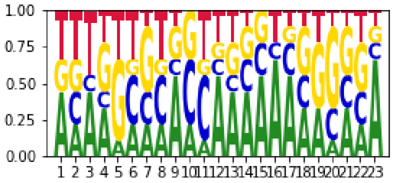
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.39: Motif Logo of “mus05r” for different length

yst05r

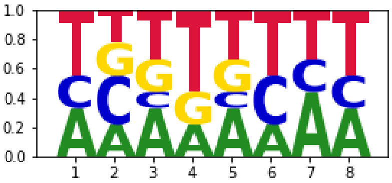
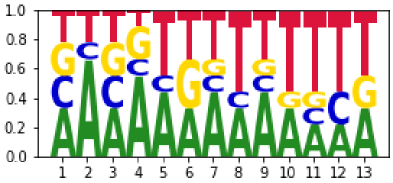
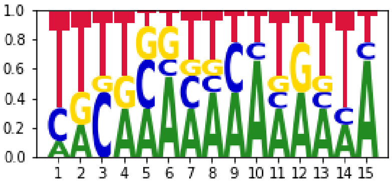
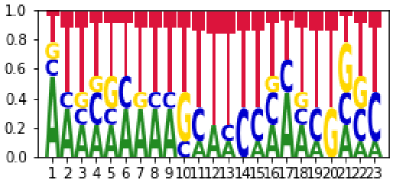
Length = 8	Length = 13
	
Length = 15	**Length = 23**
	

Table 6.40: Motif Logo of “yst05r” for different length