DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY OF MOHAMED BOUDIAF - M'SILA

FACULTY OF MATHEMATICS AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

End of study dissertation for obtaining the Academic Master Degree

DOMAINE: Mathematics and Informatics
FILIERE: Informatics
OPTION: Information Systems and Software Engineering

Entitled

Motif discovery of protein sequences based on Prosite base

Comparison study of different tools

Presented by:

BOUKRAA Djihad **and** BOUKRAA Mohamed Islam

The jury composed by:

Dr. Yaakoubi Rachad	University of M'sila	President
Mr. BENAZI Makhlouf	University of M'sila	Supervisor
Dr. LOUNNAS Bilal	University of M'sila	Co- Supervisor
Mr. Ould mohamedi Nadjib	University of M'sila	Examinator

Academic Year: 2020 / 2021

DEDICATE

Praise to ALLAH, who enabled me to appreciate this step in my academic career the result of effort and success Graduation note, by the grace of ALLAH Almighty, is dedicated to:

My mother, may ALLAH have mercy on her, who was credited with getting me to this stage

To my father, may ALLAH protect him and take care of him, who paved the path of knowledge for us and didn't forget us in his supplication

To my generous family who supported me and are still my brothers and sisters

To my brother and companion in the completion of this note: Mohamed

To my distinguished professors, from the beginning of my academic journey to its completion, God willing

Faculty of Mathematics and Computer science
* Department of Informatics *

To professor and supervisor, Dr. Lounnas Bilal, who made a great effort to accomplish this work

To my friends and everyone who has had

a positive impact on my life.

Boukraa Djihad

DEDICATE

Praise to ALLAH, who enabled me to appreciate this step in my academic career to complete the graduation memorandum thanks to God, dedicate my graduation to:

My mother, may ALLAH have mercy on her, who was credited with getting me to this point.

To my father, may ALLAH preserve and protect him, who paved the way to us and didn't forget us in his supplication

To my generous family who supported me and are still my brothers and sisters

To my sister and companion in the completion of this note: Djihad

To my distinguished professors, from the beginning of my academic journey to its completion, God willing

Faculty of Mathematics and Computer science
* Department of Informatics *

To professor and supervisor, Dr. Lounnas Bilal, who made a great effort to accomplish this work

To my friends and everyone who has had

a positive impact on my life.

Boukraa Mohamed Islam

ACKNOWLEDGMENTS

In The Name of ALLAH, The Most Beneficent, The Most Merciful.

All praise belongs to ALLAH alone, and blessings and peace be upon the final Prophet.

Thanks go to every professor who has informed us from the earliest stages of study to this point.

Faculty of Mathematics and Computer science
* Department of Informatics *

Professors, administrators and colleagues

We also raise our thanks to our professor, Dr. Lonas Bilal.

Discussion Committee: Yaakoubi Rashad, Ould Mohamedi Najib

We also thank everyone who has extended a helping hand to us from near or far.

Contents

List of Figures

List of Tables

1

General Introduction

1.1 preamble:

Pattern recognition is an important field to many computer science domains. most of the pattern recognition researches is focused on string matching which are considered as a very important field in many other research fields. A wide range of research areas benefit from string matching techniques to solve their own problems, one of this fields is bioinformatics Which rely heavily on the use of string matching techniques to solve its problems Which can take longer time as well as a larger effort to solve using traditional techniques, Most biological problems are related to the analysis of biochemical molecules (DNA, RNA, and protein).

Motif discovery is one of the most widely studied problems in bioinformatics ever since genomic and protein sequences have been available. The purpose of motif discovery is to discover patterns in nucleotide or protein sequences in order to better understand the structure and function of the molecules the sequences represent. A motif refers to a region (a subsequence) of protein or DNA sequence that has a specific structure, The presence of a motif may be used as a base of protein classification. The strings are the basic support for data representation and exchange in a simple and more efficient way. String matching is a most important problem in computer science. it aimed to searching a query string (or pattern) in a given text. Generally, the size of the pattern to search of has to be smaller than the given text.

1.2 Problem statement:

Among the problems that caught our attention the problem of online platform for the most of tools used in the analysis and extraction of motif as well as lack of exportation of results with the required format, adding to the need for biologists to use an independent tool and more flexible, a comparison with using the tools provided by electronic sites. In this dissertation we focus on study known techniques for discovering patterns in biological sequences. as well we’re going to do a comparative study of some online analysis tools.

1.3 Motivation:

Nucleotide and protein sequences contain patterns or motifs that have been preserved through evolution because they are important to the structure or function of the molecule. In proteins, these conserved sequences may be involved in the binding of the protein to its substrate or to another protein, may comprise the active site of an enzyme or may determine the three-dimensional structure of the protein. Nucleotide sequences outside of coding regions in general tend to be less conserved among organisms, except where they are important for function, that is, where they are involved in the regulation of gene expression. Discovery of motifs in protein and nucleotide sequences can lead to determination of function and to

elucidation of evolutionary relationships among sequences.

1.4 4. Dissertation outline:

In this dissertation we divide it into three chapters the first chapter contain two sections, the first is bioinformatics which includes some of molecular concepts, as well as we talk about some of its tasks and applications. in the second section we talk about types, systems and algorithms of pattern recognition.

The second chapter too contain two sections, in the first one we talk about motif discovery, representation of motifs and methods, algorithms and tools of motif discovery.in the second section we talk about prosite data base methodology, structure and tools.

The third chapter which is a comparison study of some motif discovery online analysis tools

2

Bioinformatics & Pattern Recognition

2.1 Bioinformatics

Bioinformatics, which will be more clearly defined below, is the discipline of quantitative analysis of information relating to biological macromolecules with the aid of computers. The development of bioinformatics as a field is the result of advances in both molecular biology and computer science over the past 30–40 years. Although these developments are not described in detail here, understanding the history of this discipline is helpful in obtaining a broader insight into current bioinformatics research. A succinct chronological summary of landmark events that have had major impacts on the development of bioinformatics is presented here to provide context. The earliest bioinformatics efforts can be traced back to the 1960s, although the word bioinformatics did not exist then. Probably, the first major bioinformatics project was undertaken by Margaret Dayhoff in 1965, who developed a first protein sequence database called Atlas of Protein Sequence and Structure. Subsequently, in the early 1970s, the Brookhaven National Laboratory established the Protein Data Bank for archiving three-dimensional protein structures. At its onset, the database stored less than a dozen protein structures, compared to more than 30,000 structures today. The first sequence alignment algorithm was developed by Needleman and Wunsch in 1970. This was a fundamental step in the development of the field of bioinformatics, which paved the way for the routine sequence comparisons and database searching practiced by modern biologists. The first protein structure prediction algorithm was developed by Chou and Fasman in 1974. Though it is rather rudimentary by today's standard, it pioneered a series of developments in protein structure prediction. The 1980s saw the establishment of GenBank and the development of fast database searching algorithms such as FASTA by William Pearson and BLAST by Stephen Altschul and coworkers.

The start of the human genome project in the late 1980s provided a major boost for the development of bioinformatics. The development and the increasingly wide spread use of the Internet in the 1990s made instant access to, and exchange and dissemination of, biological data possible.

These are only the major milestones in the establishment of this new field. The fundamental reason that bioinformatics gained prominence as a discipline was the advancement of genome studies that produced unprecedented amounts of biological data. The explosion of genomic sequence information generated a sudden demand for efficient computational tools to manage and analyze the data. The development of these computational tools depended on knowledge generated from a wide range of disciplines including mathematics, statistics, computer science, information technology, and molecular biology. The merger of these disciplines created an information-oriented field in biology, which is now known as bioinformatics [91].

2.1.1 Biology Basics

No matter what type of bioinformatics one is interested in, basic understanding of existing knowledge of biology especially molecular biology is a must[35].

Cells:

The basic component of all organisms is the cell. Many organisms are unicellular, which means one cell itself is an organism. However, for higher species like animals and plants, an organism can contain thousands of billions of cells.

Cells are of two major types: prokaryotic cells and eukaryotic cells. Eukaryotic cells are cells with real nucleus, while prokaryotic cells do not have nucleus. Living organisms are also categorized as two major groups: prokaryotes and eukaryotes according to whether their cells have nucleus.
Prokaryotes are the earlier forms of life on the earth, which includes bacteria and archaea.
All higher organisms are eukaryote, including unicellular organisms like yeasts and higher organisms like plants and animals.
The bacteria E. coli is a widely studied prokaryote. (figure 2.1) shows the structure of an E. coli cell, as a representative of prokaryotic cells.

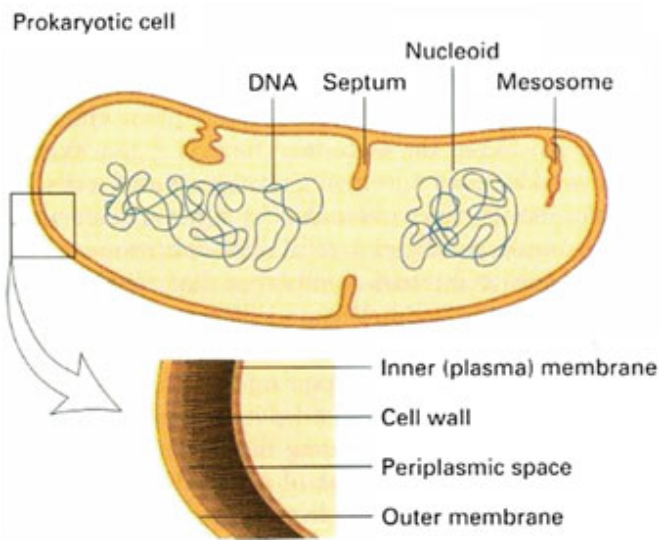


Figure 2.1: A prokaryotic cell

Eukaryotic cells have more complex structures, as shown in the example of a human plasma cell in (figure 2.2) In eukaryotic cells, the key genetic materials, DNA, live in nucleus, in the form of chromatin or chromosomes [35].

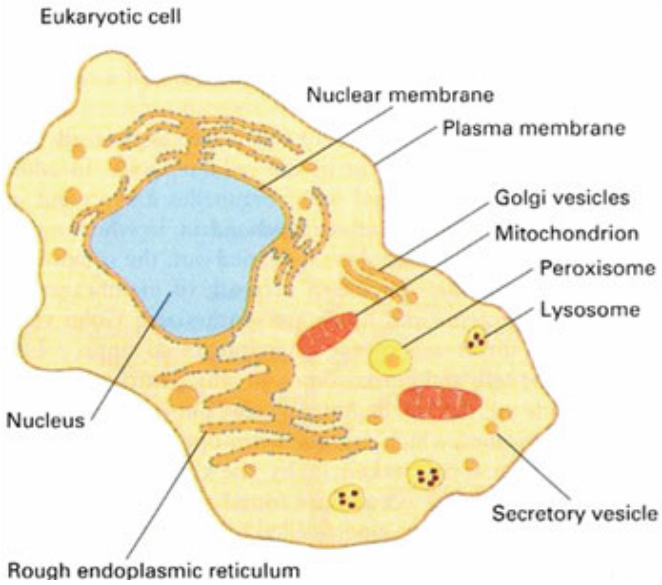


Figure 2.2: A eukaryotic cell

Protein:

Proteins are chains of amino acids. There are 20 types of standard amino acids used in lives. The procedure of translation converts the information from the language of nucleotides to the language of amino acids. The translation is done by a special dictionary: the genetic codes or codon. (figure 2.3) shows the codon table.

First letter	Second letter: U	Second letter: C	Second letter: A	Second letter: G	Third letter
U	UUU, UUC Phenylalanine	UCU, UCC, UCA, UCG Serine	UAU, UAC Tyrosine	UGU, UGC Cysteine	U C A G
	UUA, UUG Leucine		UAA, UAG Stop codon	UGA Stop codon; UGG Tryptophan	
C	CUU, CUC, CUA, CUG Leucine	CCU, CCC, CCA, CCG Proline	CAU, CAC Histidine; CAA, CAG Glutamine	CGU, CGC, CGA, CGG Arginine	U C A G
A	AUU, AUC, AUA Isoleucine	ACU, ACC, ACA, ACG Threonine	AAU, AAC Asparagine; AAA, AAG Lysine	AGU, AGC Serine; AGA, AGG Arginine	U C A G
	AUG Methionine; start codon				
G	GUU, GUC, GUA, GUG Valine	GCU, GCC, GCA, GCG Alanine	GAU, GAC Aspartic acid; GAA, GAG Glutamic acid	GGU, GGC, GGA, GGG Glycine	U C A G

Figure 2.3: The genetic codes

Every three nucleotides code for one particular amino acid. The three nucleotides are called a triplet. Because three nucleotides can encode 64 unique items, there are redundancies in this coding scheme, as shown in (figure 2.3) Many amino acids are coded by more than one codon. For the redundant codons, usually their first and second nucleotides are consistent, but some variation in the third nucleotide is tolerated. AUG is the start codon that starts a protein sequence, and there are three stop codons CAA, CAG, and UGA that stop the sequence.

(Figure 2.4a) illustrates the central dogma in prokaryotes. First, DNA double helix is opened and one strand of the double helix is used as a template to transcribe the mRNA. The mRNA is then translated to protein in ribosome with the help of tRNAs.

(Figure 2.4b) illustrates the central dogma in eukaryotes. There are several differences with the prokaryote case. In eukaryotic cells, DNAs live in the nucleus, where they are transcribed to mRNA similar to the prokaryote case. However, this mRNA is only the preliminary form of message RNA or pre-mRNA. Pre-mRNA is processed in several steps: parts are removed (called spicing), and ends of 150–200 As (called poly-A tail) are added. The processed mRNA is exported outside the nucleus to the cytoplasm, where it is translated to protein [35].

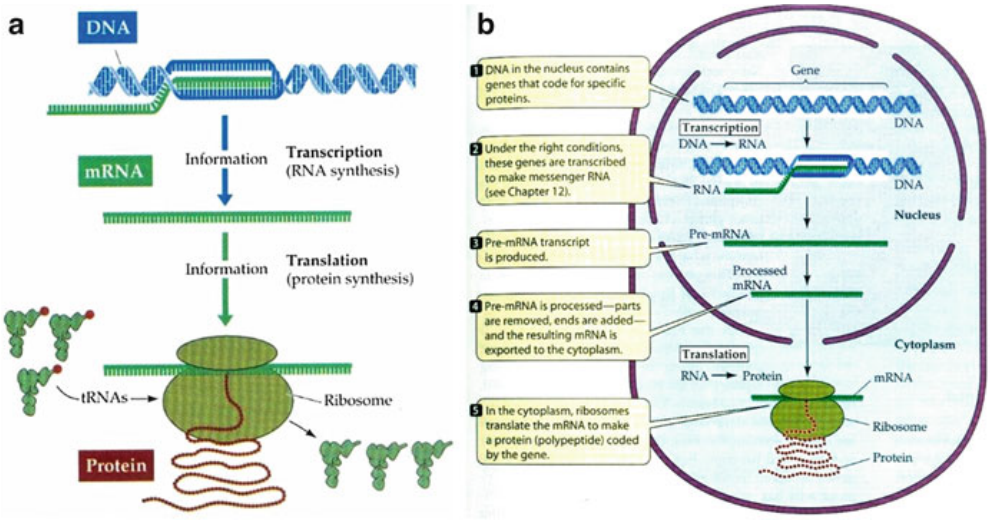


Figure 2.4: The central dogma

DNA, RNA and Chromosome:

Every cell in the human body contains a nucleus, with the exception of red blood cells, which lose this structure as they mature. Within the nucleus are tightly coiled threadlike structures known as chromosomes, each chromosome has within its hundreds or thousands of genes each with a specific location, consisting of the inherited genetic material known as DNA. Here we'll learn about DNA, RNA and chromosome [17].

DNA (Deoxyribonucleic Acid) :

DNA is a long molecule containing the genetic information of the cell and organism. Each cell contains the same DNA molecules (except in case of disease or mutation) in their nucleus. DNA is also found in mitochondria (mitochondrial DNA). The base unit of the DNA is the nucleotide. It is composed of three elements:[46]

- a phosphate element.
- a sugar that is deoxyribose.
- a nitrogen base: adenine (A), guanine (G) (puric bases), thymine (T) and cytosine (C) (pyrimidic bases).

The DNA consists of two strands of nucleotides (strands) forming a double helix see (figure 2.5) They are linked together by weak bonds (hydrogen bonds) at the nucleotides in the following manner: the "A" of one chain binds to the "T" of the other and vice versa, and the "G" of one chain binds to the "C" of the other and vice versa. One helix turn corresponds to ten nucleotide pairs.

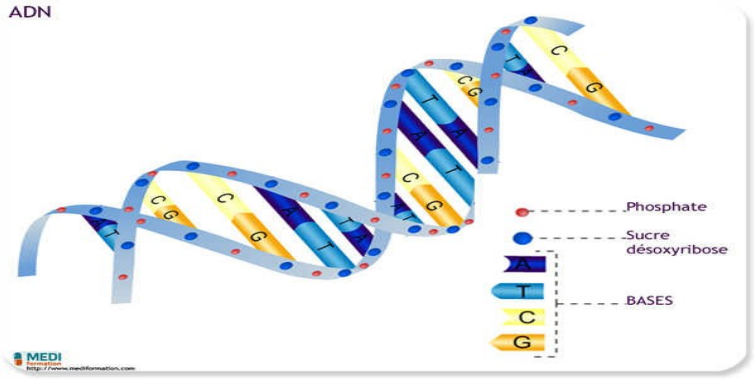


Figure 2.5: the structure of DNA

RNA(Ribonucleic Acid):

The cell synthesizes several types of RNA:[46]
mRNA (messenger RNA): quantitatively the least important but which will reproduce the genetic code in a form such that the information can be transmitted out of the nucleus in the cytoplasm where it will be used to synthesize proteins; the transfer RNA that will be used to transport each amino acid specifically to the place where the proteins will be manufactured.
ribosomal RNA : the most abundant, essential component of ribosomes, is used with ribosomal proteins to catalyze and direct protein synthesis.
newly discovered microRNAs (interference RNA): that play an important role in the regulation of transcription and translation .
RNA differs from DNA in nucleotide "T" where it is replaced by nucleotide "U".

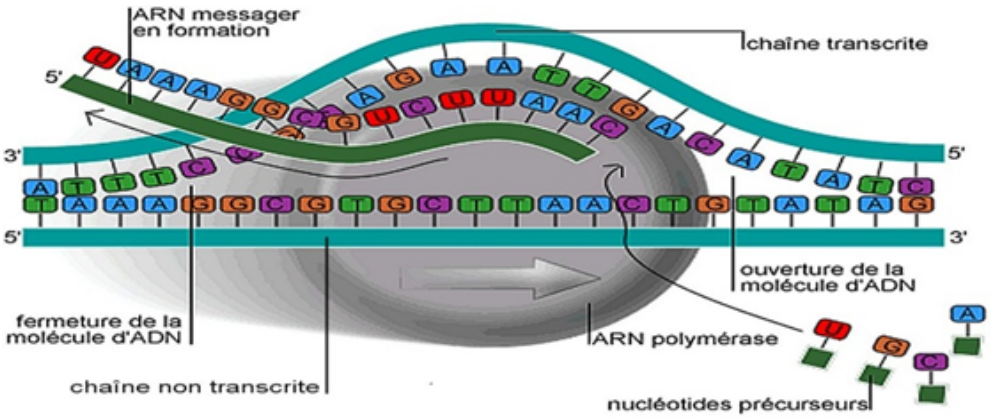


Figure 2.6: the RNA structure

Chromosome:

In the nucleus of each cell, the DNA molecule is packaged into thread-like structures called chromosomes. Each chromosome is made up of DNA tightly coiled many times around proteins called histones that support its structure [42].

The representation of the pairs of chromosomes is called karyotype, chromosomes are lined up in order of size there. To establish a karyotype consists in isolating chromosomes and then in classifying them according to their size, the position of their centromere and their profile of specific bands. Every living kind has a karyotype which is peculiar to it [46].

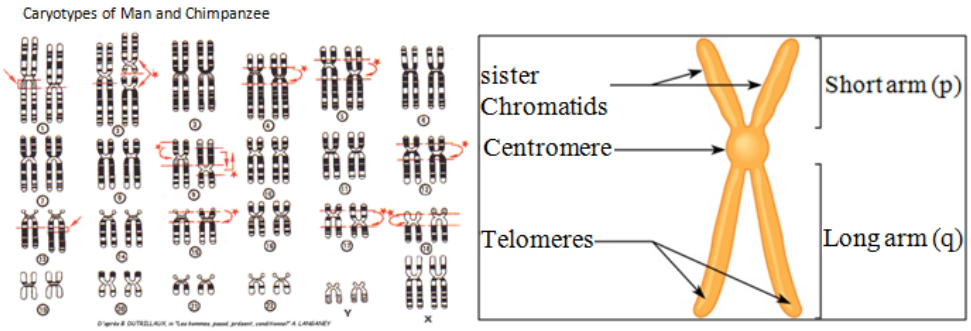


Figure 2.7: Chromosomes structure

Genome and Genomics:

Genome:

The genome of a living organism is the genetic information that allows the organism to live and evolve. It contains all the genetic information necessary for the functioning of the cell and therefore of the whole organism [46].

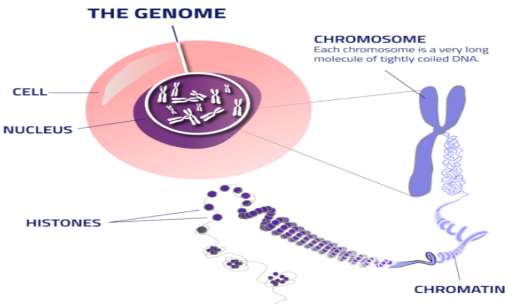


Figure 2.8: The Genome cell [11]

Genomics:

Genomics is the science that aims at the comprehensive study of genomes, it is currently an important scientific challenge on several levels. Genomics makes it possible to study all the genes, of a given species, their function, their role and their distribution on the chromosomes and the relationships between them. A sequenced genome is a text. Consisting of four letters (A, C, G, T), there remains a huge work of decryption to be able. Interpreting this text and exploring molecular structures and processes that are fundamental to life Basically three tasks remain to be done:

- Identification of genes and their function.
- Understanding molecular interaction networks.
- Compare this genome to that of other spaces.

The interests of such work are major :

- Evolution of species (the theory of evolution)
- Cell functioning: understanding the mechanisms of gene regulation.
- Medicine: identify genes that cause disease and explain the causes of complex diseases
- Study of the spread of diseases.
- Pharmaceutical: aid in the design of remedies and treatments.
- Ecology: conservation of fauna and flora.
- Nutrition: Genetically Modified Organisms (GMOs).

DNA Sequencing:

DNA sequencing is the process of determining the nucleic acid sequence – the order of nucleotides in DNA. It includes any method or technology that is used to determine the order of the four bases: adenine, guanine, cytosine, and thymine.

The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery[84].

The sequencing reaction is a key technique that enables the completion of sequencing the human genome. (figure 2.9) illustrates the principle of the widely used Sanger sequencing technique[35].

The technique is based on the complementary base-pairing property of DNA. When a single-strand DNA fragment is isolated and places with primers, DNA polymerase, and the four types of dNTP(deoxyribonucleoside triphosphate) , a new DNA strand complementary to the existing one will be synthesized. In the DNA sequencing reaction, ddNTP(dideoxyribonucleoside triphosphate) is added besides the above components, and the four types of ddNTPs are bound to four different fluorescent dyes. The synthesis of a new strand will stop when a ddNTP instead of a dNTP is added. Therefore, with abundant template single-strand DNA fragments, we'll be able to get a set of complementary DNA segments of all different lengths, each one stopped by a colored ddNTP. Under electrophoresis, these segments of different lengths will run at different speeds, with the shortest segments running the fastest and the longest segments running the slowest. By scanning the color of all segments ordered by their length, we'll be able to read the nucleotide at each position of the complementary sequence and therefore read the original template sequence. This technique is implemented in the first generation of sequencing machines.

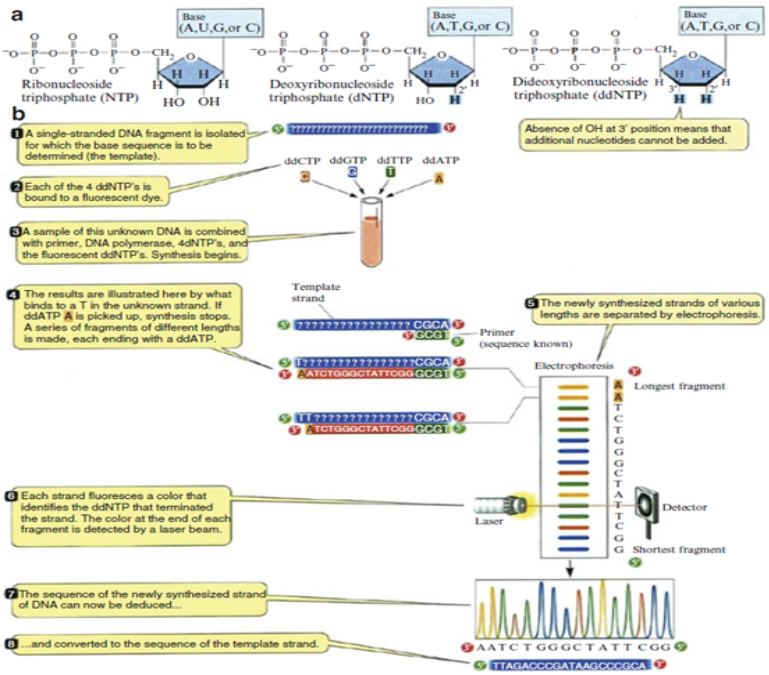


Figure 2.9: Sequencing reaction

The new technology can read huge amount of shorter DNA sequences at much higher efficiency. (figure 2.10) shows the brief concept of sequencing methods.

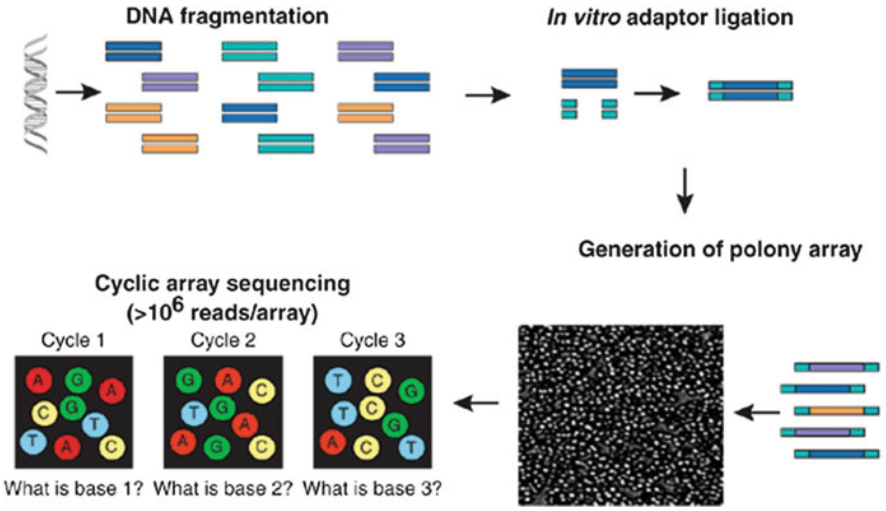


Figure 2.10: Flowchart of the second-generation sequencing (This picture is derived from)

RNA Sequencing:

RNA-Seq (named as an abbreviation of "RNA sequencing") is a sequencing technique which uses next generation sequencing (NGS) to reveal the presence and quantity of RNA in a biological sample at a given moment, analyzing the continuously changing cellular transcriptome. Specifically, RNA-Seq facilitates the ability to look at alternative gene spliced transcripts, post transcriptional modifications, gene fusion, mutations/SNPs and changes in gene expression over time, or differences in gene expression in different groups or treatments. In addition to mRNA transcripts, RNA-Seq can look at different populations of RNA to include total RNA, small RNA, such as miRNA, tRNA, and ribosomal profiling RNA-Seq can also be used to determine exon/intron boundaries and verify or amend previously annotated 5’ and 3’ gene boundaries.

Recent advances in RNA-Seq include single cell sequencing and in situ sequencing of fixed tissue Prior to RNA-Seq, gene expression studies were done with hybridization-based microarrays. Issues with microarrays include cross- hybridization artifacts, poor quantification of lowly and highly expressed genes, and needing to know the sequence a priori Because of these technical issues, transcriptomics transitioned to sequencing-based methods. These progressed from Sanger sequencing of Expressed Sequence Tag libraries, to chemical tag-based methods (e.g., serial analysis of gene expression), and finally to the current technology, next-gene sequencing of cDNA (notably RNA - eq)[88].

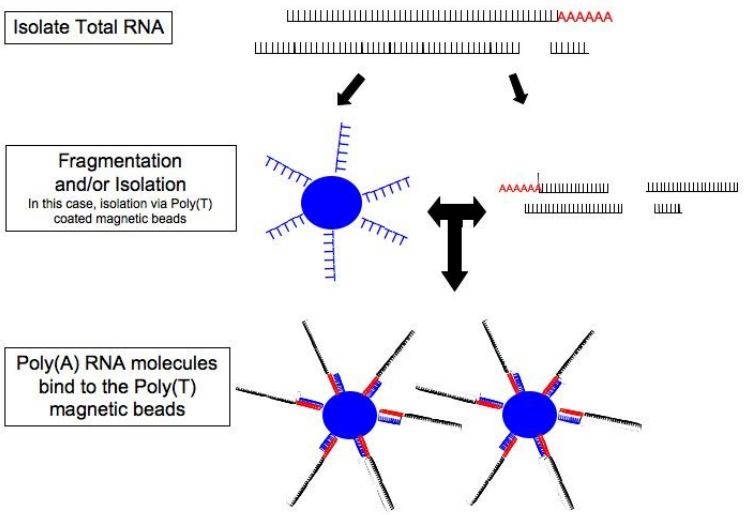


Figure 2.11: RNA Sequencing

Transcriptomics and DNA Microarrays:

Transcriptomics is the study of the transcriptome. The term transcriptome is now widely understood to mean the complete set of all the ribonucleic acid (RNA) molecules expressed in some given entity, such as a cell, tissue, or organism[44].

The genome can be viewed as the original blueprint of the cell. When a gene is to take effect, it is transcribed into mRNAs as the acting copy, according to which proteins are made. This procedure is called the expression of a gene.

When more proteins of some type are needed, more mRNAs will be made. Therefore, the abundance of the mRNA in the cell indicates the expression level of the corresponding gene. The genomic information in all or most cells of an organism is the same, but genes express differently at different developmental stage and in different tissues. many genes show distinctive tissue-specific expression patterns. That means they may be expressed highly in one type of cells but not in other cells. Basic cellular processes are realized by tightly regulated gene expression programs. Therefore, it is important to study the expression profiles of the whole repertoire of genes. The study of all transcripts is called transcriptomes [35]. A DNA microarray (also commonly known as DNA chip or biochip) is a collection of microscopic DNA spots attached to a solid surface. Scientists use DNA microarrays to measure the expression levels of large numbers of genes simultaneously or to genotype multiple regions of a genome[83].

It has been the most commonly used technique during the last two decades to globally monitor cellular abundances of transcript species[78]. The DNA microarray is a key high-throughput technique in transcriptomic investigations. At

the same time, it can measure abundance of mRNAs of thousands or more genes. As mRNAs often degrade rapidly, usually complementary DNAs (cDNAs) reverse transcribed from the mRNAs are used in the measurement. The basic principle of microarrays is also the complementary base-pairing hybridization of DNAs. Pieces of different DNA fragments (called probes) are placed on a small chip. The probes were designed in ways that they can represent individual genes. When the samples' cDNAs are applied on the chip, they'll hybridize with the probes whose sequences are complementary to the cDNA sequences, and those DNAs that do not hybridize to any probe will be washed off.

There are two different types of DNA microarrays: the printed cDNA microarray (cDNA microarray for short) and the oligonucleotide microarray. The major difference is their ways of preparing the probes. (figure 2.12) illustrates the basic principle of the two types of methods[35].

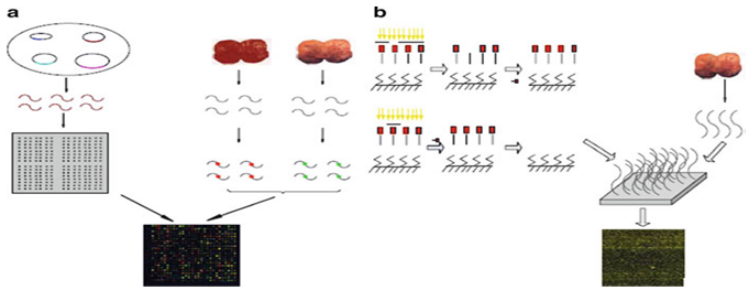


Figure 2.12: DNA microarrays. (a) Printed cDNA microarray. (b) Oligonucleotide microarray

A DNA microarray is a collection of microscopic DNA spots attached to a solid surface. Each DNA spot contains many thousands of copies of a specific DNA sequence, known as probes. These usually correspond to a short section of a gene.

Each microarray includes one or a few probes sets for each interrogated gene. These are used to hybridize a cDNA sample (the target) under high-stringency conditions. Probe–target hybridization is usually detected and quantified by detection of fluorophore-, silver-, or chemiluminescence-labeled targets to determine the relative abundance of transcripts in the target sample. Data on about 700 000 sample hybridizations performed on DNA microarrays are accessible through the databases Gene Expression Omnibus (GEO) at NCBI, and ArrayExpress at EBI.

Because DNA microarrays require spotting of nucleotide probes corresponding to known transcripts, only the abundances of these transcripts can be monitored. Quantification of previously unknown transcripts – often specific to the particular cell type being interrogated, and therefore particularly relevant to the phenotype of this cell type – is impossible. Moreover, probes are usually shared between multiple splice forms of the same gene, and unless specific array designs are employed, it

is impossible to deconvolute the abundances of individual alternative transcript isoforms from overall gene expression[78].

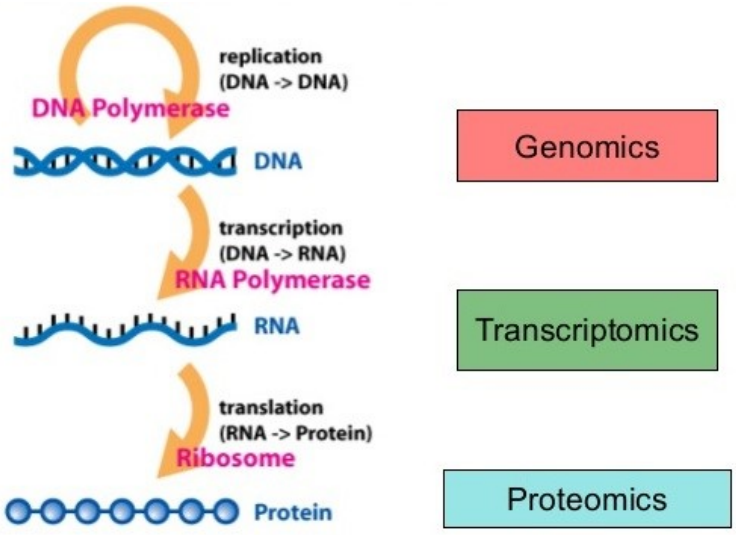


Figure 2.13: relationship between genes and proteins [77]

2.1.2 What Is Bioinformatic?

Bioinformatic is an interdisciplinary research area at the interface between computer science and biological science. According to Luscombe et al, bioinformatic is a union of biology and informatic.

Bioinformatics involves the technology that uses computers for storage, retrieval, manipulation, and distribution of information related to biological macromolecules such as DNA, RNA, and proteins. The emphasis here is on the use of computers because most of the tasks in genomic data analysis are highly repetitive or mathematically complex.

The use of computers is absolutely indispensable in mining genomes for information gathering and knowledge building[91].

There are a lot of other definitions of bioinformatics Among them the following:

Bioinformatics is a science that uses computers, software, and databases to solve, explain, and interpret many biological questions. There are two major biosciences in which bioinformatics is used mainly and increasingly: genomics (study of the composition and function of the total gene pool of an organism) and proteomics (study of the composition and function of the sum total of organism proteins) [4].

On the other hand, bioinformatics can be defined as the computational branch of molecular biology [16]. Bioinformatics differs from a related field known as

computational biology.

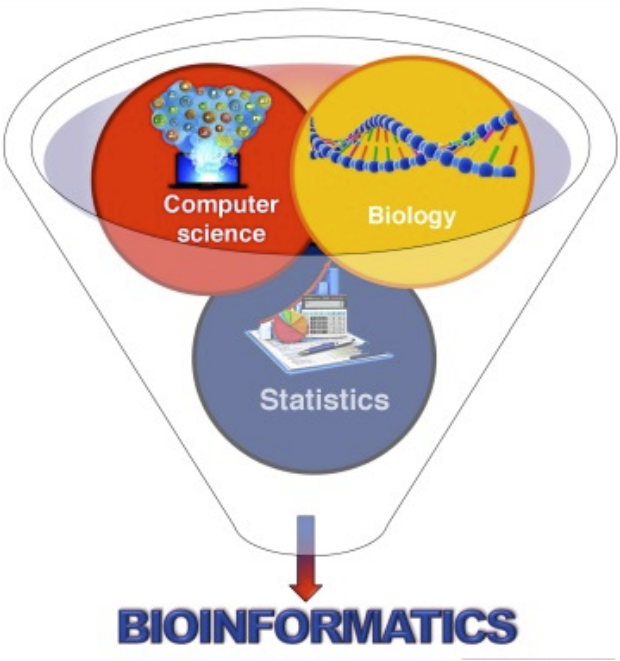


Figure 2.14: Bioinformatics [56].

Bioinformatics is limited to sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered computational molecular biology. However, computational biology encompasses all biological areas that involve computation. For example, mathematical modeling of ecosystems, population dynamics, application of the game theory in behavioral studies, and phylogenetic construction using fossil records all employ computational tools, but do not necessarily involve biological macromolecules [91].

2.1.3 Bioinformatic Goal

The ultimate goal of bioinformatics is a better understanding of the living cell and how it works at the molecular level. By analyzing raw molecular sequences and structural data, bioinformatics research can generate new insights and provide a "global" perspective to the cell. The reason that cell functions can be better understood by analyzing sequence data is ultimately that the flow of genetic information is dictated by the "central doctrine" of biology. The resolution of functional problems using sequencing and sometimes structural approaches has proved to be a productive endeavor [91].
organize vast reams of molecular biology data in an efficient manner.

- develop tools that aid in the analysis of such data.
- interpret the results accurately and meaningfully [45].

2.1.4 Bioinformatic Tasks

Large quantities of sequence data are being published, for organisms from bacteria to higher mammals. It will take decades to analyze the data. Here are a few of the tasks involved in the analysis:[66]

Sequence assembly:

Sequencing involves a “shotgun” approach where DNA fragments 1000bp long are sequenced with significant over coverage. These then have to be “assembled”.

Annotation:

The assembled genomes have to be “annotated”: genes identified and marked out, their functions identified, and so on.

Motif finding:

Non-coding DNA contains “regulatory” regions where proteins called “transcription factors” bind to “turn on” genes. Identifying such regions, and binding sites for individual TFs, is of great importance. TFs typically bind to small “motifs”, so the task is to find over represented short “motifs” in larger quantities of sequence.

Sequence alignment:

In the last two tasks, it is very useful to compare genomes of previously sequenced species. “Comparative genomics” is becoming a very important subfield. Detection and alignment of homologous sequence is an important task here.

Phylogenetic trees:

Given sequence data from different species, it is useful to reconstruct their phylogenetic relationship.

Algorithms exist for all these tasks, but all are evolving with increasing understanding of the function of non-coding DNA, increasing mathematical and algorithmic sophistication in the methods, and increasing raw computational power available to tackle these tasks.

2.1.5 Applications

Bio-informatics has a vast application in genomics, molecular research, and biomedical sciences.

- Sequence Analysis of DNA, RNA, and Proteins.
- **Prediction of protein structure:** from the amino acid sequences while keeping in mind various factors such as hydrogen bonding, shape, polarity, hydrophobicity, and more.
- **Genome annotation:** helps in the identification of genes location along with the other coding regions on a genome and determining what these genes do.
- **Comparative genomics:** comparing and analysis of the genetic materials among different species, helps in studying of functions of genes, their mode of inheritance, and species evolution.
- **Drug discovery:** bioinformatics tools could be used to understand better the mechanisms of the disease, such as identifying the responsible gene sequence, hence to find new drug targets.
- **Pharmacogenomics:** Validate new drug targets and to tailor the medicines to the patients that act on the case/disease origin rather than on the symptoms. Perhaps, could develop drugs with even better therapeutic properties than the existing ones.

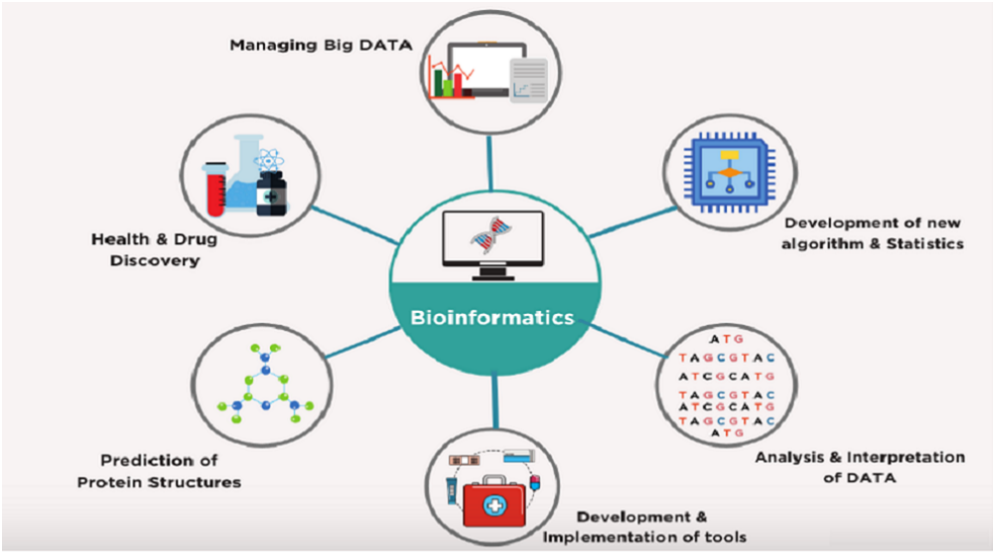


Figure 2.15: Bioinformatics Applications

Bioinformatics has not only become essential for basic genomic and molecular biology research but is having a significant impact on many areas of biotechnology

and biomedical sciences.
Other vital areas of bio-informatics include: biological databases, molecular phylogenetics, personalized medicines, metabolomics, and transcriptomics (analysis of RNA expression)[48].

2.1.6 Limitations

Having recognized the power of bioinformatics, it is also important to realize its limitations and avoid over-reliance on and over-expectation of bioinformatics output.

In fact, bioinformatics has a number of inherent limitations [91], Among them:

- Completely relying on the information is dangerous if the info is inaccurate.
- Quality of bioinformatics predictions depends on:
  - Quality of the data.
  - Sophistication of the algorithms.
- - Bioinformatics and experimental biology are complementary:
  - Bioinformatics results need to be consistent with Experimental biology.
- Sequence data contain errors.
- Downstream interpretation of sequence data will be wrong if the sequences are or the annotation there of is wrong.
- Many algorithms lack capability and sophistication to truly reflect reality.
- Outcome of computation also depends on available computing power [79].

2.1.7 Conclusion

Bioinformatics has become an important part of many areas of biology, bioinformatic techniques such as image and signal processing allow extraction of useful results from large amounts of raw data. In the field of genetics and genomics [23].

Bioinformatics has got major impacts on biotechnology and its application. The vast amount of data generated by human genome project or by another genome sequencing project would be unmanageable without the bioinformatics technique.

Bioinformatics also quickened the drug discovery, vaccine design and also the design of anti-microbial agents. Bioinformatics is also used to understand gene and also genome [62].

2.2 Pattern Recognition

Pattern recognition is a useful tool for the analysis of behavior of nonlinear complex systems in absence of fundamental equations describing them. Using this methodology creates possibility for a so called "technical" analysis that involves a heuristic search for relationships between available system information and its features inaccessible for direct measurements [37].

The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories. Its ultimate goal is to optimally extract patterns based on certain conditions and to separate one class from the others. The application of Pattern Recognition can be found everywhere.

Human beings are pattern recognizers, not just because of this recognition ability, but especially because we are aware of it. We can handle it and also teach the patterns to others and discuss with them our observations. The ability to judge the similarity between objects or events is called generalization. The question of how our mind travels from observations to memory and to generalization is thereby the basic scientific question of pattern recognition and how this process can be integrated in and taught to a computer, a challenge for its algorithms.

Pattern Recognition can be implemented with the use of Machine Learning algorithms. These algorithms perform classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation [36].

RECOGNIZING the objects and the surrounding environment is a trivial task for human beings. But if the point of implementing it artificially came, then it becomes a very complex task. Pattern Recognition provides the solution to various problems from speech recognition, face recognition to classification of handwritten characters and medical diagnosis.

The various application areas of pattern recognition are like bioinformatics, document classification, image analysis, data mining, industrial automation, biometric recognition, remote sensing, handwritten text analysis, medical diagnosis, speech recognition, GIS(Geographic Information System) and many more.

Three processes take place in pattern recognition task. First step is data acquisition. Data acquisition is the process of converting data from one form (speech, character, pictures etc.) into another form which should be acceptable to the computing device for further processing. Data acquisition is generally performed by sensors, digitizing machine and scanners. Second step is data analysis. After data acquisition the task of analysis begins. During data analysis step the learning about the data takes place and information is collected about the different events and pat-

tern classes available in the data. This information or knowledge about the data is used for further processing. Third step used for pattern recognition is classification. Its purpose is to decide the category of new data on the basis of knowledge received from data analysis process. Data set presented to a Pattern Recognition system is divided into two sets: training set and testing set. System learns from training set and efficiency of system is checked by presenting testing set to it. The performance of the pattern recognition techniques is influenced by mainly three elements (i) amount of data (ii) technology used(method) (iii) designer and the user. The challenging job in pattern recognition is to develop systems with capability of handling massive amounts of data. The various models opted for pattern recognition are:

Statistical Techniques, Structural Techniques, Template Matching, Neural Network based techniques, Fuzzy models and Hybrid Models [19].

2.2.1 What is pattern recognition?

Pattern: is everything around in this digital world.

A pattern can either be seen physically or it can be observed mathematically by applying algorithms. Example: The colors on the clothes, speech pattern etc.

Some other definitions of pattern recognition:

Pattern recognition is the process of recognizing patterns by using machine learning algorithm. Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation [60].

Pattern recognition is a process of finding regularities and similarities in data using machine learning data [64].

Pattern recognition is the scientific discipline that allows us to classify objects into several categories or classes that can be further used to perform analysis and improve certain things [29].

Pattern recognition is a cognitive process that happens in our brain when we match some information that we encounter with data stored in our memory [90].

2.2.2 Pattern recognition types

Pattern recognition is the scientific discipline that allows us to classify objects into several categories or classes that can be further used to perform analysis and improve certain things. The best-known approaches for pattern recognition are:

Template matching:

Template Matching is used to determine the similarity between two entities (points, curves, or shapes) of the same type. The pattern to be recognized is matched with a stored template along with geometrical transformations. This approach has some obvious disadvantages of being too rigid and having the need for lots of templates [29].

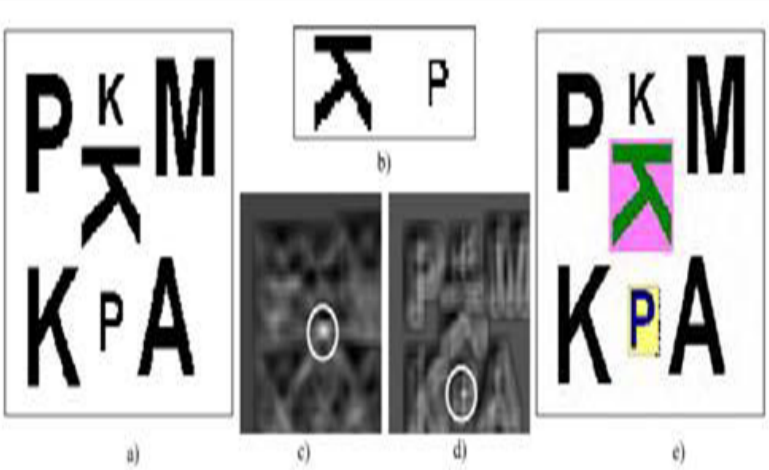


Figure 2.16: example of template matching model for pattern recognition [65]

Statistical classification:

In this method, each pattern is represented in terms of some features or measurements. The main objective of this approach is to establish decision boundaries in the feature space. This separates patterns belonging to different classes creating some rules for an inter-class boundary [29].

Syntactic or structural matching:

This method works on a hierarchy framework where a pattern is said to be composed of simple sub-patterns that are themselves built from yet simpler sub-patterns. Considered equivalent to languages where primitives are alphabets which make words then lines than the page and then documents. [29]

String matching:

String matching is used in almost all the software applications straddling from simple text editors to the complex NIDS(Network Intrusion Detection Systems).[15]

String matching is to find all occurrences of a pattern in a given text. Is to Find a given pattern p [1...m] in text T [1...n] with n>=m. P occurs with shift

s (beginning at s+1): P [1] =T[s+1], P [2] =T[s+2], . . . , P[m]=T [s + m].
example: the pattern P=abab, T=abcababababbc, P occurs at s=3 and s=5.

- text is the string that we are searching.
- pattern is the string that we are searching for.
- Shift is an offset into a string.

Basic Classification:

String matching can be classified to four algorithms:

1. Naive algorithm:
   The naive approach for solving the string searching problem is accomplished by performing a Brute-Force comparison of each character in the pattern at each possible placement of the pattern in the string. This algorithm is O(mn) in the worst case.
2. Rabin – Karp algorithm:
   String matching algorithm that compares string's hash values, rather than string themselves. Performs well in practice, and generalized to other algorithm for related problems, such as two-dimensional pattern matching.
3. Knuth-Morris-Pratt algorithm:
   It is improved on the Brute-force algorithm and the new algorithms capable of running O(m+n) in the worst case. This algorithm improves the running time by taking advantage of tagged borders.
4. Boyer-Moore algorithm:
   The idea behind the Boyer-Moore algorithm is information gain Here information is gained by beginning the comparison from the end of the pattern instead of the beginning. It performs the string searching task in sub linear time in the average case, which even KMP algorithm could not accomplish at that time.

The problem of string matching:

Given a string 'S', the problem of string-matching deals with finding whether a pattern 'p' occurs in 'S' and if 'p' does occur then returning position in 'S' where 'p' occurs.

O(mn) approach: One of the most obvious approach towards the string-matching problem would be to compare the first element of the pattern to be searched 'p', with the first element of the string 'S' in which to locate 'p'. If the first element of 'p' matches the first element of 'S', compare the second element of 'p' with second element of 'S'. If match found proceed likewise until entire 'p' is found. If a mismatch is found at any position, shift 'p' one position to the right and repeat comparison beginning from first element of 'p'.

2.2.3 Pattern recognition system

A pattern recognition system can be regarded as a process that allows it to cope with real and noisy data. Whether the decision made by the system is right or not mainly depending on the decision made by the human expert [41].

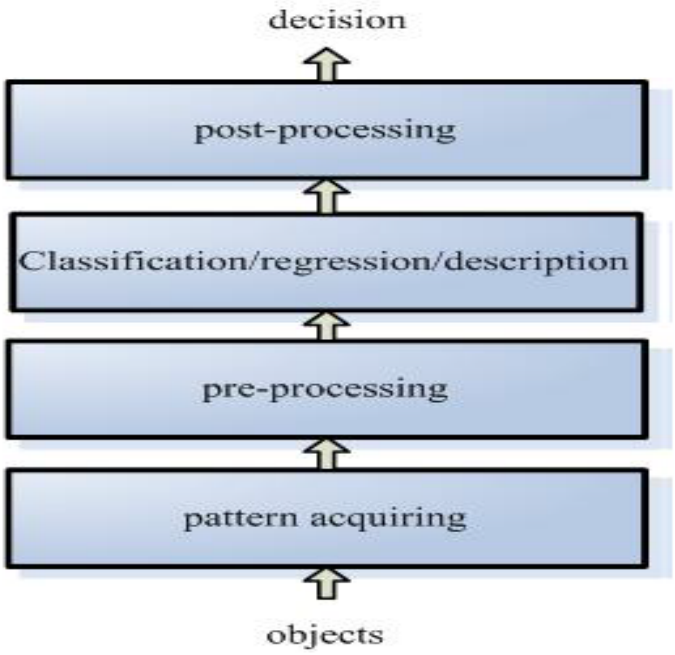


Figure 2.17: the composition of a PR system

The structure of pattern recognition system:

A pattern recognition system based on any PR method mainly includes three mutual associate and differentiated processes. One is data building; the other two are pattern analysis and pattern classification. data building converts original information into vector which can be dealt with by computer. Pattern analysis task is to process the data, such as feature selection, feature extraction, data dimension compress and so on.

The aim of pattern classification is to utilize the information acquired from pattern analysis to discipline the computer in order to accomplish the classification.

A very common description of the pattern recognition system that includes five steps to accomplish. The step of classification, regression and description showed in (figure 2.17) is the kernel of the system. Classification is a PR problem of assigning an object to a class, the output of the PR system is an integer label, such as classifying a product as “1” or “0” in a quality control test.

Regression is a generalization of a classification task, and the output of the PR

system is a real-valued number, such as predicting the share value of a firm based on past performance and stock market indicators. Description is the problem of representing an object in terms of a series of primitives, and the PR system produces a structural or linguistic description. A general composition of a PR system is given below.

The classification of pattern recognition system:

The classification of pattern recognition system - Rule based system.

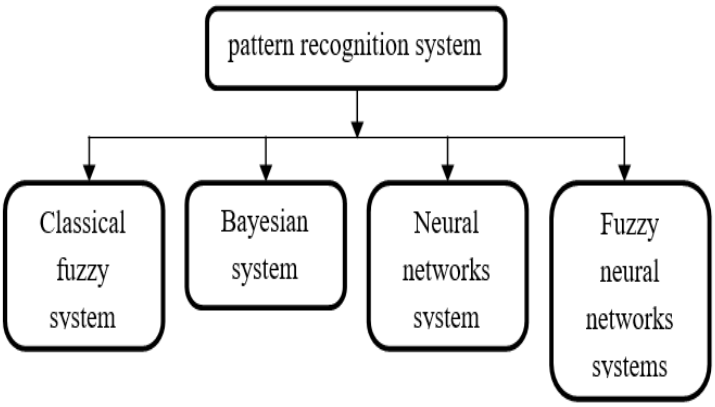


Figure 2.18: the composition of a PR system

These are mainly classification of PR system, whether the system is successful mainly depends on his decision like an expert or not.

2.2.4 Pattern recognition algorithms

PR algorithms can be categorized into three types:

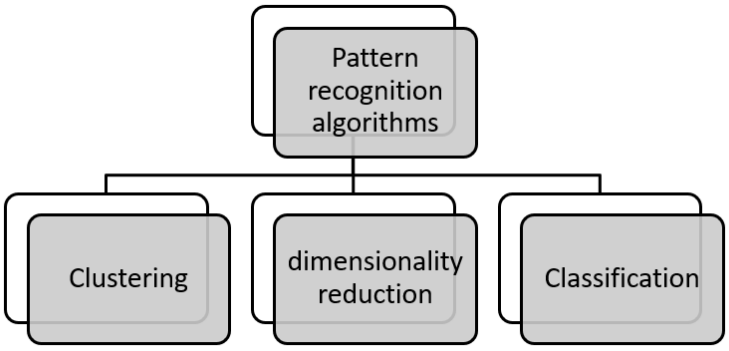


Figure 2.19: Types of pattern recognition algorithms

Clustering:

A first natural question to ask when exploring a large dataset is whether it clusters, i.e., contains distinct subgroups of similar objects. Such groups may lead to new insights.

There are two main types of clustering algorithms:

1. partitional:
   a certain (simple) cluster model is assumed, and the fit of a number of such models to the data is optimized. Examples of partitional methods are k-means and GMMs(Gaussian mixture models), but there is a large body of literature on different clustering methods.

2. hierarchical clustering:
   a dendrogram is constructed, usually by iteratively grouping objects and clusters that are most similar (figure 2.20a). This dendrogram can subsequently be cut at a certain level to end up with a specific number of clusters (figure 2.20b). Hierarchical clustering has become popular in the microarray era to help create heatmaps. Irrespective of the type of algorithm used, a user defines or assumes - perhaps implicitly - what number of clusters to look for, when objects are similar.

   It is important to realize that different assumptions can lead to very different groupings. A clustering is therefore never an objective result: finding that objects group is not proof (yet) of a relation, but at most a starting point for further experimentation [59].

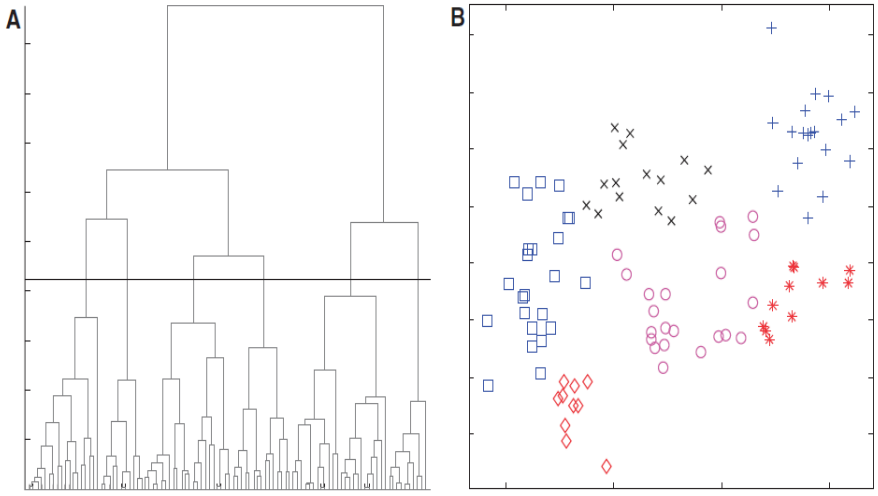


Figure 2.20: Hierarchical clustering. (A) The dendrogram joins objects and clusters; the height of the stem indicates their distance. (B) The clustering resulting when the dendrogram is cut at the dotted line

dimensionality reduction:

Dimensionality reduction is especially useful in determining sparse feature representations by discarding noisy or irrelevant measurements or by combining relevant measurements into a smaller number of features [59].

There are two types of benefits for applying dimensionality reduction by feature selection for the classification process:

firstly, by eliminating unnecessary features, it is possible to eliminate dataset noise that degrades the quality of the classification model.

secondly, the problem dimension is decreased and the efficiency is increased. The criteria for dimensionality reduction can vary, and in terms of classification problems are usually referred to the classification accuracy, model efficiency, level of dimension reduction, or composition of former criteria. Dimensionality reduction algorithms have been utilized as a support for solving various and often real problems. There are three general types of dimensionality reduction algorithms:[25]

1. wrapper methods that use a classification algorithm as a black box for the evaluation of feature subsets.
2. filter methods that form feature subsets in the preprocessing phase, and do not depend on the employed classification algorithm.
3. embedded methods that form a feature subset in the training process and are specific to a given classification algorithm .

Classification:

Classification algorithm can be classified to eight algorithms:

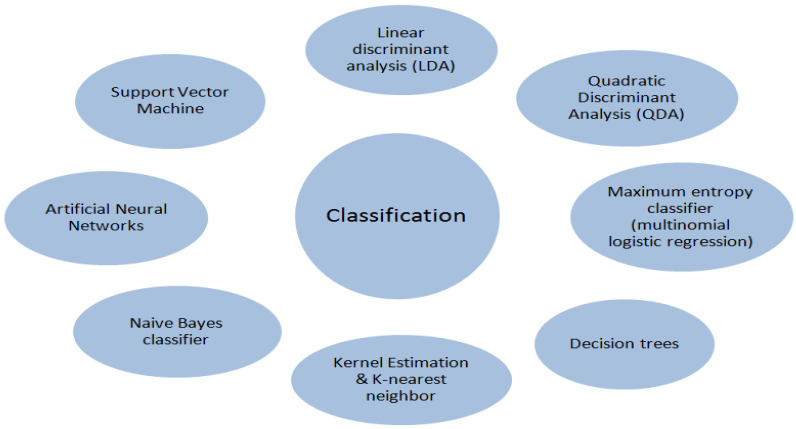


Figure 2.21: Classification Algorithms

1. Linear discriminant analysis (LDA):
Linear Discriminant Analysis (LDA) assumes that the classes follow multivariate Gaussian distributions with different means but all have the same covariance matrix [7, p.22 23]. That means, in the density function Equation (1), the value of $\Sigma_t = \Sigma, \forall t = 1, ..., T$. We call it shared covariance. The equation of shared covariance is:

$$\Sigma = \frac{1}{N} \sum_{p=1}^{N} \left(x_p - m_{class(x_p)}\right) \left(x_p - m_{class(x_p)}\right)^T$$

(2) In Equation (2), N is the number of points, $x_p$where $p \in \{1, ..., N\}$ are the points in the data set. All classes share one covariance matrix – that means, the shape of all classes is the same[94].

2. Quadratic Discriminant Analysis (QDA) :
Quadratic Discriminant Analysis (QDA) only assumes that each class obeys the Gaussian distribution with its own parameter settings. So, each class has its own covariance matrix, and the shapes of the classes can be different. Figure 47 shows the differences between Quadratic Discriminant Analysis and Linear Discriminant Analysis. From the diagram we can see, for Linear Discriminant Analysis, the shapes of all the classes stay the same and they can be separated by a straight line. But for Quadratic Discriminant Analysis, the shapes of different classes are different, and the separating boundary is a quadratic curve. Quadratic Discriminant Analysis is more powerful than Linear Discriminant Analysis, for Quadratic Discriminant Analysis does not require all the classes have identical covariance[94].

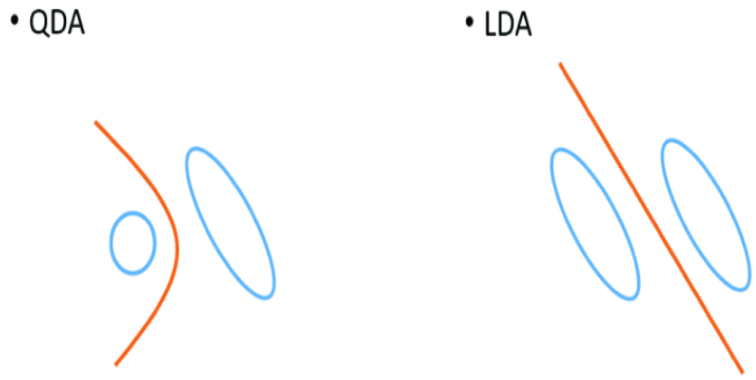


Figure 2.22: Quadratic and Linear discriminant decision boundaries

3. Maximum entropy classifier (multinomial logistic regression) :
In statistics, a maximum entropy classifier model is a regression model which generalizes logistic regression by allowing more than two discrete outcomes. This forms a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given

a set of independent variables (which may be real-valued, categorical-valued etc.). The actual goal of the multinomial logistic regression model is to predict the categorical data. Maximum entropy classifiers are commonly used as an alternative to Naive Bayes classifiers because they do not require statistical independence of the independent variables (commonly known as features) that serve as the predictors. This algorithm may not be appropriate to learn large number of classes since it is slower than for a Naive Bayes classifier. Multinomial logistic regression is a particular solution to the classification problem that assumes that a linear combination of the observed features and some problem-specific parameters can be used to determine the probability of each particular outcome and the best values of the such parameters for a given problem are usually determined from some training data [65].

4. Decision trees :

   - A Visual Representation of Choices, Consequences, Probabilities, and Opportunities [47].
   - A Way of Breaking Down Complicated Situations to Easier to Understand Scenarios, By applying:
     - Logic
     - Likely Outcome
     - Quantitative decision

Figure 2.23: decision trees

5. Kernel Estimation & K-nearest neighbor :

In pattern recognition or classification, the k-nearest neighbor algorithm is a technique for classifying objects based on closest training examples in the problem space. K-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the

simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor. The k-NN algorithm can also be adapted for use in estimating continuous variables. One such implementation uses an inverse distance weighted average of the k-nearest multivariate neighbors. This algorithm functions as follows:[54]

a) Compute Euclidean or Mahalanobis distance from target plot to those that were sampled.

b) Order samples taking for account calculated distances.

c) Choose heuristically optimal k nearest neighbor based on RMSE(Root Mean Square Error) done by cross validation technique.

d) Calculate an inverse distance weighted average with the k-nearest multivariate neighbors.

6. Naive Bayes classifier : Naive Bayes classifier is a simple, probabilistic and statistical classifier which is based on Bayes theorem (from Bayesian statistics) with strong (naive) independence assumptions and maximum posteriori hypothesis. As Bayesian classifiers are statistical in nature, they can predict the probability of a given sample belonging to a particular class. The underlying probability model to this classifier can be termed more appropriately as an “independent feature model” because a naive Bayes classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. Such an assumption is called class conditional independence. It is made to simplify the computation involved and, in this sense, is considered “naive”. We can explain this classifier with a small example. A fruit is considered to be an apple if it is red in color, round in shape, and around 5" in diameter. Although these features depend on each other or upon the existence of the other features, a naive Bayes classifier takes all of these properties to independently contribute to the probability that this fruit is an apple. The naive Bayes classifier is trained using a supervised learning approach that just requires consideration of each attribute in each class separately. So the training in naive Bayes classifier is considered to be very easy and fast. To estimate the parameters in naive Bayes model it uses the principle of maximum likelihood method in many practical applications. Testing in this algorithm is also very straightforward and simple; just look the tables and calculate conditional probabilities with normal distributions. The advantage of Naive Bayes model is that it only requires a small amount of training data to estimate the parameters, i.e., means and variances of the variables which are necessary for classification [65].

7. Artificial Neural Networks : It is an interconnected network of a group of artificial neurons. An artificial neuron can be considered as a computational model which is inspired by the natural neurons present in human brain. Unlike natural neurons, the complexity is highly abstracted when modeling artificial neurons. These neurons basically consist of inputs (like synapses), which are further multiplied by a parameter known as weights (strength of each signal), and then computed by a mathematical function which determines the activation of the neuron. After this there is another function that computes the output of the artificial neuron (sometimes in dependence of a certain threshold). Thus, the artificial networks are formed by combining these artificial neurons to process information. We can train ANN for best matched solution;

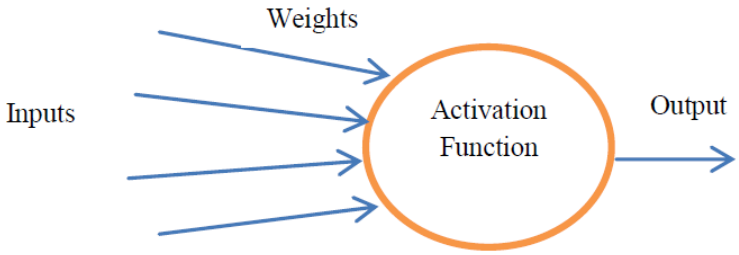


Figure 2.24: An artificial neuron

ANN can perform fuzzy matching and provides the optimal solution. It also acts as a classifier in pattern recognition. It falls under the category of supervised learning where the model initially learns from the training data set and then classifies the test image using the learnt knowledge [65].

8. Support Vector Machine :

The support vector machine has been chosen because it represents a framework both interesting from a machine learning perspective. A SVM is a linear or non-linear classifier, which is a mathematical function that can distinguish two different kinds of objects. These objects fall into classes, this is not to be mistaken for an implementation. To work with SVM we use leaner kernel for implementation. In linear algebra and functional analysis, the kernel of a linear operator L is the set of all operands v for which L(v) = 0.

That is, if L : V $\rightarrow$ W, then $ker\,(L)\,\{v \in V : L\,(v) = 0\}$ where 0 denotes the null vector in W. The kernel of L is a linear subspace of the domain V.

The kernel of a linear operator Rm $\rightarrow$ Rn is the same as the null space of the corresponding n $\times$ m matrix. Sometimes the kernel of a linear operator is referred to as the null space of the operator, and the dimension of the kernel is referred to as the operator's nullity [54].

2.2.5 Applications

It is true that application was one of the most important elements for PR theory. Pattern Recognition has been developed for many years, and the technology of PR has been applied in many fields such as artificial intelligence, computer engineering, nerve biology, medicine image analysis, archaeology, geologic reconnoitering, space navigation, armament technology and so on. Detailed applications, such as below:[41]

- Computer vision
  - The first vision system presented was supposing the objects with geometric shapes and optimized edges extracted from images.
- Computer aided diagnosis
  - Medical imaging, EEG, EEG signal analysis Designed to assist physicians
- Character recognition
  - Automated mail sorting, processing bank checks.
  - Scanner captures an image of the text.
  - Image is converted into constituent characters.
- Speech recognition
  - Human computer interaction.
  - Universal access, Microphone records acoustic signal.
  - Speech signal is classified into phonemes and words.
- Safety
  - Face recognition.
  - Identifying fingerprints.
- Astronomy
  - Classifying galaxies by shape
  - Astronomical telescope image analysis
  - Automatic spectroscopy
- Bioinformatics
  - DNA sequences analysis
  - DNA micro array data analysis

  - Research of heredity
- Agriculture
  - Output analysis
  - Soil evaluating
  - Extraction mineral characterization in coffee and sugar
- Geography
  - Earthquake analysis
  - Rocks classification
- Engineering
  - Fault diagnosis for vehicle system
  - Recognition of automobile Type
  - Improve the safety performance of automobile
- Military affairs
  - Aviation photography analysis
  - Automatism Aim recognition

2.2.6 Conclusion

Pattern recognition is nearly everywhere in our life, each case relevant to decision, detection, retrieval can be a research topic of pattern recognition. The mathematics of pattern recognition is widely-inclusive, the methods of game theory, random process, decision and detection, or even machine learning can used for this task, which make pattern recognition a really wide research [14].

As we are moving away from the conventional form of data to the stage of big data, the analytics of such huge data through pattern recognition techniques would be of tremendous advantage for any industry. There is a lot of similarity between the data from various industries and our machine learning algorithms are also getting smarter day by day. So, the process of finding a pattern is becoming more intuitive and at the same time, its demand is also increasing [29].

In its broadest sense pattern recognition is the heart of all scientific inquiry, including understanding ourselves and the real-world around us. And the developing of pattern recognition is increasing very fast, the related fields and the application of pattern recognition became wider and wider [41].

3

Motif Discovery & Prosite Database

3.1 Motif discovery

Biology has been transformed by the availability of numerous complete genome sequences for a wide variety of organisms, ranging from bacteria and viruses to model plants and animals to humans. Genome sequencing and analysis is constantly evolving and plays an increasingly important part of biological and biomedical research. This has led to new challenges related to the development of the most efficient and effective ways to analyze data and to use them to generate new insights into the function of biological systems. The completion of the genome sequences is just a first step toward the beginning of efforts to decipher the meaning of the genetic "instruction book." Whole genome sequencing is commonly associated with sequencing human genomes, where the genetic data represent a treasure trove for discovering how genes contribute to our health and well being. However, the scalable, flexible nature of NGS (Next Generation Sequencing) technology makes it equally useful for sequencing any species, such as agriculturally important livestock, plants, or disease related microbes.

The exponential increase in the size of the datasets produced by this new generation of instruments clearly poses unique computational challenges. A single run of a NGS machine can produce terabytes of data. Efficient treatment of all this data will require new computational approaches in terms of data storage and management, but also new effective algorithms to analyze the data and extract useful knowledge.

The major challenge today is to understand how the genetic information encoded in the genome sequence is translated into the complex processes involved in the organism and the effects of environmental factors on these processes. Bioinformatics plays a crucial role in the systematic interpretation of genome information, associated with data from other high throughput experimental techniques, such as structural genomics, proteomics, or transcriptomics. A widely used tool in all these stages is the comparison (or alignment) of the new genetic sequences with existing sequences. Identification of coding regions involves alignment of known genes to the new genomic sequence. functional significance is most often assigned to the protein coding regions by searching public databases for similar sequences and by transferring the pertinent information from the known to the unknown protein. A wide variety of computational algorithms have been applied to the sequence comparison problem in diverse domains, notably in natural language processing [61].

Motif discovery is one of the sequence analysis problems under the application layer and it is one of the significant difficulties in bioinformatics applications. A DNA motif refers to a short similar repeated pattern of nucleotides that has biological meaning. Sequence motifs also called regulatory elements exist in RR (Regulatory Region) in eukaryotic gene.

Sequence motifs have constant size and are often repeated and conserved, but

at the same time, they are tiny (about 6-12 bp) and the intergenic regions are very long and highly variable that make motif discovery a problematic task. These patterns play an essential role in recognizing TF-BSs (Transcription Factor Binding Sites) that help in learning the mechanisms for regulation of gene expression. Different types of motifs are planted motifs, structured motifs, sequence motifs, gapped motifs and network motifs [33].

3.1.1 What is Motif Discovery?

In Bioinformatics, a sequence motif is a nucleotide or amino-acid sequence pattern that is widespread and has been proven or assumed to have a biological significance [7].

Motif: is a set of sites where each site is a continuous range of positions representing a subsequence from a DNA sequence [3].

Motifs: are frequently occurring patterns. Motifs in biological sequences can indicate the presence of certain biological characteristics. In general, these could represent patterns in any kind of biological sequences such as DNA sequences, RNA sequences, protein sequences, etc. In DNA, a motif may correspond to a protein binding site; in proteins, a motif may correspond to the active site of an enzyme or a structural unit necessary for proper folding of the protein [53].

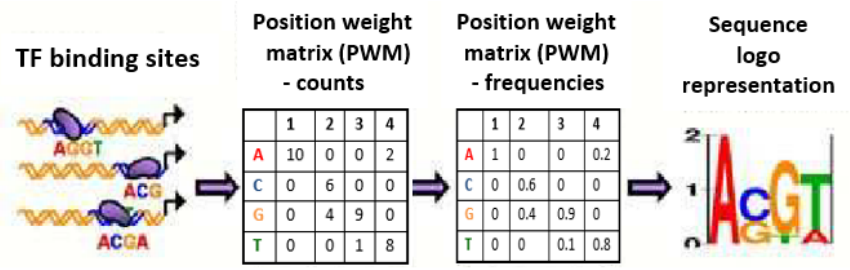
Motif: is a short sequence that represents an expression of characterizing biological function [8].

Motif Discovery: In Bioinformatics, a sequence motif is a nucleotide or amino-acid sequence pattern that is widespread and has been proven or assumed to have a biological significance [7].

Motif Discovery (or 'motif finding'): in biological sequences can be defined as the problem of finding short similar sequence elements (building the 'motif') shared by a set of nucleotide or protein sequences with a common biological function [92].

3.1.2 Representation of motifs

Over the years, a variety of motif representation models have been developed to take into account the complexity of protein motifs. The models are attempts to construct generalizations based on known functional motifs, and are used to help characterize the functional sites and to facilitate their identification in unknown protein sequences. They can be divided into two main categories [61].



	1	2	3	4
A	10	0	0	2
C	0	6	0	0
G	0	4	9	0
T	0	0	1	8

	1	2	3	4
A	1	0	0	0.2
C	0	0.6	0	0
G	0	0.4	0.9	0
T	0	0	0.1	0.8

Figure 3.1: Motif representation [32]

Deterministic models:

There are two main kinds of deterministic models: regular expressions and consensus sequences. The regular expressions used in motif discovery denote a subset of regular languages and are typically composed of exact symbols, ambiguous symbols, fixed gaps and/or flexible gaps. A consensus sequence represents a collection (or neighborhood) of binding site sequences that are at most at a certain Hamming distance1 of the underlying consensus sequence. Each binding site sequence in this collection is called a motif occurrence or consensus occurrence. The number of mismatches depends largely on the size of the motif. There are a few variants to these two models. A first alternative imposes a restriction on the location of mismatches along the consensus sequence. That is, a consensus occurrence can present at most a certain number of mismatches in the first i nucleotides, and so on. On the other hand, a second variant takes into account the sum of mismatches between all consensus occurrences and the underlying consensus sequence [13].

Probabilistic models:

The most widely used probabilistic model is without doubt the PWM (position weight matrix), also known as PSSM (position specific scoring matrix), that assumes independence between positions.The score of an aligned substring is the log-likelihood of the substring under a product multinomial distribution. PWM scores can also be described in a physical framework as the sum of binding energies for all nucleotides aligned with the PWM [63].

Probabilistic models can be used to overcome such loss of information. The PSSM (position specific scoring matrix), also known as the probability weight matrix (PWM), is undoubtedly one of the most widely used probabilistic models. This model is represented by a matrix where each entry (i,a) is the probability of finding an amino acid a at the ith position in the sequence motif. For example, for a set of motifs: [61]

WSEW

WSRW

CSKW

CSKW

YSKY

The corresponding PSSM is shown in (Table 3.1).

Position	1	2	3	4
C	0.4	0.0	0.0	0.0
E	0.0	0.0	0.2	0.0
K	0.0	0.0	0.6	0.0
R	0.0	0.0	0.2	0.0
S	0.0	1.0	0.0	0.0
W	0.4	0.0	0.0	0.8
Y	0.2	0.0	0.0	0.2

Table 3.1: Example of a position specific scoring matrix (PSSM)

Although in this example, PSSM containing entries having a value of 0, in general, pseudo counts are applied, especially when using a small dataset, in order to allow the calculation of probabilities for new motifs. The information summarized in the PSSM can also be represented by a sequence logo, which is a graphical representation of the motif conservation as shown in (figure 3.2). A logo consists of a stack of letters at each position in the motif, where the relative sizes of the letters indicate their frequency in the sequences. The total height of the letters corresponds to the information content of the position, in bits.

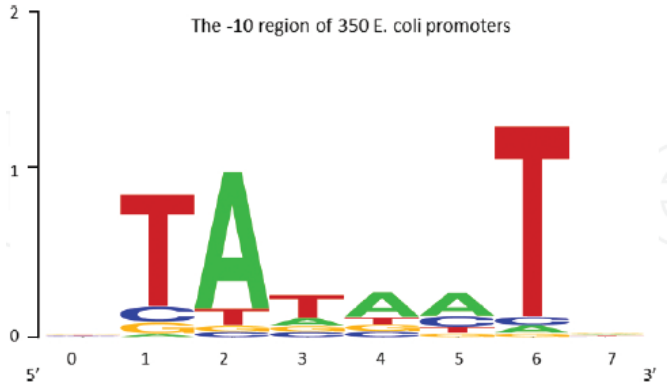


Figure 3.2: An example of a sequence logo for representing patterns in biological sequences. The logo represents the Pribnow box, a conserved region found upstream of some genes in prokaryotic genomes.

Another widely used probabilistic model is the HMM (Hidden Markov Model), a statistical model that is generally applicable to time series or linear sequences. They were first introduced in bioinformatics for DNA sequences. An HMM can be visualized as a finite state machine that moves through a series of states and produces some kind of output. The HMM generates a protein sequence by emitting amino acids as it progresses through a series of states. Each state has a table of amino acid emission probabilities, and transition probabilities for moving from state to state [61].

The probabilistic models are much more expressive than the deterministic models. In fact, all oligos, regular expressions and mismatch expressions can be represented as PWMs. However, a major benefit of the deterministic models is that they often allow exhaustive discovery of optimal motifs [63].

3.1.3 Motif discovery methods

Given a set of functionally related sequences, the main aim of motif discovery algorithms is to find new and a priori unknown motifs that are frequent, unexpected, or interesting according to some formal criteria. The methods used to discover such motifs follow the same general schema, as shown in figure 3.3. They can be grouped into two main categories: alignment-based methods and methods that search for motifs in unaligned sequences [61].

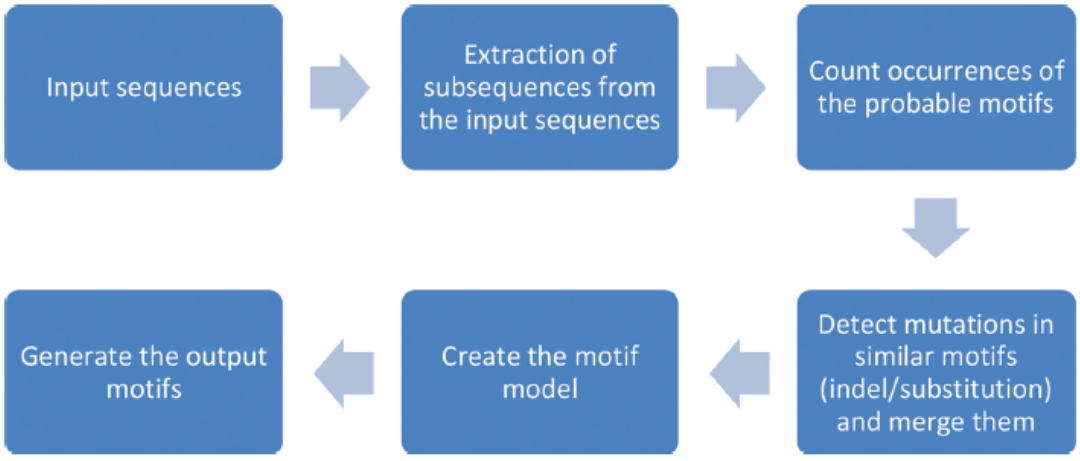


Figure 3.3: General motif discovery process

Alignment-based methods:

Alignment-based methods for motif discovery first construct a multiple sequence alignment of the set of sequences, where each sequence of amino acids is typically represented as a row within a matrix. Gaps are inserted between the amino acids so that identical or similar characters are aligned in successive

columns. Once the multiple alignments are constructed, the patterns are extracted from the alignment by combining the substrings common to most of the sequences.

One of the first automatic methods for the identification of conserved positions in a multiple alignment was the AMAS (Alignment Manipulation And Summary) program, using a set-based description of amino acid properties.

Since then, a large number of different methods have been proposed. For example, Al2Co calculates a conservation index at each position in a multiple sequence alignment using weighted amino acid frequencies at each position. The DIVAA (Directed In Vivo Angiogensis Assay) method is based on a statistical measure of the diversity at a given position. The diversity measures the proportion of the 20 possible amino acids that are observed.

The advantage of the alignment-based approach is that no upper limit has to be imposed on the length of the motifs. Moreover, these algorithms usually do not need as input a maximum threshold value for the motif distance from the sequences. In general, this approach works well if the sequences are sufficiently similar and the patterns occur in the same order in all of the sequences. Unfortunately, this is not usually the case and therefore most methods for motif discovery in protein sequences assume that the input sequences are unaligned.

Alignment-free methods:

The vast majority of motif discovery methods in bioinformatics are alignment-free approaches that do not rely on the initial construction of a multiple sequence alignment. Instead, they generally search for patterns that are overrepresented in a given set of sequences. The simplest solution is to generate all possible motifs up to a maximum length l, and then to search separately for the approximate occurrences of each motif in the set of sequences. Once a list of candidate patterns is obtained, the ones with the highest significance scores are selected. This approach guarantees to find all motifs that satisfy the input constraints. Moreover, the sequences can be organized in suitable indexing structures, such as suffix trees, etc., so that the time needed by the algorithm to search for a single motif is linear in the overall length of the sequences.

This simplistic approach has an evident disadvantage: the number of candidate motifs, and therefore the time required by the algorithm, grows exponentially with the length of the sequences. Computing a significance score for each motif further increases the time required by the algorithm. A number of more efficient tools have been developed to address these issues and in the next chapter, we will discuss some of the more widely used ones.

3.1.4 Motif discovery techniques

The motif discovery technique consists of three main stages [33].

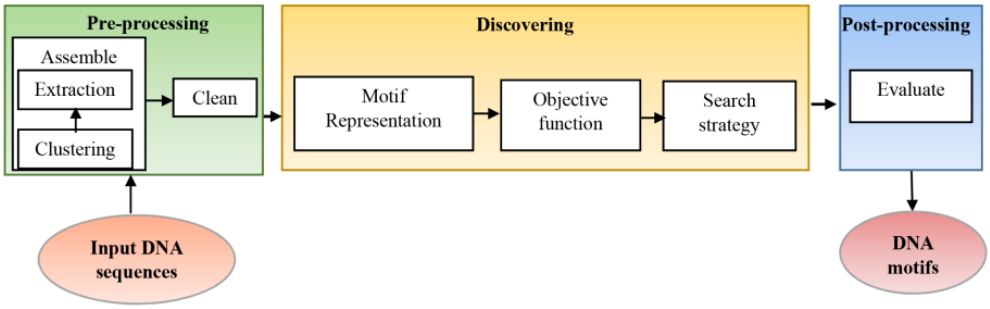


Figure 3.4: General block diagram of motif discovery technique

1. Pre-processing:
   It is preparing the DNA sequences for accurate motif discovery by assembling and clean steps. In assembling step, it is advised to select as many target sequences as possible that may contain motifs, try to keep sequences as short as possible, and remove sequences that are unlikely to contain any motifs. Assembling step is done by clustering the input sequences based on some information and then extracting the de-sired sequences in an appropriate sequence database. Then, cleaning the input sequences to mask or remove confounding sequences is necessary.

2. Discovering:
   The middle stage is the motif discovery approach that begins by representing the sequences. There are two ways to represent the motifs: consensus string and Position-specific Weight Matrices (PWM). Consensus string has the same length of DNA sequence motif; it allows to degenerate symbols in a string using IUPAC (International Union of Pure and Applied Chemistry) code while PWM is a matrix of 4 x m where m is the motif length. Every position in the matrix represents the probability of each nucleotide at each index position of the motif. After motif representation, the suitable objective function is determined and finally appropriate search algorithm is applied. There are hundreds of algorithms for motif extraction that most of them are listed in (Table 3.2).

3. Post-processing: Post-processing evaluates the resultant motifs. This paper presents a more general classification of the sequence motifs extraction methods. Most of them are mentioned with a comparison among them.

3.1.5 Motif discovery algorithms

There are four types of motif discovery algorithms:enumerative, probabilistic, nature inspired and combinatorial [33].

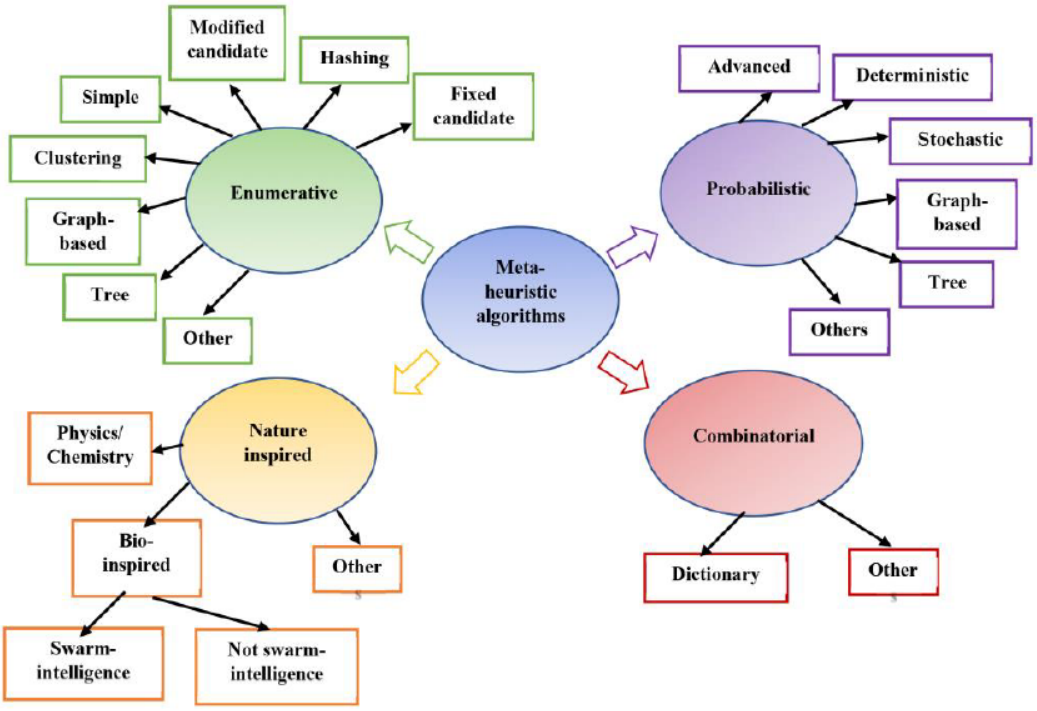


Figure 3.5: Classification of motif discovery algorithms as enumerative, probabilistic, nature inspired and combinatorial types

1. Enumerative approach (word enumeration approach)
   are exponential-time algorithms that require a long time to detect the larger l and inefficient for handling dozens of sequences.

   - search the whole search space to determine which ones appear with possible substitutions.
   - searches for consensus sequences.
   - they are only suitable for short motifs.
   - require many parameters determined by the users such as motif length, the number of mismatches allowed, and a minimum number of sequences that the motif has to appear in.
   - The word enumeration approach can be accelerated by using specialized data structures such as suffix trees or parallel processing.

   The enumerative approach can be classified into many classes: [33]

   a) Simple word enumeration

      The first class is based on simple word enumeration. Some existing al-

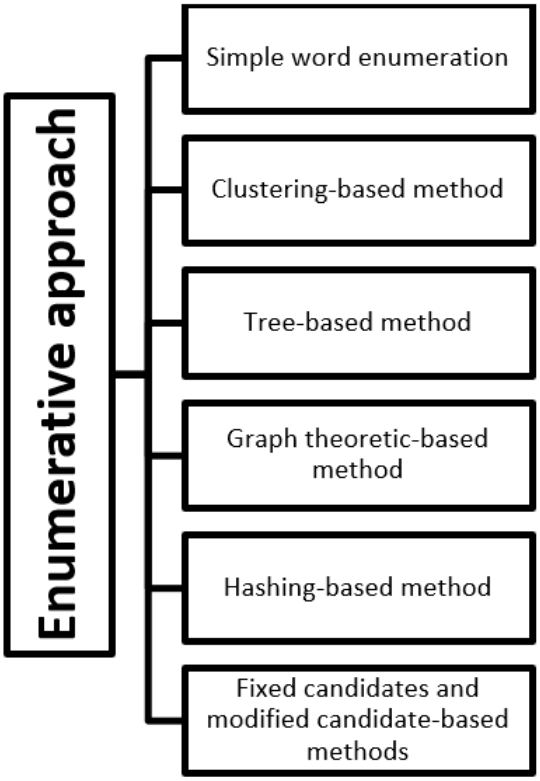


Figure 3.6: Enumerative approach classes

gorithms in this class are YMF and DREME.

- YMF (Yeast Motif Finder) algorithm developed by Sinha et al. It detects short motifs with a small number of degenerate positions in yeast genomes using consensus representation.
  - YMF enumerates all motifs in the search space approach.
  - calculates the z-score to produce those motifs with greatest z-scores.
- DREME (Discriminative Regular Expression Motif Elicitation) algorithm proposed by Bailey et al. It calculates the significance of motifs using Fisher's Exact test.

  The algorithm starts with generating a set of short k-mers, followed by applying Fisher's Exact test on two sets of DNA sequences (Input set and back-ground set) using a significance threshold to calculate the significance of each word (No wildcards) of length three to eight that occurs in the positive sequences and select the most significant words for being used in the inner loop where they passed as "seed" REs to perform a beam search that determines the most significant

generalizations of them (One wildcard).

b) Clustering-based method

Sharov et al proposed word clustering method called CisFinder to detect short motif with high processing speed in large sequences (up to 50 Mb). Firstly, one should define nucleotide substitution matrix for each n-mer word, then calculate PFMs (Position Frequency Matrices) for n-mer word counts with and without gaps in both test and control sets.

c) Tree-based method

Pavesi et al presented Weeder algorithm based on count matching patterns with specific and most extreme mismatches.

At first, the motifs are represented using consensus sequence and based on the difference between the k-mers of the input sequences and the consensus under a limited number of substitutions, k-mers are assembled and each group is evaluated with a specific measure of significance.

d) Graph theoretic-based method

The graph-theoretic method represents a motif in-stance, as a clique; the graph G is built by representing each l-mer in the input sequences by vertex and the edge between a pair of vertices representing a pair of l-mer in different input sequences having the Hamming distance between the substrings which is less than or equal to 2d. Then, cliques of size N are searched for in this graph. Popular graph-theoretic methods are WIN-NOWER, Pruner, and cWINNOWER.

e) Hashing-based method

Buhler et al developed random projection algorithms for a PMP that projects every l-mer in the input data into a smaller space by hashing. Initially, a projection of l-dimensional space onto a k-dimensional sub-space for all subsequences in the input set is developed, and random projection is constructed by choosing random k positions from l position. Using this projection, each l-mer is hashed to its corresponding bucket. After projections, each bucket contains l-mer more than a threshold and this is called qualified bucket. Random hashing is repeated n times to ensure the qualified bucket at least more than once. Finally, profile for each of them should be computed to get the most probable l-mer in the sequence that was represented as consensus sequences. In previous studies, random projection was developed using uniform projection and low-dispersion projection algorithms, respectively.

f) Fixed candidates and modified candidate-based methods

there is a proposed algorithm called Ref-Select to select reference

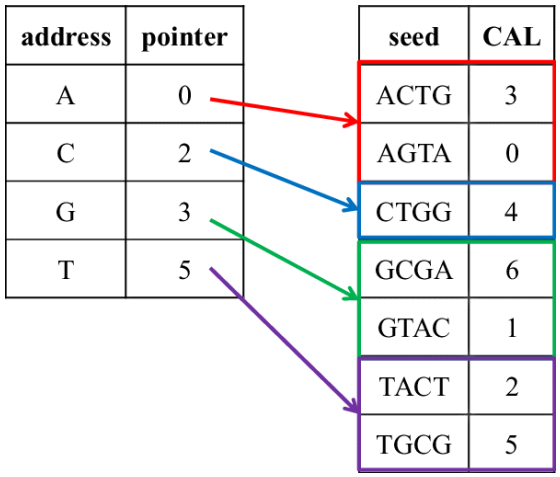


Figure 3.7: Hashing example [49]

sequences for PMP. The reference sequences are the sequences that don't contain motif instances, so, this method tries to select the reference sequences that generate a small number of candidate motifs as possible. The algorithm consists of two steps; firstly, for every two sequences in input dataset D, the number of candidate motifs generated from them should be computed using the Hamming distance between every two l-mers. Then, the set with candidate motifs, as small as possible, is selected as a reference set.

2. Probabilistic approach

It constructs a probabilistic model called PSWM (position Specific Weight Matrix) or motif matrix that specifies a distribution of bases for each position in TFBS to distinguish motifs vs. non-motifs and it requires few search parameters [33].

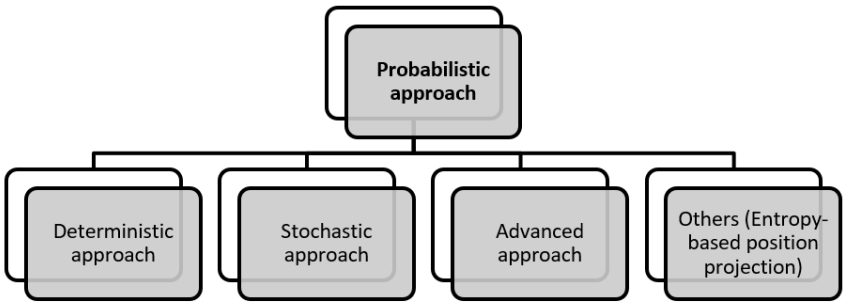


Figure 3.8: Probabilistic approach classes

a) Deterministic approach

EM (Expectation Maximization) is the famous example of deterministic approach. EM for motif finding was first introduced by Lawrence et al. it consists of two main steps:

1. the first called "Expectation step" that estimates the values of some set of unknowns based on a set of parameters.
2. The second step is "Maximization step" that uses those estimated values to refine the parameters over several iterations.

EM is used to identify conserved areas in unaligned DNA and proteins with an assumption that each sequence must contain one common site, the parameters, in this case, they are the entries in the PWM (position weight matrix) and the background nucleotide probabilities while our unknowns are the scores for each possible motif position in all of the sequences.

There are several algorithms based on EM. MEME (Multiple EM for Motif Elicitation) is a popular motif recognition program that optimizes PWMs using the EM algorithm. It has several versions. The idea of MEME algorithm is to find an initial motif and then use expectation and maximization steps to improve the motif until the values in the PWM do not improve or the maximum number of iterations is reached.

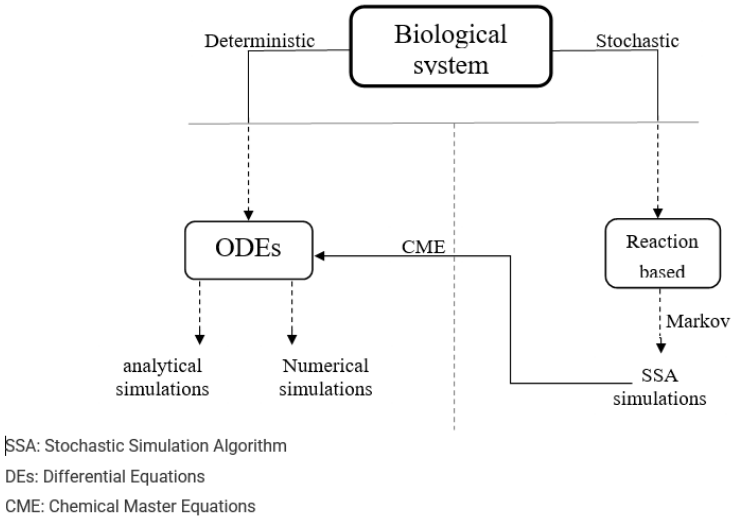


Figure 3.9: The modeling schema in Systems Biology [12]

b) Stochastic approach

Gibbs sampling is a famous stochastic approach, similar to EM algorithm. Pseudo code of the Gibbs sampling algorithm for motif detection follows these steps:

1. Random initializing of motif positions in the input N sequences with an assumption of the presence of one motif per sequence.
2. Choosing one sequence at random.
3. Computing PWM for the other N-1 sequences using staring positions of motifs and background probabilities for each base using the non-motif positions.
4. Calculating probability of each possible motif location in the removed sequence using PWM and back-ground probabilities.
5. For the removed sequence, choosing a new starting position based on step 4

Steps 2-5 should be iterated until the values in the PWM do not improve or the maximum number of iterations has been reached.

Many methods have been developed that implement the concept of Gibbs' sampling to extend its functionality. Hughes et al proposed Align ACE (Aligns Nucleic Acid Conserved Elements) algorithm based on Gibbs' sampling with some improvements:

- The motif model was changed to fit the source genome because the base frequencies for non-site sequence is fixed.
- Both strands of the input sequence are considered and no circumstance overlapping is al-lowed.
- Iteratively, aligned sites were masked out to find multiple different motifs.
- It uses an im-proved near-optimum sampling method.

BioProspector algorithm is also based on Gibbs sampling with several improvements:

- It uses a Markov model estimated from all promoter noncoding sequences to represent the non-motif background in order to improve the motif specificity.
- It can find two-block motifs with variable gap.
- Sampling with two thresholds allows every input sequence to include zero or multiple copies of the motif.

c) Advanced approach

Different algorithms were proposed based on Bayesian approach. Jensen et al proposed an algorithm based on Bayesian approach with Markov chain Monte Carlo.

Xing et al proposed LOGOS (Integrated Local and Global motif sequence model) algorithm that combines between HMDM (Hidden Markov

Dirichlet-Multinomial) for local alignment model for each different motif and HMM (Hidden Markov model) for global motif distribution model for the occurrence of multiple motifs.

Recently, Siebert et al developed a BaMM (Bayesian Markov Model) approach that trains higher order Markov models to build the dependency model. BaMM algorithm is more complex than PWMs wherein the PWMs cannot model correlations among nucleotides because PWMs nucleotide probabilities are independent of nucleotides at other positions. In the proposed algorithm, Bayesian approach using Markov models makes optimal use of the available information while avoids training by the decrease in number of parameters.

the different algorithms were presented like clustering methods based on Bayesian approach.

d) Others

EPP (Entropy-based position projection) algorithm was proposed to escape from local optima. This algorithm based on projection process depends on the relative entropy in each position of motif instead of random projection.

3. Nature-inspired algorithms

Nature-inspired algorithms are classified according to the sources of inspiration into three main categories.

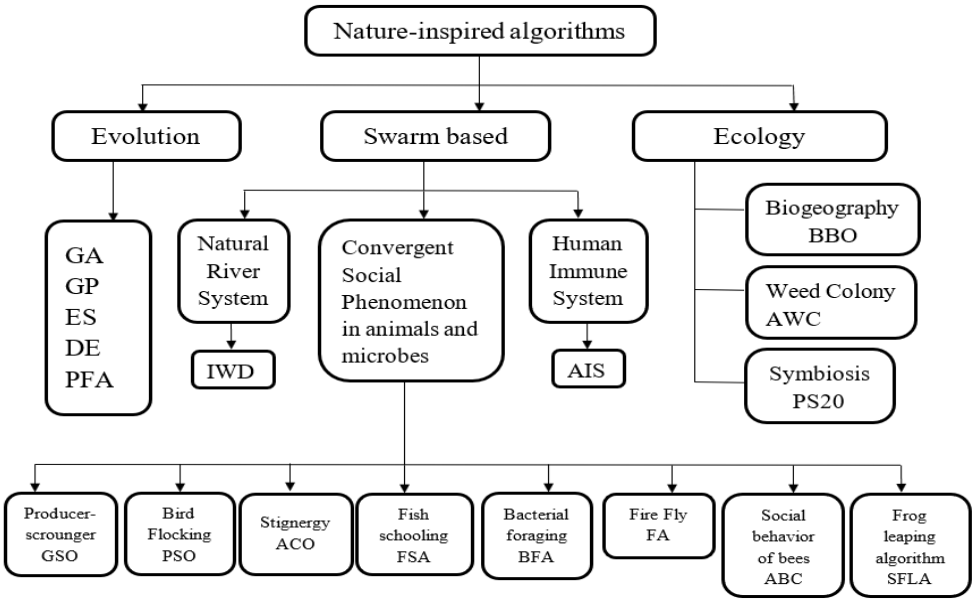


Figure 3.10: nature inspired algorithms [51]

a) GA

Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, in research, and in machine learning. GAs was developed by John Holland and his students and colleagues at the University of Michigan, most notably David E. Goldberg and has since been tried on various optimization problems with a high degree of success. Genetic Algorithms are sufficiently randomized in nature, but they perform much better than random local search (in which we just try various random solutions, keeping track of the best so far), as they exploit historical information as well [75].

Basic Terminology:

- Population: It is a subset of all the possible (encoded) solutions to the given problem. The population for a GA is analogous to the population for human beings except that instead of human beings, we have Candidate Solutions representing human beings.
- Chromosomes: A chromosome is one such solution to the given problem.
- Gene: A gene is one element position of a chromosome.
- Allele: It is the value a gene takes for a particular chromosome.
- Genotype: Genotype is the population in the computation space.
- Phenotype: is the population in the actual real world solution space in which solutions are represented in a way they are represented in real world situations.
- Decoding and Encoding: Decoding is a process of transforming a solution from the genotype to the phenotype space, while encoding is a process of transforming from the phenotype to genotype space.
- Fitness Function: A fitness function simply defined is a function which takes the solution as input and produces the suitability of the solution as the output.
- Genetic Operators: These alter the genetic composition of the offspring. These include crossover, mutation, selection, etc.

Basic Structure:

- We start with an initial population (which may be generated at random or seeded by other heuristics),

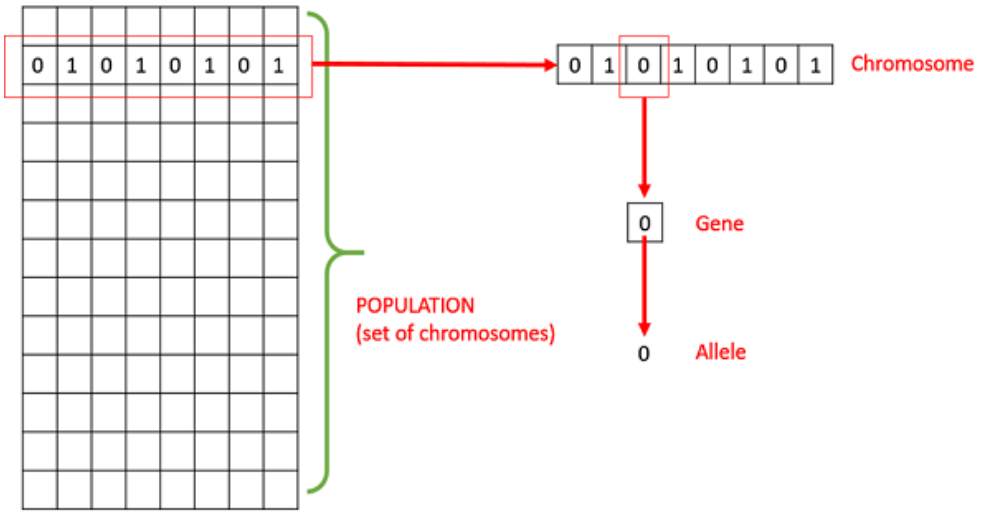


Figure 3.11: Genetic Algorithm basics

- Select parents from this population for mating.
- Apply crossover and mutation operators on the parents to generate new off-springs.
- Finally, these off-springs replace the existing individuals in the population and the process repeats.
- In this way genetic algorithms actually try to mimic the human evolution to some extent.

Advantages of Gas:

- Does not require any derivative information (which may not be available for many real-world problems).
- Is faster and more efficient as compared to the traditional methods.
- Has very good parallel capabilities.
- Optimizes both continuous and discrete functions and also multi-objective problems.
- Provides a list of “good” solutions and not just a single solution.
- Always gets an answer to the problem, which gets better over the time.
- Useful when the search space is very large and there are a large number of parameters involved.

Limitations of Gas:

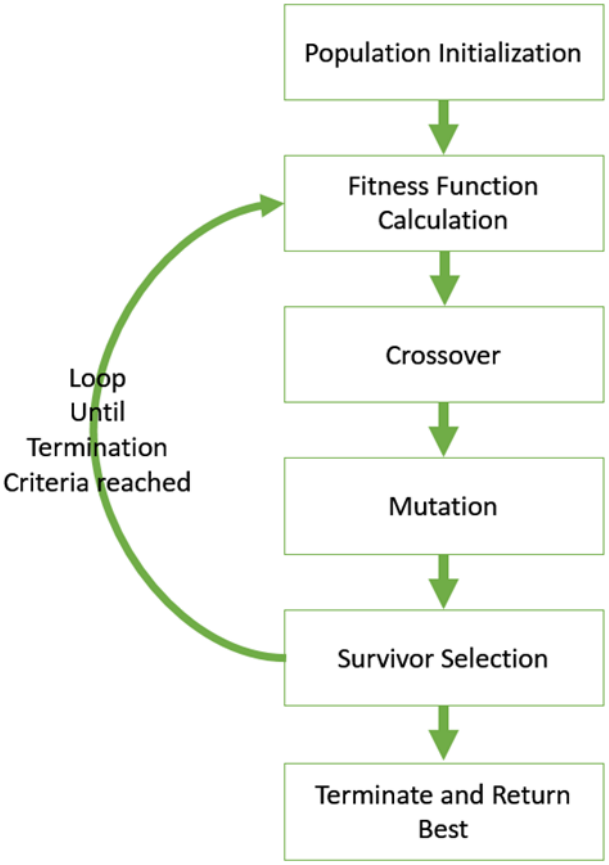


Figure 3.12: Basic Structure of GA

- GAs are not suited for all problems, especially problems which are simple and for which derivative information is available.
- Fitness value is calculated repeatedly which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution.

b) PSO

Particle Swarm Optimization is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 inspired by the social behavior of birds or schools of fish.

PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a pop-

ulation of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. The difference is in the way the generations are updated [74].

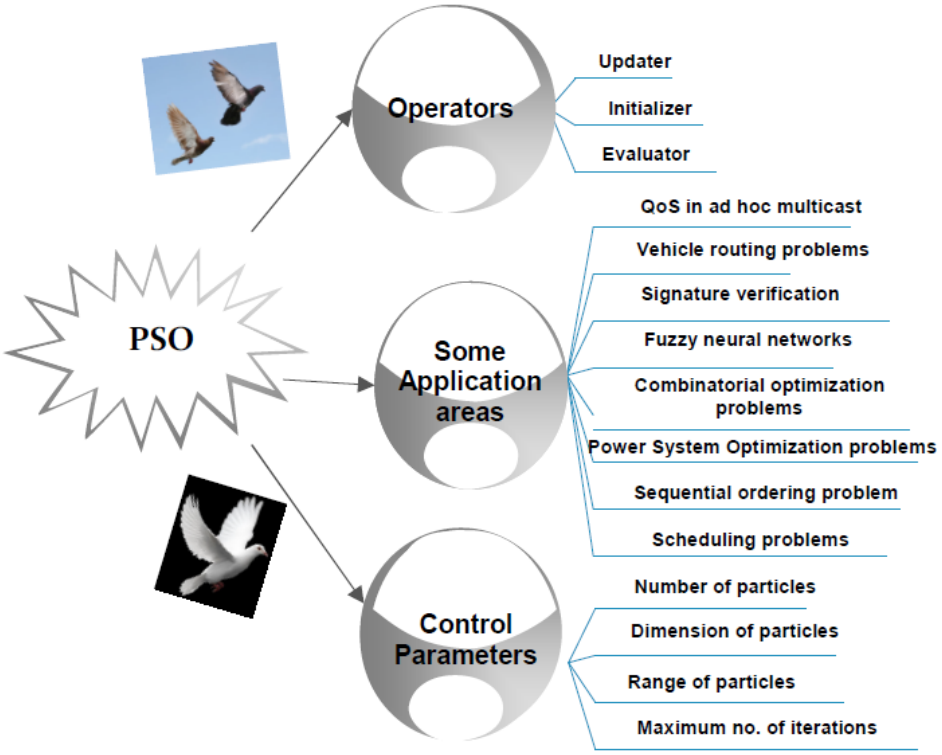


Figure 3.13: Particle Swarm Optimization (PSO) algorithm [55]

PSO makes use of a velocity vector to update the current position of each particle in swarm. The position of each particle is updated based on the social behavior that a population of individuals, the swarm in the case of PSO, adapts to its environment by returning to promising that were previously discovered. The process is stochastic in nature and makes use of the memory of each particle, as well as knowledge gained by the swarm as a whole. The outline of a basic PSO algorithm is follows: [76]

- Start with an initial set of particles, typically randomly distributed through the design space.
- Calculate a velocity vector for each particle in the swarm.
- Update the position of each particle, using its previous position and the updated velocity vector.
- Go to step 2 and repeat until convergence.

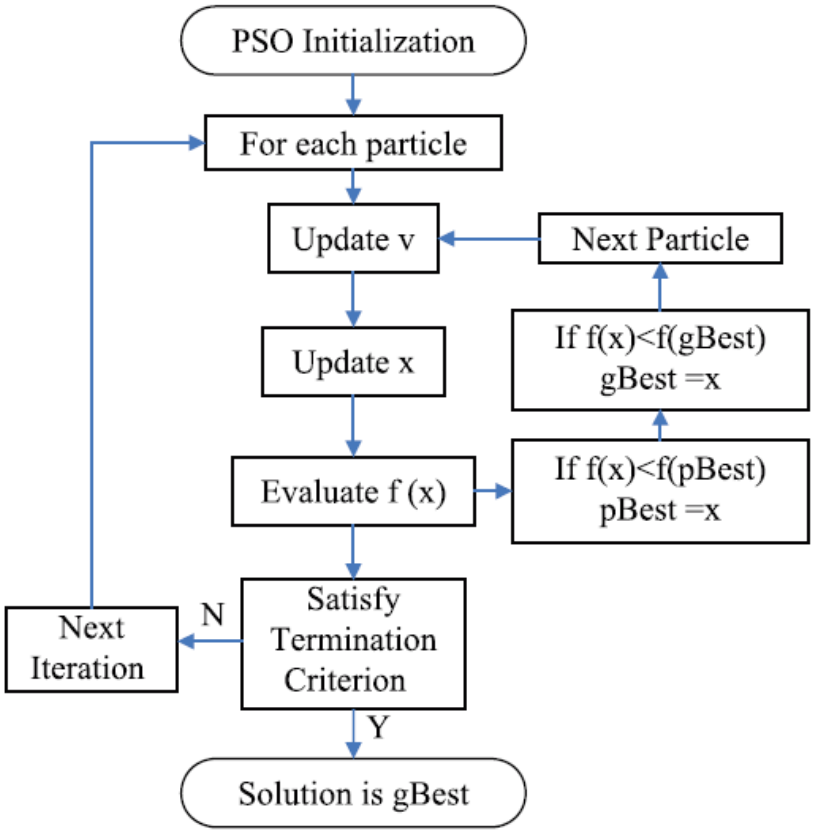


Figure 3.14: Diagram of particle swarm optimization (PSO)[80]

Notions:

- The best solution visited so far in its memory is called pbest
- The best solution visited by any particle and attraction towards this solution is called gbest.
- For an n-dimensional search space, the position and velocity of the ith particle are represented by Yi= (yi1, yi2, . . . , yin) and Vi= (vi1, vi2, vi3 . . . , vin), respectively.
- The previous best position is denoted as

  Pi= (pi1, pi2, . . . , pin). Pg is the global best particle in the swarm.
- For the swarm S, the new velocity of each particle is calculated according to the following equation:

  Vin (t+1) = vin (t)+c1 r1 (pin-yin) +c2 r2 (pg-yin)

  i. The position is updated using: Yin (t+1) = Yin (t)+ Vin (t+1)

  ii. Where i = 1, 2. . . S represents the particle index and n=1, 2. . .

N represents the dimension. c1 and c2 are cognitive and social scaling parameters, respectively.

- At each iteration, pbest and gbest are updated for every particle as per their fitness values.
- The procedure is iteratively repeated until some stop criterion is reached or satisfactory fitness level has been reached. [33]

c) ABC algorithm

Artificial Bee Colony (ABC) algorithm is a swarm-based metaheuristic algorithm for solving combinatorial optimization problems. The intelligent foraging behavior of honey bees is the inspiration for the Artificial Bee Colony Algorithm. This algorithm is specifically based on the model for the foraging behavior of honey bee colonies. The foraging behavior of bees has been adapted as a useful computational algorithm to solve complex problems in different domains [55].

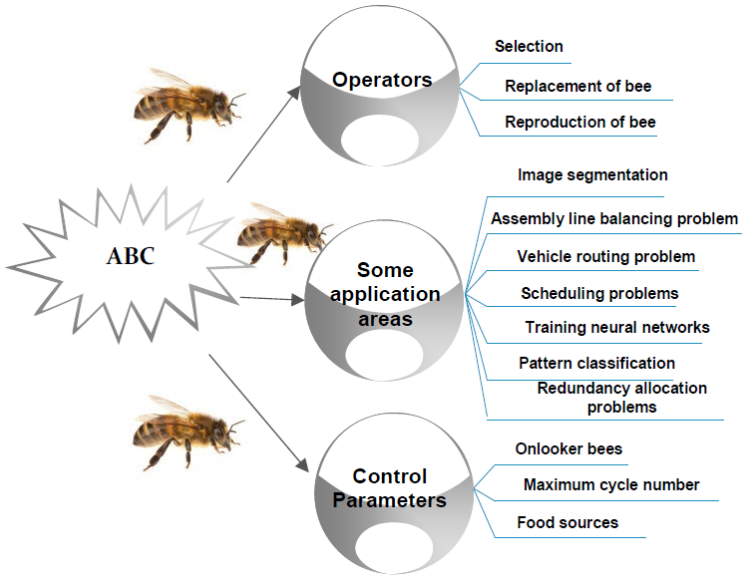


Figure 3.15: Artificial Bee Colony (ABC) algorithm

Artificial Bee Colony (ABC) algorithm is used to solve various optimization problems in different areas. figure 3.15 shows the operators, control parameters and usage areas of the ABC algorithm. Some of the areas include routing problems, scheduling problems, image segmentation and redundancy allocation problems, etc.

Algorithm: Four steps in the artificial bee colony (ABC) algorithm [80].

1. Employed artificial bees to find food sources within the neighborhood of the food sources in their memory.

2. Employed artificial bees pass the messages to onlookers within the hive, and then the onlookers make decision of one of the food sources.
3. Onlookers make a decision of food source within the neighborhood of the food sources chosen by themselves.
4. An employed artificial bee whose source has been rejected restarts to search a new food source randomly.

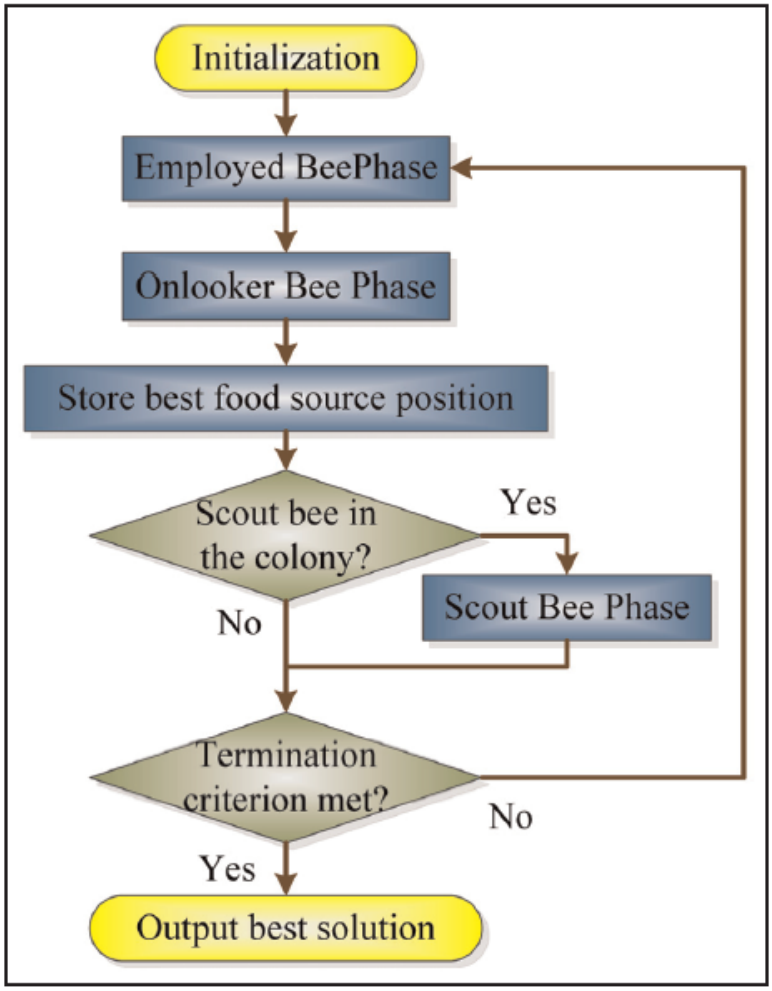


Figure 3.16: The General Flowchart of the ABC Algorithm

d) ACO algorithm

ACO algorithm: is a metaheuristic optimization technique that mimics the behavior of real ants, which try to find the shortest path to the food from their nest. The ants explore randomly the area sur-rounding their nest, and while moving, they leave a chemical pheromone trail on the ground that helps them to go to the nest. Ants interact with each other through this chemical component. The quantity of pheromone is propor-

tional to the quantity and the quality of the food and this pheromone will be guided to other ants for the food source. When evaporation occurs, it reduces the attractive strength of pheromone [33].

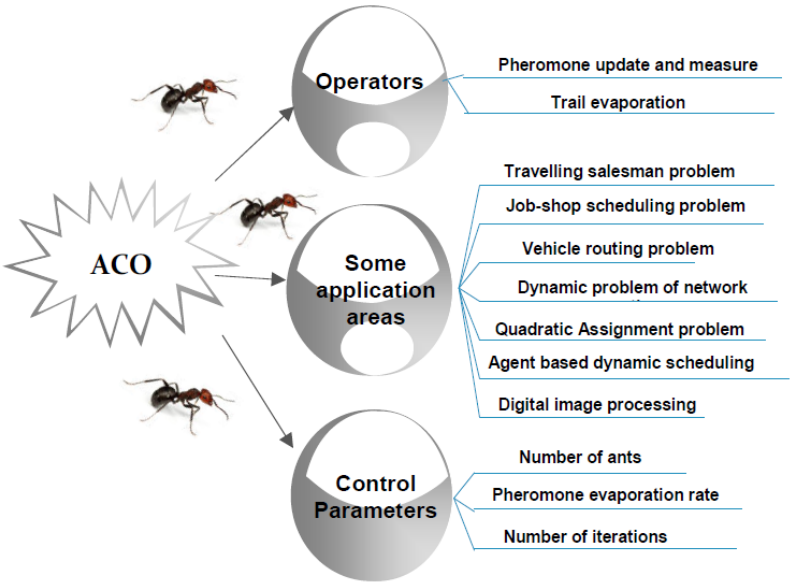


Figure 3.17: Ant Colony Optimization (ACO) algorithm [55]

Algorithm: [20]

- Ant colony algorithms are typically use to solve minimum cost problems.
- We may usually have N nodes and A undirected arcs
- There are two working modes for the ants: either forwards or backwards
- The ant's memory allows them to retrace the path it has followed while searching for the destination node
- Before moving backward on their memorized path, they eliminate any loops from it. While moving backwards, the ants leave pheromones on the arcs they traversed.
- At the beginning of the search process, a constant amount of pheromone is assigned to all arcs. When located at a node i an ant k uses the pheromone trail to compute the probability of choosing j as the next node:

$$\mathrm{P}_{ij}^{k} = \frac{\tau_{ij}^{\alpha}}{\sum_{j \in \mathrm{N}_i^k} \tau_{ij}^{\alpha}} \; if \; j \in \mathrm{N}_i^k$$

$$\mathrm{P}_{ij}^{k} = \; 0 \; if \; j \notin \mathrm{N}_i^k$$

- Where N_{ij}^{k} is the neighborhood of ant k when in node i
- When the arc (i, j) is traversed, the phenomone value changes as follos:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^{k}$$

- By using this rule, the probability increase that forthcoming ants will use this arc.
- After each ant k has moved to the next node, the pheromones evaporate by the following equation to all the arcs:

$$\tau_{ij} \leftarrow (1 - p)\,\tau_{ij},\ \forall\,(i,\ j) \in A$$

ACO System pseudocode: [20]

Often applied to TSP (Travelling Salesman Problem): shortest path between n nodes. Algorithm in pseudocode:

Initialize Trail Do While (stopping criteria not satisfied) – Cycle Loop

Do Until (Each Ant Completes a Tour) – Tour Loop

Local Trail Update

End Do

Analyze Tours Global Trail Update End Do

Steps for solving a problem by ACO: [20]

i. Represent the problem in the form of sets of components and transitions, or by a set of weighted graphs, on which ants can build solutions
ii. Define the meaning of the pheromone trails
iii. Define the heuristic preference for the ant while constructing a solution
iv. If possible, implement an efficient local search algorithm for the problem to be solved.
v. Choose a specific ACO algorithm and apply to problem being solved
vi. Tune the parameter of the ACO algorithm.

e) CS algorithm: [33]

- CS is a new simple heuristic search algorithm that is more efficient than GA and PSO.

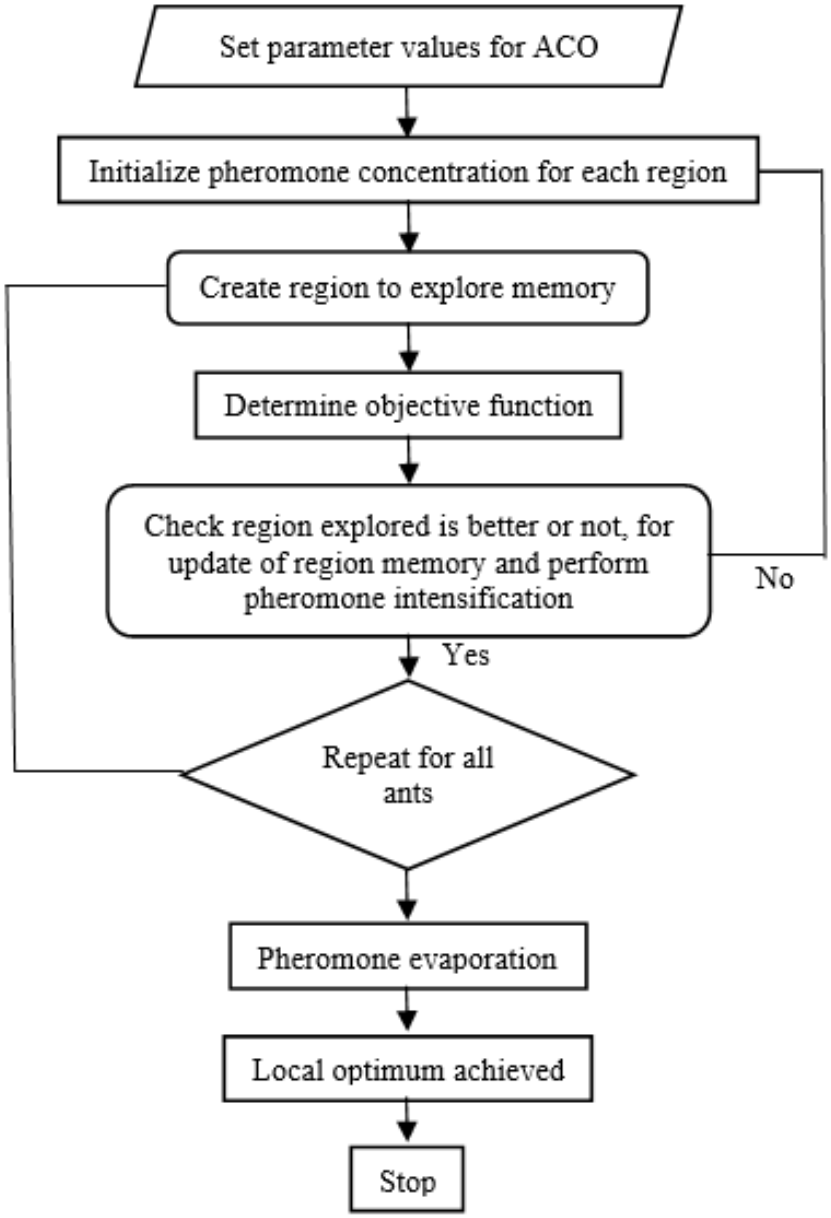


Figure 3.18: Process flow of ant colony optimization [2]

- CS is inspired from brood parasitism reproduction behavior of some cuckoo species in combination with Levy flight behavior.
- The cuckoos lay their eggs in nests of the other birds with the abilities of selecting the lately spawned nests and removing existing eggs to increase the hatching probability of their eggs.
- If host birds discover these eggs, they either throw them away or

abandon the nest and build a new nest.

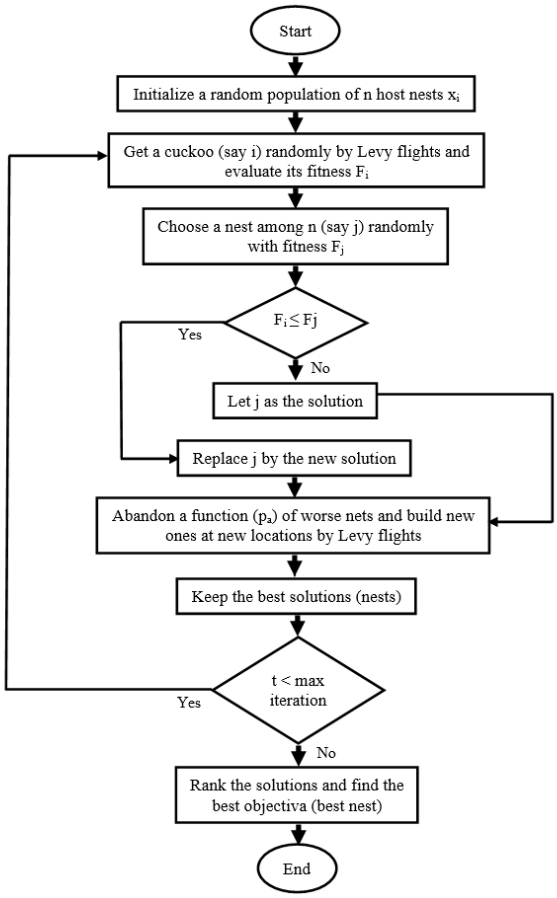


Figure 3.19: Flowchart of CS algorithm[57]

CS is characterized by the subsequent rules:

- Each cuckoo lays one egg at a time and disposes its egg in a random selected nest.
- The nests that have high quality of eggs (Solutions) are the best and will continue to the following generations.
- The number of accessible host nests is fixed, and a host bird can discover a parasitic egg with a probability Pa E [0,1].
- To simulate the behavior of cuckoo reproduction, each egg in a nest is a solution and each cuckoo's egg is a new solution.
- The aim is to supplant a not-so good solution in the nests with newer and better solutions by Levy flights:

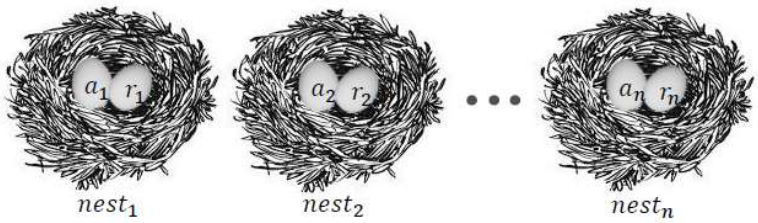
$$y_i^{(t+1)} = y_i^t + \alpha \bigoplus Levey(\lambda)$$

Where $y_i^{(t+1)}$ is a new solution, y_i^t is the current location, α is the step size and Levy (λ) is the transition probability or random walk based on the Levy flights.

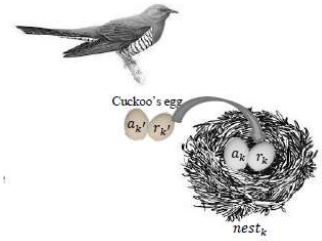
- CS is an effective global optimization algorithm and has many applications in different fields.
- Ebtehal et al applied CS and Modified Adaptive Cuckoo Search (MACS) algorithm on PMP. The MACS algorithm enhances the basic CS algorithm by grouping parallel, incentive, information and adaptive strategies.

The main concepts of Cuckoo search algorithm:

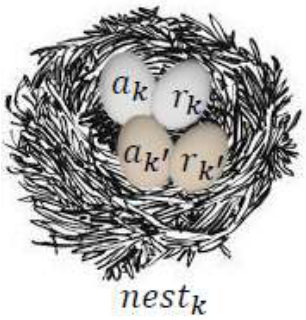
1. Generate initial population of n host nests.
   $(ai,\ ri)$: a candidate for optimal parameters



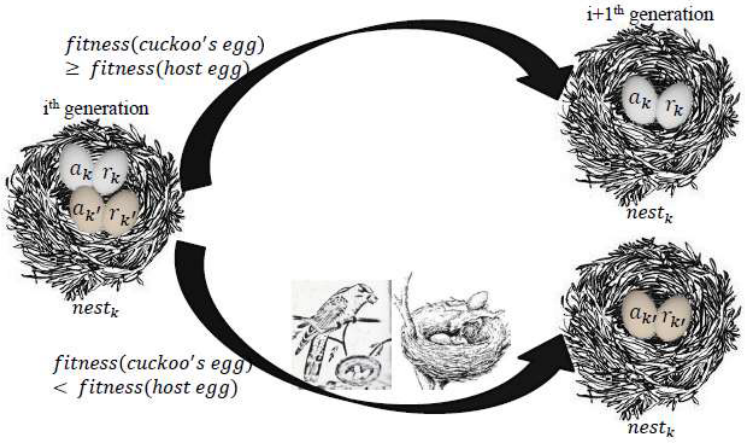
2. Lay the egg (ak', bk') in the k nest.
   - K nest is randomly selected.
   - Cuckoo's egg is very similar to host egg. Where
     $ak' = ak + Randomwalk\,(Levy\ flight)\ ak$
     $rk' = rk + Randomwalk\,(Levy\ flight)\ rk$



3. Compare the fitness of cuckoo's egg with the fitness of the host egg. Root Mean Square Error (RMSE)



4. If the fitness of cuckoo's egg is better than host egg, replace the egg in nest k by cuckoo's egg.



5. If host bird notices it, the nest is abandoned and new one is built. (p <0.25) (to avoid local optimization)

Iterate steps 2 to 5 until termination criterion satisfied [69].

Some of other motif discovery algorithms:

N°	Algorithm	Operating principle	Ref
Enumerative approach			
1	YMF (Yeast Motif Finder)		[68]
2	MITRA (MIsmatch TRee Algorithm)		[21]
3	RecMotif	Graph-theoretic	[71]
4	ListMotif		[70]
5	TreeMotif		[72]
Probabilistic approach			
6	MEME		[6]
7	STEME		[58]
8	MITSU		[39]
Genetic Algorithm (GA)			
9	GAMI		[18]
10	GAEM		[81]
11	MRPGA		[82]
PSO			
12	PMbPSO		[93]
13	LPBS	Standard PSO	[1]
Artificial bee colony (ABC) algorithm			
14	Multiobjective ABC		[31]
15	MO-ABC/DE	ABC	[30]
Ant colony optimization (ACO) algorithm			
16	MFACO	ACO with Gibbs sampling	[10]
CS algorithm			
17	MACS	CS	[24]
Combinatorial			
18	MUSA	Probabilstic and machine learning approaches	[43]
19	EMD	Multiple algorithms	[34]

Table 3.2: Some of motif discovery algorithms [33]

3.1.6 Motif discovery tools:

search for motifs in a set of unaligned sequences is a complex problem because many factors come into play, such as the precise start and end boundaries of the motif, the size variability (presence of gaps or not), or stronger or weaker motif conservation during evolution. in the following we will present of the programs that are specifically designed to search for motifs in protein sequences that are biologically significant.

1. MEME:
    - MEME is an example of a deterministic optimization algorithm.
    - It allows discovery of motifs in DNA or protein sequences based on expectation maximization (EM).
    - MEME discovers at least three motifs, each of which may be present in some or all of the input sequences.
    - MEME chooses the width and number of occurrences of each motif automatically in order to minimize the “E-value” of the motif, i.e., the probability of finding a similarly well conserved pattern in random sequences.
    - With default parameters, only motif widths between 6 and 50 are considered, but the user has the possibility to change this as well as several other parameters (options) of the motif discovery.
2. Pratt:
    - Pratt is based on probabilistic optimization.
    - It first searches the space of motifs, as constrained by the user, and compiles a list of the most significant sequences that matches at least the user-defined minimum number of sequences.
    - If the user has not switched off the refinement, these motifs will be input to one of the motif refinement algorithms.
    - The most significant motifs resulting from this are then output to a file.
3. qPMS:
    - qPMS stands for quorum planted motif search.
    - The program searches for motifs in either DNA or protein sequences.
    - It uses the (l, d) motif search algorithm known as the planted motif search.
    - qPMS takes as input a set of sequences and two values, l and d.
    - It returns all sequences M of length l, which appear in at least q

4. SLiMFinder:
    - SLiMFinder identifies novel short linear motifs (SLiMs) in a set of sequences.
    - SLiMs are micro-domains that have important functions in many diverse biological pathways.
    - SLiMmediated functions include post-translational modification, subcellular localization, and ligand binding.
    - SLiMs are generally less than 10 amino acids long, many of which will be “flexible”
    - in terms of the conserved amino acid, SLiMFinder constructs such motifs by grouping dimers into longer patterns:
        - motifs with fixed amino acid positions are identified and then grouped to include amino acid ambiguity and variable length wildcards.
        - Finally, motifs that are over represented in a set of unrelated proteins are identified.
5. Dilimot:
    - Dilimot proceeds as follows:
        - in the first step, a user provided set of protein sequences is filtered to eliminate repetitive sequences as well as the regions least likely to contain linear motifs.
        - In the second step, over represented motifs are identified in the non-filtered sequences and ranked according to scores that take into account the background probability of the motif, the number of sequences containing the motif, the size of the sequence set, and the degree to which the motif is conserved in other orthologous proteins.
6. Motif Hound:
    - Motif Hound is suitable for the discovery of small and degenerate linear motifs.
    - The method needs two input datasets:
        - a background set of protein sequences and a subset of this background set that represents the query sequences.
        - Motif Hound first enumerates all possible motifs present in the query sequences, and then calculates the frequency of each motif in both the query and the background sets.

7. FIRE pro:
    - FIRE pro stands for finding informative regulatory elements in proteins.
    - Its main goal is to discover protein motifs that correlate with the biological behavior of the corresponding proteins.
    - FIRE pro calculates a mutual information measure between frequent k-mer motifs and a “protein behavior profile” containing experimental data about the function of the proteins.

Pattern Recognition - Analysis and Applications Most of these programs need prior knowledge about either the input sequences or the motif structure. Furthermore, they are generally designed to discover frequent motifs that occur in all or most of the sequences [61].

Program	Description	Advantages	Disadvantages
Teiresias	Finds motifs that are frequent in a set of related sequences	Does not need background sequences; Very fast	Too many redundant motifs discovered
MEME	Finds motifs in related sequences using Gibbs sampling and expectation maximization	Does not need background sequences; Fast, Multi-thread version available; User friendly output	User defines the number of motifs to discover
Pratt	Discovers flexible motifs in related sequences	Does not need background Sequences	Unable to discover effectively exact motifs
qPMS	Finds over represented motifs in a set of sequences based on Quorum Planted Motif Search	Fast; Low memory consumption	Limited to 20 proteins sequences
SlimFinder	Finds over represented motifs in a set of unrelated sequences relative to background sequences	Well documented; Can use filters	Needs background sequences
MotifHound	Exhaustively finds motifs over represented in a set of unrelated sequences relative to background sequences	Exhaustive exploration of motifs; Can use filters Fast; Multi-thread version available	Needs background sequences

Dilimot	Finds over represented motifs in a set of unrelated sequences relative to a background sequences	Integrates several types of sequence information on motifs	Needs background sequences; Source code not available
FirePro	Correlates over represented motifs in a set of sequences with specific functions or behaviors	User friendly output	Needs background sequences

Table 3.3: Advantages and limitations of the most used motif discovery programs.[61]

3.1.7 Benefits of motifs discovery:

discovery of information encoded in biological sequences is assuming a distinguished role in identifying genetic diseases and in deciphering biological mechanisms.

1. This information is usually encoded in patterns frequently occurring in the sequences.
2. Remove repeating Motif discovery is the critical step to understand the regulatory mechanism of genes.
3. The motifs can represent patterns which activate or inhibit the transcription process and are responsible for regulating gene expression.
4. In Bioinformatics, motif discovery is becoming very important because they represent conserved sequences which can be biologically meaningful.
5. It could be essential to the analysis and understanding of the biological data.
6. If a pattern occurs frequently, it ought to be important or meaningful in some way.
7. Motifs are recurring patterns in biodata that are presumed to have a biological function.
8. Often, they indicate sequence specific binding sites for proteins such as nucleases and TFs.
9. The discovery of patterns in DNA, RNA, and protein sequences has led to the solution of many vital biological problems.
10. Motif discovery for protein sequences is important for identifying structurally or functionally important regions and understanding proteins functional components, or active sites [53].

3.1.8 Limitations of motif discovery:

Awareness of the limitations of motif discovery can guide you to more success. In this section we're going to mention some motif discovery limitations.

1. Some limitations have to do with the difficulty of discovering weak motifs in the face of noise.
2. Spurious motifs are another source of difficulty.
3. You can often think of motif discovery as a "needle-in-a-haystack" problem where the motif is the "needle" and the sequences in which it is embedded is the "haystack."
4. Many DNA motifs (e.g., TFBSs) tend to have low levels of similarity among occurrences, so it is especially important to limit sequence length and the number of "noise" sequences (ones not containing occurrences) in the input sequence set.
5. Over-representation depends inversely on the length of the sequences, so it is always good to limit the length of the input sequences as much as possible [38].

3.1.9 Conclusion:

Traditional gene sequencing methods and exact motif recognition algorithms are generally expensive and time consuming. Swarm-based intelligence computation algorithms have been proposed to solve motif recognition problems.

However, they usually have difficulties in achieving satisfactory results, especially when the objective search space possesses much noise subsequences [26].

The field of motif discovery brings together researchers from several disciplines, in particular from biology, statistics and informatics. Additionally, research in the field is fairly recent and moving at a fast pace. This has resulted in a broad range of computational methods that are described with different vocabulary and different focus, making it difficult to spot similarities as well as differences between methods [63].

The motif discovery algorithms are classified into four classes of enumerative, probability, nature in-spired and combinatorial ones and each one has many subclasses. The enumerative technique is an exhaustive search with a simple concept, and it is the only technique that ensures to find all motifs (Except weak motifs). However, it is very slow, and requires a lot of parameters; as a result, it becomes difficult to deal with either long motifs or big data. Moreover, the degenerative positions are limited because of restricted representation of motifs [33].

3.2 Prosite:

databases have been around for the best part of half a century. One of the very first protein databases was the Atlas of Protein Sequence and Structure developed by the late Margaret Dayhoff who founded the PIR (Protein Information Resource).

A series of books were published from 1965 to 1978 until the quantity of data grew so much that an electronic form was made available to the scientific community, known as the PIR-International Protein Sequence Database. Swiss-Prot, the protein sequence knowledge base founded in 1986 by Amos Bairoch, took its inspiration from PIR but strove to develop a database that was non redundant and extremely well documented. a great many diverse databases have sprouted, one of them is the ProSite database, which we will talk about in the following [28].

3.2.1 What is Prosite?

is an annotated collection of motif descriptors dedicated to the identification of protein families and domains [67].

The core of the PROSITE database is composed of two text files:

- PROSITE.DAT: is a computer readable file that contains all the information necessary to programs that make use of PROSITE to scan sequence(s) for the occurrence of patterns or profiles. This file includes, for each of the entry described, statistics on the number of hits obtained while scanning the SWISS-PROT protein database for a pattern or profile. Cross-references to the corresponding SWISS-PROT entries as well as to matched sequences from the PDB 3D-structure database are also provided.
- PROSITE.DOC: contains textual information that fully documents each pattern or profile. Release 17.18 of PROSITE (August 4, 2002) contains 1147 documentation entries that describe 1567 different motif descriptors. In addition to these entries, a collection of 152 pre-release profiles (see below) is also available [67].

PROSITE is a method of determining what is the function of uncharacterized proteins translated from genomic or cDNA sequences. It consists of a database of biologically significant sites and patterns formulated in such a way that with appropriate computational tools it can rapidly and reliably identify which known family of protein (if any) the new sequence belongs to [52].

PROSITE is a protein database. It consists of entries describing the protein families, domains and functional sites as well as amino acid patterns and profiles in them. These are manually curated by a team of the Swiss Institute of Bioinformatics and tightly integrated into Swiss-Prot protein annotation [87].

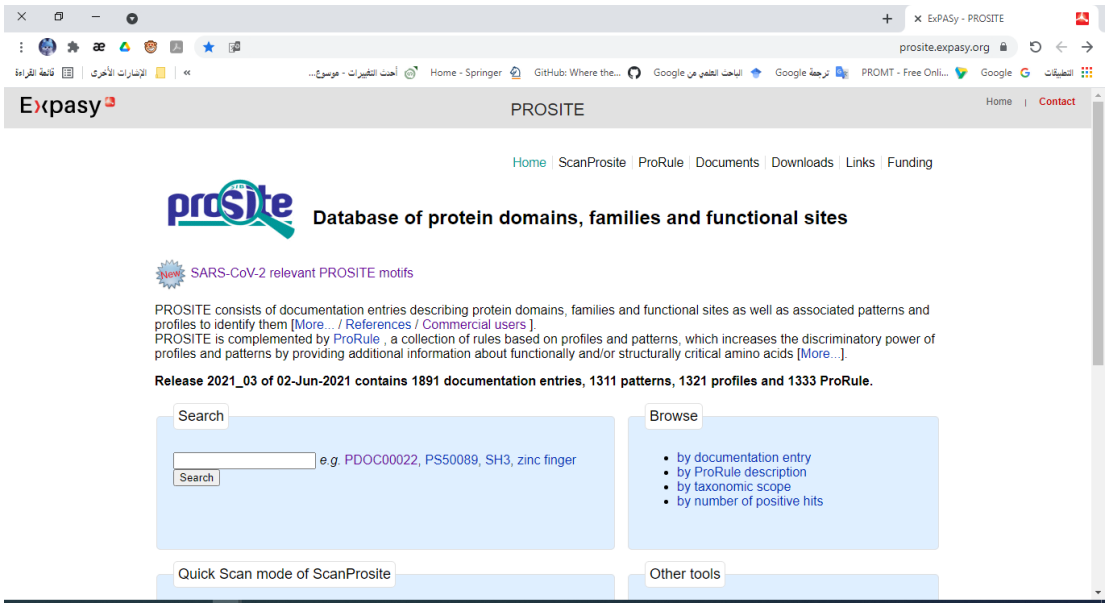


Figure 3.20: PROSITE web page

3.2.2 History:

PROSITE was created in 1989 by Amos Bairoch, who directed the group for more than 20 years Since July 2009, the director of the PROSITE, swiss-Prot and Vital-IT groups is loannis Xenarios Since July 2018, the director of PROSITE and Swiss-Prot is Alan Bridge [9][87].

3.2.3 Prosite methodology:

PROSITE Patterns:

Patterns are qualitative descriptors: either they match or they do not. If there is a mismatch at one of the positions the pattern will not match, even if the mismatch is a conservative, biologically feasible substitution.

These are segments of amino acids sequence arranged in a structure which act as a signature of a particular protein family.

These biologically significant regions or residues are generally:

- Enzyme catalytic sites.
- Prostethic group attachment sites.
- Amino acids involved in binding a metal ion.
- Cysteines involved in disulphide bonds.

- Regions involved in binding a molecule (ADP / ATP, GDP / GTP, Calcium, DNA, etc...) or another protein [9].

PROSITE Profiles:

Profile are quantitative motifs descriptors that consider the overall similarity on the entire length of domains or proteins and not just the most conserved parts of them.

A mismatch at a highly conserved position can thus be accepted provided that the rest of the sequence displays a sufficiently high level of similarity.

The enhanced sensitivity of generalized profiles allows the detection of poorly conserved domains or families.

An advantage of profile over pattern is that they characterize protein domain over their entire length, not just the most conserved part of it.

Despite their obvious advantages. profiles are not superior to pattern for all purposes. In fact, the two types of descriptors have complementary qualities. Profiles covering complete domain are more suitable for predicting protein structural properties.

Whereas Patterns confined to small regions with high sequence similarity are often powerful predictors of protein functions such as enzymatic activities [9].

Repeats identification:

Generally, repeats possess high amino acid substitution rates and their identification is highly problematic. Even if the presence of a certain repeat family is known, the exact locations and number of repetitive units often cannot be determined using current profile search. We have implemented a context dependent threshold that allows the detection of strongly divergent repeats when well characterized ones have already been identified.

Our approach aims to set a lower acceptance threshold for sub-optimal alignments of profiles to proteins containing repeats. This is accomplished by scanning the profile against a randomized database of sequences where the occurrence of at least one copy of the repeat has been assessed with high confidence. The computed lower acceptance threshold is then used both for the detection of additional copies of the same repeat within the protein, and for the identification of new distantly related members of the protein family.

Two complementary approaches were designed to increase the sensitivity of profiles for the detection of repeats. One approach, RDM1 (Repeats Detection Method 1) consists in defining (computing) a low acceptance threshold placed at level -1 in the profile. For simplicity we will call level 0 cutoff protein-threshold

and level -1 cutoff minimal-threshold. When the profile is compared with a given sequence a list of matches with scores greater than the minimal-threshold is collected. The matches are considered as significant, only if at least a hit with a score greater than the protein-threshold has been detected in the target protein. In a target sequence, where the occurrence of a particular domain has been reported, the minimal-threshold represents the score above which the probability of detecting additional copies of the same domain by chance is close to zero.

However, the detection of repeats in proteins where no single domain scores above the protein-threshold remains critical. This is typically the case for more distantly related members of a protein family. To obviate this problem a second approach was devised, Repeats Detection Method 2 (RDM2). The sum of the scores of alignments with scores greater than the minimal-threshold is computed. If the sum of the individual domain scores is larger than a threshold (the sum-of-scores-threshold), these domains are considered to be true homologues. Based on the inspection of the list of positive hits found upon databases searches, we found that a good estimate for the sum-of-scores-threshold is the value of the sum of the protein-threshold with the minimal-threshold. This value was chosen since it represents in theory the minimal match score that would be detected when aligning a profile to a member of a given protein family containing only two copies of a repeat.

RDM1 and RDM2 were implemented in the ps_scan PROSITE scanning program, the standalone version of Scan Prosite (6). ps_scan allows to scan a protein sequence (either from UniProtKB/Swiss-Prot or UniProtKB/TrEMBL or provided by the user) for the occurrence of patterns and profiles stored in the PROSITE database. The modified ps_scan program applies as default RDM1 and/or RDM2 when run with profiles for repetitive domains. Profiles for repetitive domains are tagged with ’R’ and ’RR’ or ’R?’ in the TEXT field of the CUT_OFF lines (LEVEL=0 and LEVEL=-1) of the profile. When the profile is tagged with ’RR’ the two methods RDM1 and RDM2 are applied, whereas when it is tagged with ’R?’ only RDM1 is applied. In the output of the program the reported matches are tagged with ’R’ or with ’r’ when the hits have been detected with RDM1 or RDM2 respectively.

Example:

MA /CUT_OFF: LEVEL=0; SCORE=246; N_SCORE=8.5; MODE=1; TEXT=’R’;

MA /CUT_OFF: LEVEL=-1; SCORE=158; N_SCORE=5.8; MODE=1; TEXT=’RR’;

or

MA /CUT_OFF: LEVEL=0; SCORE=246; N_SCORE=8.5; MODE=1; TEXT=’R’;

MA /CUT_OFF: LEVEL=-1; SCORE=158; N_SCORE=5.8; MODE=1; TEXT=’R?’;

[52]

Database conventions:

1. General structure:
   The PROSITE database is composed of two ASCII (text) files.

   PROSITE.DAT is a computer readable file that contains all the information necessary to programs that make use of PROSITE to scan sequence(s) for the occurrence of patterns or profiles with patterns and/or matrices. This file includes, for each of the entry described, statistics on the number of hits obtained while scanning the SWISS-PROT protein database for a pattern or profile. Cross-references to the corresponding SWISS-PROT entries as well as to matched sequences from the PDB 3D-structure database are also provided. The second file PROSITE.DOC contains textual information that fully documents each pattern or profile. Release 17.18 of PROSITE (August 4, 2002) contains 1147 documentation entries that describe 1567 different motif descriptors. In addition to these entries, a collection of 152 pre-release profiles (see below) is also available [67]. We must point out that we strongly urge software developers to build software tools that make use of both files. A list of patterns or profiles present in a sequence is not very useful to biologists without the relevant documentation [52].

Data file structure:

1. Structure of an entry:
   The entries in the database data file (PROSITE.DAT) are structured so as to be usable by human readers as well as by computer programs. Each entry in the database is composed of lines. Different types of lines, each with its own format, are used to record the various types of data which make up the entry. The general structure of a line is the following:

Characters	Content
1 to 2	Two-character line code. Indicates the type of information contained in the line
3 to 5	Blank
6 up to 128	Data

   The currently used line types, along with their respective line codes, are listed below:

ID	Identification (Begins each entry; 1 per entry)
AC	Accession number (1 per entry)
DT	Date (1 per entry)
DE	Short description (1 per entry)
PA	Pattern (>=0 per entry)
MA	Matrix/profile (>=0 per entry)
PP	Post-processing (>=0 per entry)
NR	Numerical results (>=0 per entry))

CC Comments (>=0 per entry)
DR Cross-references to UniProtKB/Swiss-Prot (>=0 per entry)
3D Cross-references to PDB (>=0 per entry)
PR Reference to associated ProRule (>=0 per entry)
DO Reference to the documentation file (1 per entry)
// Termination line (Ends each entry; 1 per entry) Lines do not extend over 78 characters, with the exception of "MA" lines whose length has no limit.

2. Example of a pattern entry:

```
ID   CUTINASE_1; PATTERN.
AC   PS00155;
DT   APR-1990 (CREATED); NOV-1997 (DATA UPDATE); MAR-2005 (INFO UPDATE).
DE   Cutinase, serine active site.
PA   P-x-[STA]-x-[LIV]-[IVT]-x-[GS]-G-Y-S-[QL]-G.
NR   /RELEASE=46.4,178022;
NR   /TOTAL=20(20); /POSITIVE=20(20); /UNKNOWN=0(0); /FALSE_POS=0(0);
NR   /FALSE_NEG=0; /PARTIAL=0;
CC   /TAXO-RANGE=??EP?; /MAX-REPEAT=1;
CC   /SITE=11, active_site;
DR   P63880, CUT1_MYCBO, T; P63879, CUT1_MYCTU, T; P63882, CUT2_MYCBO, T;
DR   P63881, CUT2_MYCTU, T; P0A537, CUT3_MYCBO, T; P0A536, CUT3_MYCTU, T;
DR   P00590, CUTI1_FUSSO, T; Q96UT0, CUTI2_FUSSO, T; Q96US9, CUTI3_FUSSO, T;
DR   P41744, CUTI_ALTBR, T; P29292, CUTI_ASCRA, T; P52956, CUTI_ASPOR, T;
DR   Q00298, CUTI_BOTCI, T; P10951, CUTI_COLCA, T; P11373, CUTI_COLGL, T;
DR   Q8X1P1, CUTI_ERYGR, T; Q99174, CUTI_FUSSC, T; P30272, CUTI_MAGGR, T;
DR   Q8TGB8, CUTI_MONFR, T; Q9Y7G8, CUTI_PYRBR, T;
3D   1AGY; 1CEX; 1CUA; 1CUB; 1CUC; 1CUD; 1CUE; 1CUF; 1CUG; 1CUH; 1CUS; 1CUU;
3D   1CUV; 1CUW; 1CUY; 1CUZ; 1FFA; 1FFB; 1FFC; 1FFD; 1FFE; 1OXM; 1XZA; 1XZB;
3D   1XZC; 1XZD; 1XZE; 1XZF; 1XZG; 1XZH; 1XZJ; 1XZK; 1XZL; 1XZM; 2CUT;
DO   PDOC00140;
```

3. Example of a profile (matrix) entry:

```
ID HSP20; MATRIX.
AC PS01031;
DT JUN-1994 (CREATED); DEC-2001 (DATA UPDATE); MAR-2005 (INFO UPDATE).
DE Heat shock hsp20 proteins family profile.
MA /GENERAL_SPEC: ALPHABET='ABCDEFGHIKLMNPQRSTVWYZ'; LENGTH=88;
MA /DISJOINT: DEFINITION=PROTECT; N1=6; N2=83;
MA /NORMALIZATION: MODE=1; FUNCTION=LINEAR; R1=-0.7971325; R2=0.0157729; TEXT='-LogE';
MA /CUT_OFF: LEVEL=0; SCORE=590; N_SCORE=8.5; MODE=1; TEXT='!';
MA /CUT_OFF: LEVEL=-1; SCORE=463; N_SCORE=6.5; MODE=1; TEXT='?';
MA /DEFAULT: M0=-8; D=-20; I=-20; B1=-50; E1=-50; MI=-105; MD=-105; IM=-105; DM=-105;
MA /I: B1=0; BI=-105; BD=-105;
MA /M: SY='D'; M=-10,26,-29,38,34,-34,-14,-2,-33,7,-24,-23,8,-6,8,-4,0,-9,-27,-33,-19,21;
MA /M: SY='I'; M=-8,-31,-23,-35,-28,7,-32,-27,27,-24,15,13,-27,-26,-24,-23,-20,-9,25,-4,2,-27;
```

MA /M: SY='R'; M=-11,-12,-26,-12,-1,-13,-23,-1,-8,1,-7,-3,-8,-11,-2,8,-9,-6,-8,-22,-3,-4;

MA /M: SY='E'; M=-11,17,-27,23,29,-24,-15,-3,-27,1,-22,-20,9,-1,6,-6,3,-4,-25,-32,-17,17;
MA /M: SY='D'; M=-7,10,-23,11,2,-25,0,-6,-26,-4,-23,-18,7,-6,-5,-8,7,7,-20,-31,-17,-2;
MA /I: I=-4; MD=-22;
MA /M: SY='D'; M=-8,17,-27,25,19,-30,-13,-5,-28,6,-25,-20,7,3,4,-1,0,-7,-24,-30,-19,10; D=-4;
MA /M: SY='D'; MA /I: I=-4; MI=0; MD=-22; IM=0; DM=-22;
M=-11,20,-25,24,16,-29,-12,-1,-27,14,-25,-16,14,-9,10,5,1,-6,-23,-28,-14,13; D=-4;
MA /I: I=-4; DM=-22;
... Some lines omitted..
MA /M: SY='K'; M=-9,-5,-25,-6,0,-22,-21,-12,-17,30,-21,-6,-3,-16,1,23,-9,-7,-6,-23,-11,0;
MA /I: E1=0; IE=-105; DE=-105;
NR /RELEASE=46.4,178022;
NR /TOTAL=195(194); /POSITIVE=190(189); /UNKNOWN=5(5); /FALSE_POS=0(0);
NR /FALSE_NEG=1; /PARTIAL=8;
CC /MATRIX_TYPE=protein_domain;
CC /SCALING_DB=reversed;
CC /AUTHOR=P_Bucher;
CC /TAXO-RANGE=A?EP?; /MAX-REPEAT=2;
CC /FT_KEY=DOMAIN; /FT_DESC=HSP20;
DR P0A5B8, 14KD_MYCBO , T; P0A5B7, 14KD_MYCTU , T; P46729, 18K1_MYCAV , T;
DR P46730, 18K1_MYCIT , T; P46731, 18K2_MYCAV , T; P46732, 18K2_MYCIT , T;
DR P12809, 18KD_MYCLE , T; P80485, ASP1_STRTR , T; O30851, ASP2_STRTR , T;
... Some lines omitted..
DR P12812, P40_SCHMA , T; Q06823, SP21_STIAU , T; O34321, YOCM_BACSU , T;
DR O12987, CRYAB_COLLI, P; O12991, CRYAB_EUDEL, P; Q91518, CRYAB_TRASC, P;
DR O12995, CRYAB_TURME, P; P81161, HS22M_LYCES, P; P30220, HS30E_XENLA, P;
DR P81083, HSP11_PINPS, P; Q9QUK5, HSPB7_RAT , P;
DR P22979, HSP6C_DROME, N;
DR Q29438, ODFP_BOVIN , ?; Q14990, ODFP_HUMAN , ?; Q61999, ODFP_MOUSE , ?;
DR Q29077, ODFP_PIG , ?; P21769, ODFP_RAT , ?;
3D 1SHS;
DO PDOC00791;
[52].

How to make use of prosite:

1. Computer programs:
   We provide programs that have been specifically developed to help use PROSITE for both patterns and profiles searches:

   - ps_scan, a program used to scan one or several PROSITE motifs against one or several protein sequences. ps_scan is available from https://ftp.expasy.org/databases/prosite/ps_scan/.

- PFTOOLS, programs used to construct profiles or scan a sequence or a sequence library against a profile or a profile library. PFTOOLS are available from:https://github.com/sib-swiss/pftools3.

2. Interactive Web access to PROSITE:
   To browse the PROSITE documentation and motif entries, users should go to http://www.expasy.org/prosite/. Web access to PROSITE allows users to benefit from the latest PROSITE updates and from hyperlinks connecting a PROSITE entry to other relevant sources of information. In addition, it has recently been made possible for the user to display the match list of a PROSITE motif as a multiple alignment available in different formats. To scan a sequence for PROSITE motifs, one can make use of the following tools:

   - ScanProsite:
     ScanProsite allows either to scan a protein sequence – from SWISS-PROT or provided by the user – for the occurrence of PROSITE motifs or to scan the SWISS-PROT, TrEMBL and/or PDB databases for the occurrence of a pattern that can originate from PROSITE or be provided by the user. ScanProsite also allows the user to visualize the position of a PROSITE motif or of his own pattern on the 3D structure (if known) of the matched proteins. Recently, we added the possibility for the user to evaluate the specificity of a pattern by using it to scan a randomized version of the current SWISS-PROT database. The URL for ScanProsite is http://www.expasy.org/tools/scanprosite .

   - ProfileScan:
     ProfileScan allows a protein sequence – from SWISS-PROT or provided by the user – to be scanned for the occurrence of profiles stored in PROSITE and in the pre-release collection. The new URL for ProfileScan is http://hits.isb-sib.ch/cgi-bin/PFSCAN [67].

4

OUR CONTRIBUTION

4.1 Introduction:

Motif discovery is one of the well-known studies in Bioinformatics. Many tools have been developed for motif discovery. Recent motif finding tools facilitate the motif detection by providing user-friendly Web interface. In this work, we reviewed four motif discovery Web tools that are capable for detecting motifs.

4.2 Brief introduction of online analysis tools:

4.2.1 ScanProsite:

It is a tool of search for motifs and patterns within protein sequences.

The ScanProsite tool allows to scan protein sequences for the occurrence of patterns, profiles and rules (motifs) stored in the PROSITE database, or to search protein database(s) for hits by specific motif(s)[73]. ScanProsite allows to scan proteins for matches against the PROSITE collection of motifs as well as against user-defined patterns [22].

4.2.2 ScanProsite:

Pfam is a database of protein families that includes their annotations and multiple sequence alignments generated using hidden Markov models. The most recent version, Pfam 33.1, was released in May 2020 and contains 18,259 families [86].

The general purpose of the Pfam database is to provide a complete and accurate classification of protein families and domains. Originally, the rationale behind creating the database was to have a semi-automated method of curating information on known protein families to improve the efficiency of annotating genomes. The Pfam classification of protein families has been widely adopted by biologists because of its wide coverage of proteins and sensible naming conventions.

The Pfam website allows users to submit protein or DNA sequences to search for matches to families in the database. If DNA is submitted, a six-frame translation is performed, then each frame is searched. Rather than performing a typical BLAST search, Pfam uses profile hidden Markov models, which give greater weight to matches at conserved sites, allowing better remote homology detection, making them more suitable for annotating genomes of organisms with no well-annotated close relatives.

4.2.3 Motif Scan:

The Motif Scan tool is a MyHits tool developed by the Swiss Institute of Bioinformatics (SIB). The tool uses databases from HAMAP, PROSITE, and Pfam to extract motifs similar to the one queried. The purpose of this tool is to identify the

motifs or pattern found in a protein sequence. Determining the motifs of a protein will assist in the classification of a protein according to its family or domain [50].

4.2.4 MOTIF:

Motif is a tool introduced by GenomeNet, the tool uses databases from PROSITE, NCBI-CDD and Pfam. it allows to:[27]

1. Search with a protein query sequence against Motif Libraries.
2. Align a protein sequence with a profile library given by a user. (PROSITE or HMMER format).
3. Search with a profile against protein sequence databases.
4. Search a protein sequence pattern (regular expression) against sequence databases.
5. Generate a profile from a set of multiple aligned sequences.

4.3 Amino acids codes:

Amino acids are coded in three letters while in the GenBank (comprehensive public database of nucleotide sequences and supporting bibliographical and biological annotation.), amino acids are coded in one letter. In the following table, we show the encoding of amino acids in one letter.[95]

Amino acid	Abbreviation 3-letters	Abbreviation 1-letter	Codon(s)
Alanine	Aln	A	GCA, GCC, GCG, GCT
Arginine	Arg	R	CGA, CGC, CGG, CGT, AGA, AGG
Aspartic acid	Asp	D	GAC, GAT
Asparagine	Asn	N	AAC, AAT
Cysteine	Cys	C	TGC, TGT
Glutamic acid	Glu	E	GAA, CAG
Glutamine	Gln	Q	CAA, CAG
Glycine	Gly	G	GGA, GGC, GGG, GGT
Histidine	His	H	CAC, CAT
Isoleucine	Ile	I	ATA, ATC, ATT
Leucine	Leu	L	CTA, CTC, CTG, CTT, TTA, TTG
Lysine	Lys	K	AAA, AAG
Methionine	Met	M	ATG
Phenylalanine	Phe	F	TTC, TTT
Proline	Pro	P	CCA, CCC, CCG, CCT
Serine	Ser	S	TCA, TCC, TCG, TCT, AGC, AGT
Threonine	Thr	T	ACT, ACC, ACG, ACT

Tryptophan	Trp	W	TGG
Tyrosine	Tyr	Y	TAC, TAT
Valine	Val	V	GTA, GTC, GTG, GTT
STOP	-	-	TAG, TAA, TGA

Table 4.1: abbreviation for each amino acid and their DNA codon

4.4 Some proteins amino acid sequences:

In this table we put some protein sequences,to use them to compare between online analysis tools.

Table 4.2: Small data base of proteins amino acid sequences

NAME	DESCRIPTION	SEQUENCE
Sialidase-2 (Mouse)	Sialidase-2 is an enzyme that in humansis encoded by the NEU2 gene.This gene belongs to a family of glycohydrolytic enzymes which removesialic acid residues from glycoproteins and glycolipids. Expression studies in COS7 cells confirmed that this gene encodes a functional sialidase. Its cytosolic localization was demonstratedby cell fractionation experiments.	MATCPVLQKETLFRTGVHA YRIPALLYLKKQKTLLAFAEK RASKTDEHAELIVLRRGSYN EATNRVKWQPEEVVTQAQL EGHRSMNPCPLYDKQTKTL FLFFIAVPGRVSEHHQLHTK VNVTRLCCVSSTDHGRTWS PIQDLTETTIGSTHQEWATFA VGPGHCLQLRNPAGSLLVP AYAYRKLHPAQKPTPFAFCF ISLDHGHTWKLGNFVAENS LECQVAEVGTGAQRMVYLN ARSFLGARVQAQSPNDGLD FQDNRVVSKLVEPPHGCHG SVVAFHNPISKPHALDTWLL YTHPTDSRNRTNLGVYLNQ MPLDPTAWSEPTLLAMGICA YSDLQNMGQGPDGSPQFG CLYESGNYEEIIFLIFTLKQAF PTVFDAQ

Zinc transporter 9 (Human)	Zinc transporter ZIP9, also known as Zrt- and Irt-like protein 9 (ZIP9) and solute carrier family 39 member 9. This protein is the 9th member out of 14 ZIP family proteins, which is a membrane androgen receptor (mAR) coupled to G proteins, and also classified as a zinc transporter protein. ZIP family proteins transport zinc metal from the extracellular environment into cells through cell membrane.	MLPGLAAAAAHRCSWSSLC RLRLRCRAAACNPSDRQEW QNLVTFGSFSNMVPCSHPYI GTLSQVKLYSTNVQKEGQG SQTLRVEKVPSFETAEGIGT ELKAPLKQEPLQVRVKAVLK KREYGSKYTQNNFITGVRAI NEFCLKSSDLEQLRKIRRRS PHEDTESFTVYLRSDVEAKS LEVWGSPEALAREKKLRKE AEIEYRERLFRNQKILREYRD FLGNTKPRSRTASVFFKGPG KVVMVAICINGLNCFFKFLAW IYTGSASMFSEAIHSLSDTCN QGLLALGISKSVQTPDPSHP YGFSNMRYISSLISGVGIFMM GAGLSWYHGVMGLLHPQPI ESLLWAYCILAGSLVSEGATL LVAVNELRRNARAKGMSFYK YVMESRDPSTNVILLEDTAAV LGVIIAATCMGLTSITGNPLYD SLGSLGVGTLLGMVSAFLIYT NTEALLGRSIQPEQVQRLTEL LENDPSVRAIHDVKATDLGLG KVRFKAEVDFDGRVVTRSYL EKQDFDQMLQEIQEVKTPEE LETFMLKHGENIIDTLGAEVD RLEKELKKRNPEVRHVDLEIL
Calmodulin (Sheep)	a calcium-binding protein that mediates cellular metabolic processes (such as the contraction of muscle fibers) by regulating the activity of calcium-dependent enzymes	MADQLTEEQIAEFKEAFSLFD KDGDGTITTKELGTVMRSLG QNPTEAELQDMINEVDADGN GTIDFPEFLTMMARKMKDTD SEEEIREAFRVFDKDGNGYIS AAELRHVMTNLGEKLTDEEV DEMIREADIDGDGQVNYEEF VQMMTAK

Leiomodin-1 (Human muscle)	The leiomodin 1 protein has a putative membrane-spanning region and 2 types of tandemly repeated blocks. The transcript is expressed in all tissues tested, with the highest levels in thyroid, eye muscle, skeletal muscle, and ovary. Increased expression of leiomodin 1 may be linked to Graves' disease and thyroid-associated ophthalmopathy.	MSRVAKYRRQVSEDPDIDSLL ETLSPEEMEELEKELDVVDPD GSVPVGLRQRNQTEKQSTGV YNREAMLNFCEKETKKLMQR EMSMDESKQVETKTDAKNGE ERGRDASKKALGPRRDSDLG KEPKRGGLKKSFSRDRDEAG GKSGEKPKEEKIIRGIDKGRV RAAVDKKEAGKDGRGEERAV ATKKEEEKKGSDRNTGLSRD KDKKREEMKEVAKKEDDEKV KGERRNTDTRKEGEKMKRAG GNTDMKKEDEKVKRGTGNTD TKKDDEKVKKNEPLHEKEAKD DSKTKTPEKQTPSGPTKPSEG PAKVEEEAAPSIFDEPLERVKN NDPEMTEVNVNNSDCITNEILV RFTEALEFNTVVKLFALANTRA DDHVAFAIAIMLKANKTITSLNL DSNHITGKGILAIFRALLQNNTL TELRFHNQRHICGGKTEMEIA KLLKENTTLLKLGYHFELAGPR MTVTNLLSRNMDKQRQKRLQ EQRQAQEAKGEKKDLLEVPK AGAVAKGSPKPSPQPSPKPSP KNSPKKGGAPAAPPPPPPLA PPLIMENLKNSLSPATQRKMGD KVLPAQEKNSRDQLLAAIRSSN LKQLKKVEVPKLLQ

Creatine kinase B-type (Human brain)	A creatine kinase B-type that is encoded in the genome of human. Reversibly catalyzes the transfer of phosphate between ATP and various phosphogens (e.g., creatine phosphate). Creatine kinase isoenzymes play a central role in energy transduction in tissues with large, fluctuating energy demands, such as skeletal muscle, heart and brain.	MPFSNSHNALKLRFPAEDEFP DLSAHNNHMAKVLTPELYAEL RAKSTPSGFTLDDVIQTGVDN PGHPYIMTVGCVAGDEESYEV FKDLFDPIIEDRHGGYKPSDEH KTDLNPDNLQGGDDLDPNYVL SSRVRTGRSIRGFCLPPHCSR GERRAIEKLAVEALSSLDGDLA GRYYALKSMTEAEQQQLIDDH FLFDKPVSPLLLASGMARDWP DARGIWHNDNKTFLVWVNEED HLRVISMQKGGNMKEVFTRFC TGLTQIETLFKSKDYEFMWNPH LGYILTCPSNLGTGLRAGVHIKL PNLGKHEKFSEVLKRLRLQKR GTGGVDTAAVGGVFDVSNADR LGFSEVELVQMVVDGVKLLIEM EQRLEQGQAIDDLMPAQK
Transthyretin (Human)	Transthyretin (TTR or TBPA) is a transport protein in the serum and cerebrospinal fluid that carries the thyroid hormone thyroxine (T4) and retinol-binding proteinbound to retinol. This is how transthyretin gained its name: transportsthyroxine and retinol. The liver secretes transthyretin intothe blood, and the choroid plexus secretes TTR into the cerebrospinal fluid.	MASHRLLLLCLAGLVFVSEAG PTGTGESKCPLMVKVLDAVR GSPAINVAVHVFRKAADDTWE PFASGKTSESGELHGLTTEEE FVEGIYKVEIDTKSYWKALGIS PFHEHAEVVFTANDSGPRRYT IAALLSPYSYSTTAVVTNPKE

Protein CBFA2T3 (Human)	This gene encodes a member of the myeloid translocation gene family which interact with DNA-bound transcription factors and recruit a range of corepressors to facilitate transcriptional repression. The t (16;21) (q24; q22) translocation is one of the less common karyotypic abnormalities in acutemyeloid leukemia. The translocation produces a chimeric gene made up of the 5’-region of the runt-related transcription factor 1 gene fused to the 3’-region of this gene. This gene is also a putative breast tumor suppressor. Alternative splicing results in transcript variants.	MPASRLRDRAASSASGSTCG SMSQTHPVLESGLLASAGCS APRGPRKGGPAPVDRKAKAS AMPDSPAEVKTQPRSTPPSM PPPPPAASQGATRPPSFTPHT HREDGPATLPHGRFHGCLKW SMVCLLMNGSSHSPTAINGAP CTPNGFSNGPATSSTASLSTQ HLPPACGARQLSKLKRFLTTL QQFGSDISPEIGERVRTLVLGL VNSTLTIEEFHSKLQEATNFPL RPFVIPFLKANLPLLQRELLHC ARLAKQTPAQYLAQHEQLLLD ASASSPIDSSELLLEVNENGK RRTPDRTKENGSDRDPLHPE HLSKRPCTLNPAQRYSPSNG PPQPTPPPHYRLEDIAMAHHF RDAYRHPDPRELRERHRPLV VPGSRQEEVIDHKLTEREWA EEWKHLNNLLNCIMDMVEKT RRSLTVLRRCQEADREELNH WARRYSDAEDTKKGPAPAAA RPRSSSAGPEGPQLDVPREF LPRTLTGYVPEDIWRKAEEAV NEVKRQAMSELQKAVSDAER KAHELITTERAKMERALAEAK RQASEDALTVINQQEDSSESC WNCGRKASETCSGCNAARYC GSFCQHRDWEKHHHVCGQSL QGPTAVVADPVPGPPEAAHSL GPSLPVGAASPSEAGSAGPSR PGSPSPPGPLDT

Melanocortin receptor 4 (Human)	Melanocortin 4 receptor is a melanocortin receptor that in humans is encoded by the MC4R gene. It encodes the MC4 protein, a G protein-coupled receptor that binds -melanocyte stimulating hormone (-MSH).	MVNSTHRGMHTSLHLWNRS SYRLHSNASESLGKGYSDGG CYEQLFVSPEVFVTLGVISLLE NILVIVAIAKNKNLHSPMYFFIC SLAVADMLVSVSNGSETIVITLL NSTDTDAQSFTVNIDNVIDSVI CSSLLASICSLLSIAVDRYFTIFY ALQYHNIMTVKRVGIIISCIWAA CTVSGILFIIYSDSSAVIICLITMF FTMLALMASLYVHMFLMARLHI KRIAVLPGTGAIRQGANMKGAI TLTILIGVFVVCWAPFFLHLIFYI SCPQNPYCVCFMSHFNLYLILI MCNSIIDPLIYALRSQELRKTFK EIICCYPLGGLCDLSSRY
Presenilin-1 (Chicken)	Presenilin-1 (PS-1) is a presenilin protein that in humans is encoded by the PSEN1 gene. Presenilin -1 is one of the four core proteins in the gamma secretase complex, which is considered to play an important role in generation of amyloid beta (A) from amyloid precursor protein (APP). Accumulation of amyloid beta is associated with the onset of Alzheimer’s disease.	MTELSAHLPQFQHGQMTENF PDNHLSNTNDNSERRRHDNS ERRRNDNPGSETNGQPQNNI QQVVDQDEEEDEELTLKYGA KHVIMLFVPVTLCMVVVVATIK SVSFYTRKDGQLIYTPFTEET DTIGQRALNSILNAAIMISVIIV MTILLVVLYKYRCYKVIHGWLI ISSLLLLFFFSFIYLGEVFKTYN VAMDYITVALIIWNFGVVGMIC IHWKGPLRLQQAYLIMISALMA LVFIKYLPEWTAWLILAVISVYD LVAVLCPKGPLRMLVETAQER NETLFPALIYSSTMVWLVNM AEEDPEGQRKASKNSTYDKQ APANQSQNEDAEADDGGFSQ EWQQQRDNRIGPIESTPESRA AVQALPSNSQTSEDPEERGVK LGLGDFIFYSVLVGKASATASG DWNTTLACFVAILIGLCLTLLLL AIFKKALPALPISITFGLVFYFAT DNLVQPFMDQLAFHQFYI

Flotillin-1 (Chimpanzee)	Caveolae are small domains on the inner cell membrane involved in vesicular trafficking and signal transduction. FLOT1 encodes a caveolae-associated, integral membrane protein.The function of flotillin 1 has not been determined	MFFTCGPNEAMVVSGFCRSP PVMVAGGRVFVLPCIQQIQRIS LNTLTLNVKSEKVYTRHGVPIS VTGIAQVKIQGQNKEMLAAAC QMFLGKTEAEIAHIALETLEGH QRAIMAHMTVEEIYKDRQKFS EQVFKVASSDLVNMGISVVSY TLKDIHDDQDYLHSLGKARTA QVQKDARIGEAEAKRDAGIRE AKAKQEKVSAQYLSEIEMAKA QRDYELKKAAYDIEVNTRRAQ ADLAYQLQVAKTKQQIEEQRV QVQVVERAQQVAVQEQEIARR EKELEARVRKPAEAERYKLERL AEAEKSQLIMQAEAEAESVRM RGEAEAFAIGARARAEAEQMA KKAEAFQLYQEAAQLDMLLEK LPQVAEEISGPLTSANKITLVSS GSGTMGAAKVTGEVLDILTRLP ESVERLTGVSISQVNHKPLRTA
RanSeq1	Random generated sequence	YQIHIRDIEMHHNHFHHDTA PKWHQPLMNMWRSAVCCAWP DHDDRGCMSPPAKPVHTYWL YWLKVHKFTMFPHHYACMDL TPCKVGAVLNMCSGDAGGAK FANSNYHCQPPCLYCGCQQG ALVERKHQNEIRVTWILFGS NAGQCTHLGGEDCVTITTRQ QSVRDILEIETFYANHNLLR DGNHLDRIYVVGYKHQDLMR QKTAYTKLPHHMCGIQVYSN RKKHDKPHGNQHRWVCSTVP LRTPTASCRF

RanSeq2	Random generated sequence	FGEHGYCCAAHKCYWTNVYY ITPFDNCQLVPYLQGTKPIE HYGNTMYQGHDPLVCDDTSL EGNAYMSKSVVLNKVYARDF YWLMYVCDEYEYHHMTGTQN RCEDMDCNFRWDYLTTYWDC GSFFHITKIWVKVISDHYSA PHQQVWAFCVPLMKFFMKNF PTYKEFHWKFFQMKDAKSWN GMKIQHAAESMNHYLNMLLS APECSAGPNPVKMYPTDEWH CWNPIHINQLDVFCYPANNE LYNILHRLKI

4.5 Comparison part:

Before we start comparison, we have to know some of the important terms that have been used in this comparison:

- UniProtKB:

  UniProt is a freely accessible database of protein sequence and functional information, many entries being derived from genome sequencing projects. It contains a large amount of information about the biological function of proteins derived from the research literature. It is maintained by the UniProt consortium, which consists of several European bioinformatics organisations and a foundation from Washington, DC, United States [89].

- PDB:

  PDB format consists of lines of information in a text file. Each line of information in the file is called a record. A PDB file generally contains several different types of records, arranged in a specific order to describe a structure [40].

- SWISS-PROT:

  is a curated protein sequence database which strives to provide a high level of annotations (such as the description of the function of a protein, its domain structure, post translational modifications, variants, etc), a minimal level of redundancy and high level of integration with other databases [5].

  - The program uses the 'ENTRY_NAME' which is the first field of the ID line as the first line of the title

- The data of the 'DE' and 'OS' lines are collected by the program and are used as the remaining lines of the title
- The 'SQ' line is used to identify the beginning of the sequence. The program collect all the following lines until the termination line is found or end is reached

• Fasta format:

FASTA Format In bioinformatics, it is a text-based format for representing either nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. The format also allows for sequence names and comments to precede the sequences. The format originates from the FASTA software package, but has now become a standard in the field of bioinformatics. The fasta format is based on a simpletext. Each sequence starts with a > followed by the sequence name, an space and, optionally, the description [85].

4.5.1 Comparison:

We applied the protein sequences from the previous table in online analysis tools.

The results obtained are as follows:

		Scan-Prosite	**Pfam**	**Motif Scan**	**MOTIF**
Transthyretin (Human)	**Time spent in searching**	6s	3s	58s	8s
	Number of motif found	2	1	12	3
Sialidase-2 (Mouse)	**Time spent in searching**	6s	3s	75s	11s
	Number of motif found	0	4	20	4
Zinc transporter 9 (Human)	**Time spent in searching**	6s	3s	63s	9s
	Number of motif found	0	5	27	3
Calmodulin (Sheep)	**Time spent in searching**	6s	4s	73s	9s
	Number of motif found	4	32	25	23

Leiomodin-1 (Human muscle)	**Time spent in searching**	7s	3s	70s	9s
	Number of motif found	1	6	55	4
Creatine kinase B-type (Human brain)	**Time spent in searching**	6s	3s	192s	10s
	Number of motif found	3	2	26	5
Protein CBFA2T3 (Human)	**Time spent in searching**	6s	4s	77s	12s
	Number of motif found	3	5	43	7
Melanocortin receptor 4 (Human)	**Time spent in searching**	6s	3s	70s	10s
	Number of motif found	2	1	27	5
Presenilin-1 (Chicken)	**Time spent in searching**	6s	3s	80s	12s
	Number of motif found	0	6	28	2
Flotillin-1 (Chimpanzee)	**Time spent in searching**	6s	3s	63s	10s
	Number of motif found	0	4	19	2
RanSeq1	**Time spent in searching**	6s	17s	57s	17s
	Number of motif found	0	0	14	0
RanSeq2	**Time spent in searching**	6s	24s	64s	9s
	Number of motif found	0	0	7	0

Table 4.3: results of testing protein sequences in the online analysis tools

4.5.2 representation of results:

The chart represent the time spent in searching for motifs:

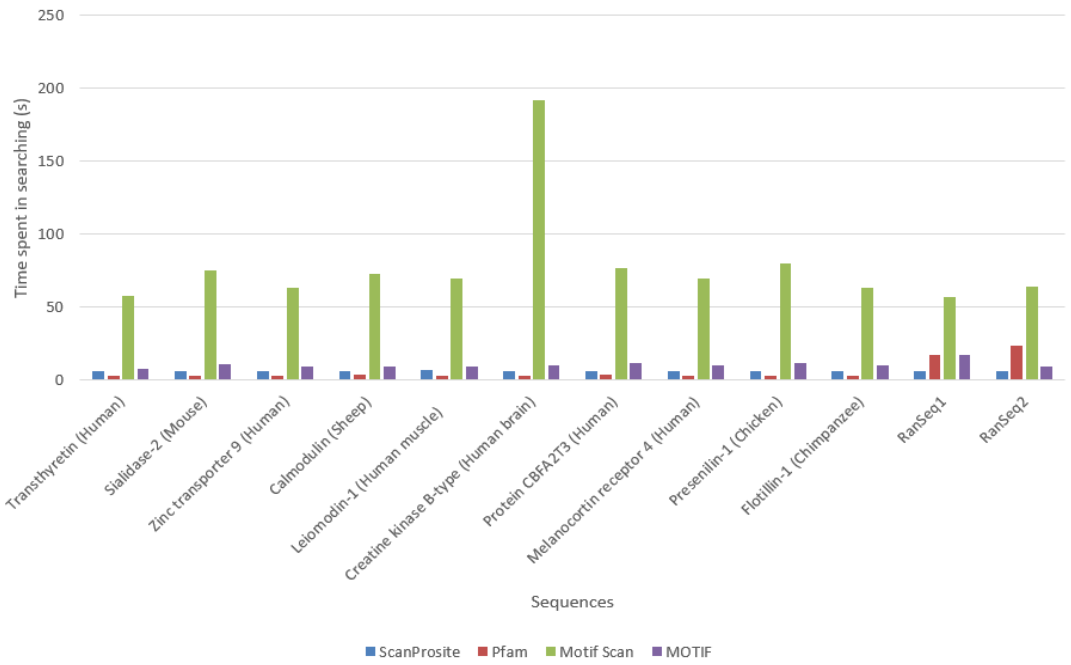


Figure 4.1: Time spent in searching for motifs

- Analysis of results of time spent in search:
  From these results, we note that there is variation in the time taken to research where the Motif Scan tool takes the longest search time in all 12 examples that we have applied, while Pfam takes the shortest duration in most examples (known protein sequences) and took longer to research random sequences.

The chart represent the number of motifs found:

- Analysis of the results of the number of patterns found:
  From these results, we see a variation in the number of patterns found where the motif scan tool found the largest number of patterns in most protein sequences used. While the rest of the tools found few patterns and found no pattern in random sequences. There are also cases(sequences) where these tools can find a large number of patterns.
- Interpretation of results:
  We interpret the duration (length and short) of the search and the number of patterns extracted (little or much) because of the difference in the number of sources and the nature of the inputs and search algorithms. The motif scan

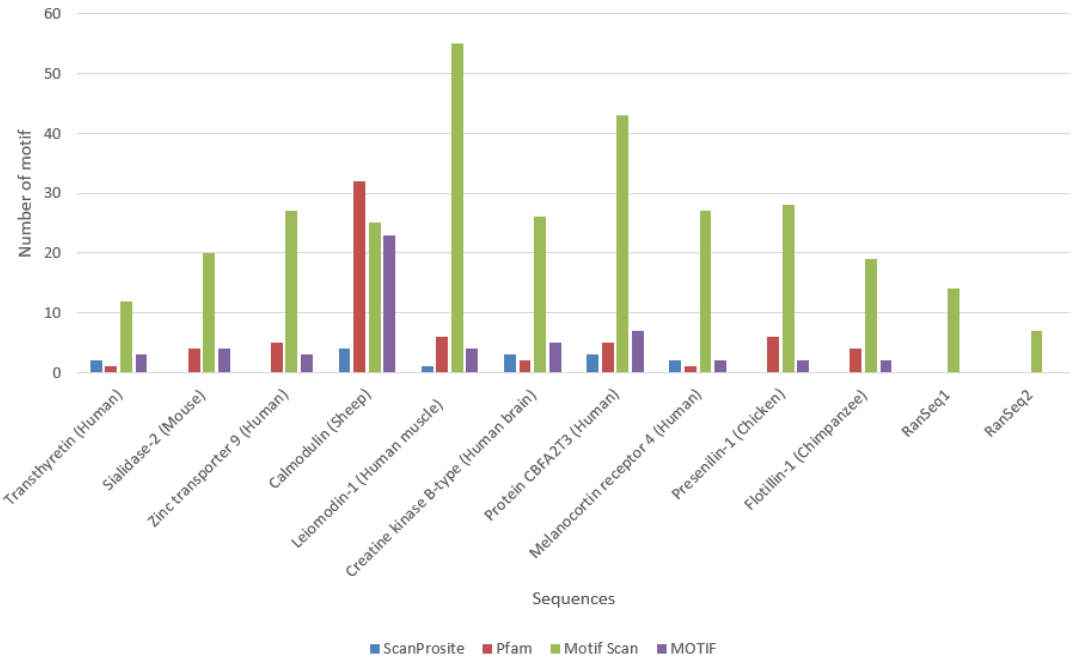


Figure 4.2: Number of motifs found

tool has many sources for research, which explains why it takes so long to extract the result, while the Pfam tool has few sources to search in it, so it takes short duration extract the results.

4.6 Conclusion:

In this chapter we showed that each motif discovery online tool has its own advantages for detecting motifs that other tools may not discover, Which makes it necessary to use multiple online motif discovery tools that implement different algorithms for obtaining significant motifs.

5

General Conclusion

5.1 General Conclusion:

Motif discovery is one of the most popular problems, which has many applications. and It is the process of identifying meaningful patterns in DNA sequences, DNA, or proteins. Motifs vary in length, position, repetition, direction and rules. In our dissertation we provided many motif discovery algorithms, and showed their way of working, theirs strength and weakness. In addition, Our comparison results showed the differences between the online analysis tools In terms of time taken and the number of motifs discovered. Finally we see that every motif discovery tool should process all of input sequences and be suited for any genome. and it should allow motif comparison in a multitude of known databases.

Abstract

This work was done to highlight the working methods of motif discovery algorithms. motifs are different in length, location, frequency, direction and rules and they determine the function of proteins. The aim of this work is to conduct a comparative study of online motif discovery tools to determine their effectiveness and to identify differences between available tools.

Résumé

Ce travail a été fait pour mettre en évidence les méthodes de travail des algorithmes de découverte de motifs. les motifs sont différents dans la longueur, l'emplacement, la fréquence, la direction et les règles et ils déterminent la fonction des protéines. Le but de ce travail est de réaliser une étude comparative des outils de découverte de motifs en ligne afin de déterminer leur efficacité et d'identifier les différences entre les outils disponibles.

ملخص

هذا العمل تم لتسليط الضوء على طرق عمل خوارزميات اكتشاف الأنماط. تكون الأنماط مختلفة في الطول و الموقع و التكرار و الاتجاه و القواعد و هي تحدد وظيفة البروتينات. و يتلخص هدف هذا العمل في إجراء دراسة مقارنة بين أدوات استكشاف الأنماط لمعرفة مدى فاعليتها و التعرف على الاختلافات بين الأدوات المتوفرة.

Bibliography

[1] Sharifah Lailee Syed Abdullah and Hazaruddin Harun. “Species motif extraction using LPBS.” In: *Proceedings of the 4 th International Conference on Computing and Informatics, ICOCI*. 2013.

[2] Zeinab E Ahmed et al. “Energy optimization in low-power wide area networks by using heuristic techniques.” In: *LPWAN Technologies for IoT and M2M Applications* (2020), pp. 199–223.

[3] Doaa Altarawy, Mohamed A Ismail, and Sahar M Ghanem. “MProfiler : A profile-based method for dna motif discovery.” In: *IAPR International Conference on Pattern Recognition in Bioinformatics*. Springer. 2009, pp. 13–23.

[4] Ahmed Mansour Alzohairy. “Introduction to Bioinformatics and Genomics.” In: Academic Library, Dokki, Cairo, 2013. Chap. Introduction of the book.

[5] athina.biol.uoa. *SWISS-PROT*. URL: http://athina.biol.uoa.gr/FT/format_SwissProt.html. (accessed: 27/06/2021).

[6] Timothy L Bailey and Charles Elkan. “The value of prior knowledge in discovering motifs with MEME.” In: *Ismb*. Vol. 3. 1995, pp. 21–29.

[7] Lubica Benuskova. *Lecture 7 : Sequence Motif Discovery*. URL: http://www.cs.otago.ac.nz/cosc348/alignments/Lecture07_MotifSearch.pdf. (accessed: 05/05/2021).

[8] Lounnas bilal. “DISCOVERY AND EXTRACTIONOF PATTERNS IN BIOLOGICAL SEQUENCES.” English. PhD thesis. Mohamed BOUDIAF University - M’sila, 2016.

[9] BiotechBox. *What is PROSITE ? know this BIOINFORMATICS tool*. URL: https://www.youtube.com/watch?v=3g5DHRH-UmA. (accessed: 09/05/2021).

[10] Salim Bouamama, Abdellah Boukerram, and Amer F Al-Badarneh. “Motif finding using ant colony optimization.” In: *International Conference on Swarm Intelligence*. Springer. 2010, pp. 464–471.

[11] Genome BritishColumbia. *understanding-genomics*. URL: https://www.genomebc.ca/why-genomics/understanding-genomics. (accessed: 10-06-2021).

[12] Giulio Caravagna. “Formal Modeling of Biological Systems With Delays.” English. PhD thesis. University of Trieste, 2009.

[13] Alexandra M Carvalho. “Motif representation and discovery.” PhD thesis. Universidade técnica de Lisboa Instituto superior técnico, 2011.

[14] Wei-Lun Chao. “Introduction to pattern recognition.” In: *National Taiwan University, Taiwan* (2009), pp. 1–31.

[15] Alokeparna Choudhury. *String matching algorithm*. URL: https://www.slideshare.net/alokeparnachoudhury/string-matching-algorithm. (accessed: 04/05/2021).

[16] Jean-Michel Claverie and Cedric Notredame. *Bioinformatics for dummies*. John Wiley and Sons, 2006.

[17] Australian Law Reform Commission et al. *Essentially Yours–The Protection of Human Genetic Information in Australia, Volume 1 and Volume 2. Report 96*. 2003.

[18] Clare Bates Congdon et al. “Preliminary results for GAMI : A genetic algorithms approach to motif inference.” In: *2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. IEEE. 2005, pp. 1–8.

[19] Rajeshwar Dass. “Pattern Recognition Techniques : A Review.” In: (Sept. 2018).

[20] Meenakshi Devi. *Ant colony optimization*. URL: https://www.slideshare.net/MeenakshiDevi/ant-colony-optimization-11696728. (accessed: 10/06/2021).

[21] Eleazar Eskin and Pavel A Pevzner. “Finding composite regulatory patterns in DNA sequences.” In: *Bioinformatics* 18.suppl_1 (2002), S354–S363.

[22] expasy. *ScanProsite - user manual*. URL: https://prosite.expasy.org/scanprosite/scanprosite_doc.html. (accessed: 29/06/2021).

[23] Umer Farooq. *Bioinformatics Software*. URL: https://www.slideshare.net/inam12/1053m2. (accessed: 10/06/2021).

[24] Jianxing Feng, Tao Liu, and Yong Zhang. “Using MACS to identify peaks from ChIP-Seq data.” In: *Current protocols in bioinformatics* 34.1 (2011), pp. 2–14.

[25] Vladimir Filipović. “Optimization, classification and dimensionality reduction in biomedicine and bioinformatics.” In: *Biologia Serbica* 39.1 (2017).

[26] Hongwei Ge et al. “Discovery of DNA Motif Utilising an Integrated Strategy Based on Random Projection and Particle Swarm Optimization.” In: *Mathematical Problems in Engineering* 2019 (2019).

[27] genome.jp. *Motif search help page*. URL: https://www.genome.jp/tools/motif/motif_help.html. (accessed: 29/06/2021).

[28] Vivienne Baillie Gerritsen and Amos Bairoch. “Protein Databases.” In: *e LS* (2001).

[29] Ayush Goel. *PATTERN RECOGNITION | Types | Use Cases*. URL: https://www.cronj.com/blog/pattern-recognition-types-use-cases. (accessed: 04/05/2021).

[30] David L González-Álvarez and Miguel A Vega-Rodrıguez. “Hybrid multiobjective artificial bee colony with differential evolution applied to motif finding.” In: *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer. 2013, pp. 68–79.

[31] David L González-Álvarez et al. “Comparing multiobjective artificial bee colony adaptations for discovering DNA motifs.” In: *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer. 2012, pp. 110–121.

[32] Naomi Habib. *Analysis of DNA Motifs Based on a Novel Motif Comparison Method*. Hebrew University of Jerusalem, 2007.

[33] Fatma A Hashim, Mai S Mabrouk, and Walid Al-Atabany. “Review of different sequence motif finding algorithms.” In: *Avicenna journal of medical biotechnology* 11.2 (2019), p. 130.

[34] Jianjun Hu, Yifeng D Yang, and Daisuke Kihara. “EMD : an ensemble algorithm for discovering regulatory motifs in DNA sequences.” In: *BMC bioinformatics* 7.1 (2006), pp. 1–13.

[35] Rui Jiang, Xuegong Zhang, and Michael Q Zhang. *Basics of bioinformatics : Lecture notes of the graduate summer school on bioinformatics of China*. Springer Science and Business Media, 2013.

[36] Susmita Karyakarte and Ila Savant. “Pattern Recognition Process, Methods and Applications in Artificial Intelligence.” In: *Pattern Recognition* 6.11 (2019).

[37] V. Keilis-Borok and A. Soloviev. “Pattern Recognition Methods & Algorithms.” In: *The Abdus Salam International Centre For Theoretical physics*. Ninth Workshop on Non-linear Dynamics and Earthquake Predictions. 2007.

[38] Jonathan Keith. *Bioinformatics Volume I : Data, Sequence Analysis, and Evolution*. Jan. 2017. ISBN: 978-1-4939-6620-2. DOI: 10.1007/978-1-4939-6622-6.

[39] Alastair M Kilpatrick, Bruce Ward, and Stuart Aitken. “Stochastic EM-based TFBS motif discovery with MITSU.” In: *Bioinformatics* 30.12 (2014), pp. i310–i318.

[40] UCSF Computer Graphics Laboratory. *PDB*. URL: https://www.cgl.ucsf.edu/chimera/docs/UsersGuide/tutorials/pdbintro.html. (accessed: 27/06/2021).

[41] Jie Liu, Jigui Sun, and Shengsheng Wang. “Pattern recognition : An overview.” In: *IJCSNS International Journal of Computer Science and Network Security* 6.6 (2006), pp. 57–61.

[42] medlineplus. *What is a chromosome?* URL: https://medlineplus.gov/genetics/understanding/basics/chromosome/. (accessed: 29/06/2021).

[43] Nuno D Mendes et al. “MUSA : a parameter free algorithm for the identification of biologically significant motifs.” In: *Bioinformatics* 22.24 (2006), pp. 2996–3002.

[44] EA Milward et al. “Transcriptomics.” In: (2016).

[45] Cory Mitcheel. *Bioinformatics*. URL: https://www.investopedia.com/terms/b/bioinformatics.asp. (accessed: 02/05/2021).

[46] Attoui Moussa. “Techniques bio-inspirées appliquées à lanalyse des séquences biologiques : Etude comparative.” MA thesis. M’sila: Mohamed BOUDIAF University - M’sila, 2016/2017.

[47] Fahim Muntaha. *An introduction to decision trees*. URL: https://www.slideshare.net/fmuntaha/an-introduction-to-decision-trees-43694869?qid=0c537761-73c2-4337-8110-5a2c09aa7081&v=&b=&from_search=1. (accessed: 28/06/2021).

[48] NUCLINEERS. *What is Bioinformatics ?*. URL: https://nuclineers.com/whats-bioinformatics. (accessed: 19/06/2021).

[49] Corey B Olson et al. “Hardware acceleration of short read mapping.” In: *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*. IEEE. 2012, pp. 161–168.

[50] omicstutorials. *Predicting motif for protein sequence- MyHits motif scan tutorial*. URL: https://omicstutorials.com/predicting-motif-for-protein-sequence-myhits-motif-scan-tutorial/. (accessed: 29/06/2021).

[51] Thomas K P. *BIO-Inspired Algorithms*. URL: https://slideplayer.com/slide/12810069/. (accessed: 09/05/2021).

[52] PROSITE. *The PROSITE database of protein domains, families and functional sites - User Manual*. URL: https://prosite.expasy.org/prosuser.html. (accessed: 09/05/2021).

[53] Nooruldeen Nasih Qader and Hussein Keitan Al-Khafaji. “Motif discovery and data mining in bioinformatics.” In: *Int. J. Comput. Technol* 13.1 (2014), pp. 4082–4095.

[54] JS Raikwal and Kanak Saxena. “Performance evaluation of SVM and k-nearest neighbor algorithm over medical data set.” In: *International Journal of Computer Applications* 50.14 (2012).

[55] C Rajan et al. “Investigation on bio-inspired population based metaheuristic algorithms for optimization problems in ad hoc networks.” In: *International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering* 9.3 (2015), pp. 163–170.

[56] Pritika Ramharack and Mahmoud ES Soliman. “Bioinformatics-based tools in drug discovery : the cartography from single gene to integrative biological networks.” In: *Drug discovery today* 23.9 (2018), pp. 1658–1665.

[57] SR Jino Ramson et al. “Nature inspired optimization techniques for image processingA short review.” In: *Nature Inspired Optimization Techniques for Image Processing Applications* (2019), pp. 113–145.

[58] John E Reid and Lorenz Wernisch. “STEME : efficient EM to find motifs in large data sets.” In: *Nucleic acids research* 39.18 (2011), e126–e126.

[59] Dick de Ridder, Jeroen de Ridder, and Marcel JT Reinders. “Pattern recognition in bioinformatics.” In: *Briefings in bioinformatics* 14.5 (2013), pp. 633–647.

[60] sakilAnsari. *Pattern Recognition | Introduction*. URL: https://www.geeksforgeeks.org/pattern-recognition-introduction. (accessed: 04/05/2021).

[61] Mourad Elloumi Salma Aouled El Haj Mohamed and Julie D. Thompson. *Motif Discovery in Protein Sequences*. URL: http://dx.doi.org/10.5772/65441. (accessed: 05/05/2021).

[62] Tuhin Samanta. *Bioinformatics & Scope In Biotechnology*. URL: https://www.slideshare.net/TuhinSamanta/bioinformatics-amp-scope-in-biotechnologytms. (accessed: 10/06/2021).

[63] Geir Kjetil Sandve and Finn Drabløs. “A survey of motif discovery methods in an integrated framework.” In: *Biology direct* 1.1 (2006), pp. 1–16.

[64] SAWON. *An Overview of Neural Approach on Pattern Recognition*. URL: https://www.analyticsvidhya.com/blog/2020/12/an-overview-of-neural-approach-on-pattern-recognition. (accessed: 04/05/2021).

[65] Priyanka Sharma and Manavjeet Kaur. “Classification in pattern recognition : A review.” In: *International Journal of Advanced Research in Computer Science and Software Engineering* 3.4 (2013).

[66] Rahul Siddharthan. *Bioinformatics : Tasks, techniques, tools*. Institute of Mathematical Sciences, CIT Campus, Taramani, Chennai 600113, 2005.

[67] Christian JA Sigrist et al. “PROSITE : a documented database using patterns and profiles as motif descriptors.” In: *Briefings in bioinformatics* 3.3 (2002), pp. 265–274.

[68] Saurabh Sinha and Martin Tompa. “YMF : a program for discovery of novel transcription factor binding sites by statistical overrepresentation.” In: *Nucleic acids research* 31.13 (2003), pp. 3586–3588.

[69] slideshare. *Cuckoo search algorithm*. URL: https://www.slideshare.net/afar1111/cuckoo-search-algorithm-40842565?from_action=save. (accessed: 10/06/2021).

[70] He Quan Sun et al. "ListMotif : A time and memory efficient algorithm for weak motif discovery." In: *2010 IEEE international conference on intelligent systems and knowledge engineering*. IEEE. 2010, pp. 254–260.

[71] He Quan Sun et al. "RecMotif : a novel fast algorithm for weak motif discovery." In: *BMC bioinformatics* 11.11 (2010), pp. 1–11.

[72] He Quan Sun et al. "Tree-structured algorithm for long weak motif discovery." In: *Bioinformatics* 27.19 (2011), pp. 2641–2647.

[73] Health Sciences Library System. *ScanProsite*. URL: https://www.hsls.pitt.edu/obrc/index.php?page=URL1150393801. (accessed: 29/06/2021).

[74] Axel Thevenot. *Particle Swarm Optimization (PSO) Visually Explained*. URL: https://towardsdatascience.com/particle-swarm-optimization-visually-explained-46289eeb2e14. (accessed: 09/06/2021).

[75] tutorialspoint. *Genetic Algorithms - Quick Guide*. URL: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_quick_guide.htm. (accessed: 09/06/2021).

[76] G. Venter and J. Sobieszczanski-Sobieski. "Particle swarm optimization." In: *AIAA Journal* 41 (2002), pp. 1583–1589.

[77] Dr.Juan Antonio Vizcaino. *proteomics : History and introduction to the course*. URL: https://www.slideshare.net/JuanAntonioVizcaino/introduction-to-the-proteomics-bioinformatics-course-2018. (accessed: 20/06/2021).

[78] Marian Walhout, Marc Vidal, and Job Dekker. *Handbook of systems biology : concepts and insights*. Academic Press, 2012.

[79] Mark Wamalwa. *What is Bioinformatics/ Computational Biology ?*. URL: https://hpc.ilri.cgiar.org/beca/training/AdvancedBFX2013_2/Oct_2013/Bioinformatics-Intro.pdf. (accessed: 20/06/2021).

[80] Shuihua Wang et al. "Multi-objective path finding in stochastic networks using a biogeography-based optimization method." In: *Simulation* 92.7 (2016), pp. 637–647.

[81] Xun Wang and Ying Miao. "GAEM : a hybrid algorithm incorporating GA with EM for planted edited motif finding problem." In: *Current Bioinformatics* 9.5 (2014), pp. 463–469.

[82] Xun Wang et al. "MRPGA : motif detecting by modified random projection strategy and genetic algorithm." In: *Journal of Computational and Theoretical Nanoscience* 10.5 (2013), pp. 1209–1214.

[83] wikipedia. *DNA microarray*. URL: https://en.wikipedia.org/wiki/DNA_microarray. (accessed: 29/06/2021).

[84] wikipedia. *DNA sequencing*. URL: https://en.wikipedia.org/wiki/DNA_sequencing. (accessed: 29/06/2021).

[85] wikipedia. *fasta format*. URL: https://en.wikipedia.org/wiki/FASTA_format. (accessed: 27/06/2021).

[86] wikipedia. *Pfam*. URL: https://en.wikipedia.org/wiki/Pfam. (accessed: 29/06/2021).

[87] wikipedia. *PROSITE*. URL: https://en.wikipedia.org/wiki/PROSITE. (accessed: 09/05/2021).

[88] wikipedia. *RNA-Seq*. URL: https://en.wikipedia.org/w/index.php?title=RNA-Seq. (accessed: 04/05/2021).

[89] wikipedia. *UniProtKB*. URL: https://en.wikipedia.org/wiki/UniProt. (accessed: 27/06/2021).

[90] HUSPI DIGITALIZING THE WORLD. *What Is Pattern Recognition in Machine Learning : Complete Guide*. URL: https://huspi.com/blog-open/pattern-recognition-in-machine-learning. (accessed: 04/05/2021).

[91] Jin Xiong. *Essential bioinformatics*. Cambridge University Press, 2006.

[92] Federico Zambelli, Graziano Pesole, and Giulio Pavesi. "Motif discovery and transcription factor binding sites before and after the next-generation sequencing era." In: *Briefings in bioinformatics* 14.2 (2013), pp. 225–237.

[93] F Zare-Mirakabad et al. "PSOMF : An algorithm for pattern discovery using PSO." In: *Proceedings of the Third IAPR International Conferences on Pattern Recognition in Bioinformatics, Melbourne. Australia*. 2008, pp. 61–72.

[94] Xi Zhang and Ata Kabán. "Experiments with Random Projections Ensembles: Linear Versus Quadratic Discriminants." In: *ICDM Workshops*. 2019, pp. 719–726.

[95] Mehdi Zineb and Meziani Fatima Zohra. "Automatisation du traducteur des séquences ADN en protéines." MA thesis. Constantine: Université des Frères Mentouri Constantine Faculté des Sciences de la Nature et de la Vie, 2017 - 2018.