IMPLEMENTATION OF A MOTIF EXTRACTION ALGORITHM FOR ANALYZING A SEQUENCE OF PROTEINS/DNA

DJEMAI ABIR

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Master

Computer science department
Faculty of Computer science and Mathematics

Supervised by Dr. Lounnas bilal

2017 – 2018

Dedicated to my lovely family and a special thanks to my dear brother **FARID**

ACKNOWLEDGEMENTS

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

We would like to express our heartfelt thanks to **ALLAH** for gave the will and the patience to finish our study despite the encountered difficulties.

We thank everyone who helped us by far or near to realize this final dissertation: We thank **Dr. LOUNNAS Bilal** who has provided us with valuable advice during all stages of this work.

We would also like to thank the members of the jury for have done the honor of accepting to judge and evaluate our work. We thank all our teachers for all the knowledge That they instilled us throughout the two years.

CONTENTS

I INTRODUCTION 1

1 GENERAL INTRODUCTION 2

1.1 Overview 2

1.2 Organization of the dissertation 3

II LITTERATEURS 4

2 BIOINFORMATICS 5

2.1 Introduction 5

2.2 What is bioinformatics? 5

2.3 A short history of bioinformatics 6

2.4 Bioinformatics goals 7

2.5 The biological system 7

2.5.1 molecular biology and bioinformatics 9

2.6 Genetics and genomics 14

2.7 Bioinformatics Tasks 14

2.7.1 Sequence Analysis 14

2.7.2 Sequence alignment 18

2.7.3 Gene prediction 21

2.7.4 Genome Annotation 22

2.7.5 Comparative Genomics 24

2.8 Application of bioinformatics 25

2.9 Relation of other fields 25

2.10 Conclusion 26

3 PATTERN RECOGNITION 27

3.1 Introduction 27

3.2 Pattern recognition 27

3.3 What is a pattern recognition? 28

3.4 Pattern Recognition System 28

3.4.1 The Structure of Pattern Recognition System 28

3.5 The process of pattern recognition 29
3.6 Types of pattern recognition 30
3.6.1 string matching 30
3.6.2 Handwriting matching 33
3.7 Pattern recognition applications 34
3.8 Pattern Recognition algorithms 35
3.8.1 The Naïve Algorithm 36
3.8.2 The Karp-Rabin Algorithm 36
3.8.3 Boyer-Moore Algorithm (BM) 37
3.8.4 Knuth-Morris-Pratt Algorithm (KMP) 37
3.9 State of the Art 38
3.9.1 New models and algorithms 38
3.10 Challenges of pattern recognition 39
3.11 Conclusion 40
4 MOTIF EXTRACTION 41
4.1 Introduction 41
4.2 Motif Extraction 41
4.3 What's a Motif? 42
4.4 Why search for motifs? 42
4.5 Types of Motifs 42
4.5.1 Deterministic motifs 42
4.5.2 Probabilistic motifs 43
4.5.3 Combining deterministic and probabilistic motifs 43
4.6 Motif Representation 44
4.6.1 Motifs and consensus sequences 44
4.7 motifs Description Languages: 45
4.7.1 Profiles 45
4.7.2 Regular expressions 46
4.7.3 Hidden Markov Models(HMMs) 46
4.8 Motif extraction algorithms 47
4.8.1 Steps of building motif discovery algorithm 47
4.8.2 Categories of Motif Discovery Algorithm 48
4.9 Issues in protein sequence motif extraction 51
4.10 Data bases of Motif 52

4.11 Conclusion 53

III OUR CONTRIBUTION 54

5 REALIZATION AND IMPLEMENTATION 55

5.1 Introduction 55

5.2 Ps_scan Algorithm 55

5.2.1 Perl language 56

5.2.2 PROSITE database 56

5.2.3 CMD command: 57

5.2.4 Parameters 57

5.3 Architecture of our application 58

5.3.1 GUI (Graphic User Interface) 59

5.4 Implementation 59

5.4.1 Development and Design 59

5.4.2 Experimental results 60

5.4.3 Our perspective 62

5.4.4 The Interfaces of application 62

5.5 Conclusion 65

IV CONCLUSION 66

6 CONCLUSION 67

6.1 Conclusions 67

BIBLIOGRAPHY 69

LIST OF FIGURES

Figure 1 The biological system and its integrated disciplines 8

Figure 2 The structure of DNA 10

Figure 3 The structure of RNA 11

Figure 4 The central dogma of molecular biology (diagrammatic) DNA is transcribed into mRNA that is translated into protein. In addition, DNA is replicated during cell division with the help of DNA polymerase. Transcription is catalyzed by the RNA polymerase. The mRNA is processed by the spliceosome, before translated into a chain of amino acids in the ribosome. tRNA helps the translation by transporting the right amino acids to the right positions as given by the mRNA. 13

Figure 5 General motif discovery process 18

Figure 6 A sequence alignment, produced by ClustalO, of mammalian histone proteins. 18

Figure 7 glabal and local alignment in protein sequence 19

Figure 8 Central Dogma and Splicing 22

Figure 9 A three types of analysis illustrated on a dataset with two classes and two measurements 27

Figure 10 Steps in Pattern Recognition 30

Figure 11 Freely written string recognition 34

Figure 12 Example of operation of the naive string matcher in a DNA string 36

Figure 13 Example of operation of the naive string matcher in a DNA string 38

Figure 14 Representation of a scoring matrix based on a multiple sequence alignment. 45

Figure 15 A short profiles HMM 47

Figure 16 Architecture of the application 59

Figure 17 The time taken to analyze protein sequence samples with one pattern 61
Figure 18 The time taken to analyze protein sequence samples with all pattern 61
Figure 19 The main interface 62
Figure 20 Scanning of sequence 63
Figure 21 Results of simple text 64
Figure 22 Results of graphical view 64

LIST OF TABLES

Part I

INTRODUCTION

1

GENERAL INTRODUCTION

1.1 OVERVIEW

Pattern recognition is an important and mainstay to many computer science disciplines. most of the pattern recognition researches is focused on string matching which are considered as a very important subject in many other research fields.

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) P in a given text. Generally the size of the pattern to be searched is smaller than the given text. The task itself can be categorized either as exact string matching or approximate string matching ,Also depending upon the kind of application. 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 protien), which can provide a lot of biological information that helps solve many problems. Analysis and extraction of this information is a difficult process due to the quantity and complexity of these data. For the purpose of processing and analyzing these data, the application of computer technologies was applied to molecular biology using computerized techniques and algorithms under the name Bioinformatics. To understand and organize this information on a large scale.

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 extportation 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 motif extraction problems using string matching techniques and the application of its

algorithms whether the exact or approximate on biological data. as well we aspire to implement an algorithm to solve this problem by building a framework tool with interface more flexible for extracting motifs from protien sequences and to provides solutions to the problems listed above.

1.2 ORGANIZATION OF THE DISSERTATION

Our dissertation is divided into two parts, the first one contain a litterateur topics to help the reader to understand the contribution of this dissertation. The part itself contain three chapters, the first one is bioinformatics we talk about the basic information of molecular biology such as tasks ,application and relation between computational biology and other fields ,in the second chapter we discussed system and methods for many pattern recognition problems and we focus on string matching problems due to its important for our dissertation,and for the third chapter we provide the problem of motif extraction and techniques and methods As we mentioned some issues in motif extraction.

The next part is where we talk about our applications with implementation of a motif extraction algorithm called Ps_scan based on prosite database ,we also talked about the user interface whitch we developed to serve this algorithm and make it more flexible as well as some percpectives we aspire.

We end this dissertation by a general conclusion about all our work.

Part II

LITTERATEURS

A three chapters will be provided in this part, the aim of these chapters is to provide a preface of our research interest. The chapters are Bioinformatics, Pattern Recognition, and Motif extraction. All of theme will contains definitions, state of the art, techniques, tools, ...etc.

2

BIOINFORMATICS

2.1 INTRODUCTION

Bioinformatics has become an important part of many areas of biology. In experimental molecular biology, bioinformatics 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, it aids in sequencing and annotating genomes and their observed mutations. It plays a role in the text mining of biological literature and the development of biological and gene ontologies to organize and query biological data. It also plays a role in the analysis of gene and protein expression and regulation.

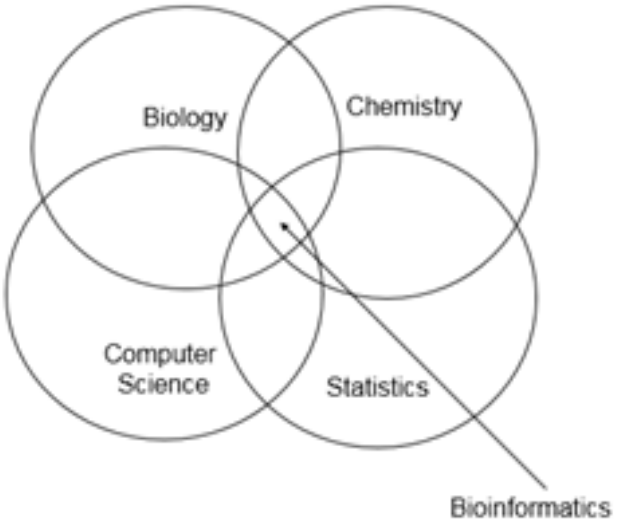
Bioinformatics tools aid in the comparison of genetic and genomic data and more generally in the understanding of evolutionary aspects of molecular biology. At a more integrative level, it helps analyze and catalogue the biological pathways and networks that are an important part of systems biology. In structural biology, it aids in the simulation and modeling of DNA, RNA, proteins as well as bio molecular interactions. [24]

2.2 WHAT IS BIOINFORMATICS?

(Molecular) bio informatics:
Bioinformatics is conceptualizing biology in terms of molecules (in the sense of Physical chemistry) and applying informatics techniques derived from disciplines such as applied math, computer science and statistics to understand and organize the information associated with these molecules, on a large scale[32]. so it is a hybrid science that links biological data with techniques for information storage .it involves the analysis of biological information using computers and statistical techniques, the science of developing and utilizing computer databases

and algorithms to accelerate and enhance biological research. Also it used in analyzing genomes, proteomes (protein sequences), three-dimensional modeling of biomolecules and biologic systems, etc. [15]

Briefly, it is a management information system for molecular biology and has many practical applications.



2.3 A SHORT HISTORY OF BIOINFORMATICS

Historically, the term bioinformatics did not mean what it means today. Paulien Hogeweg and Ben Hesper coined it in 1970 to refer to the study of information processes in biotic systems. This definition placed bioinformatics as a Parallel field to biophysics (the study of physical processes in biological systems) or biochemistry (the study of chemical processes in biological systems).

Bioinformatics started over a century ago when Gregor Mendel, an Austrian monk cross-fertilized different colors of the same species of flowers. Mendel illustrated that the inheritance of traits could be more easily explained if it was controlled by factors passed down from generation to generation. Since Mendel, bioinformatics and genetic record keeping have come a long way. In 1988, the Human Genome organization (HUGO) was founded. The first complete genome map was published of bacteria Haemophilus Influenza. In 1990, the Human Genome Project was started which lasted close to 15 years.

By 1991, a total of 1879 human genes had been mapped. In France, in 1993, Genethon, a human genome research center produced a physical map of the human genome. Three years later, Genethon published the final version of the human genetic map. This concluded the end of the first phase of the Human Genome Project[24], After that in 2000 The A. thaliana genome (100 Mb) is sequenced .
By 2003 Human Genome Project Completed.[26]

2.4 BIOINFORMATICS GOALS

The primary goal of bioinformatics is to increase the understanding of biological processes. What sets it apart from other approaches, however, Its focus on developing and applying computational techniques from different disciplines such as pattern recognition, data mining, and machine learning in order to achieve the primary goal which is understanding the process of biology.[26]

- enable the discovery of new biological insights and to create a global perspective from which unifying principles in biology can be derived.
- To build new techniques that can handle the massive amounts of biological information.
- As the biological informations becomes so larger and more complex, one of the goal of bioinformatics is to build more computational tools to deal with such kind of informations.

2.5 THE BIOLOGICAL SYSTEM

Systems biology is an emerging approach applied to biomedical and biological scientific research. it is based inter-disciplinary field of study that focuses on complex interactions within biological systems,[54] It is a holistic approach to deciphering the complexity of biological systems that starts from the understanding that the networks that form the whole of living organisms are more than the sum of their parts. It is collaborative, integrating many scientific disciplines biology, computer science, engineering, bioinformatics, physics and others to predict how

these systems change over time and under varying conditions, and to develop solutions to the worlds most pressing health and environmental issues.[17]

One of the outreaching aims of systems biology is to model and discover emergent properties, properties of cells, tissues and organisms functioning as a system whose theoretical description is only possible using techniques which fall under the remit of systems biology.

As biological regulation extends on several levels, examples of biological systems are groups of organisms, or on the organ- and tissue in mammals and other animals, the circulatory system, the respiratory system, the nervous system, etc.

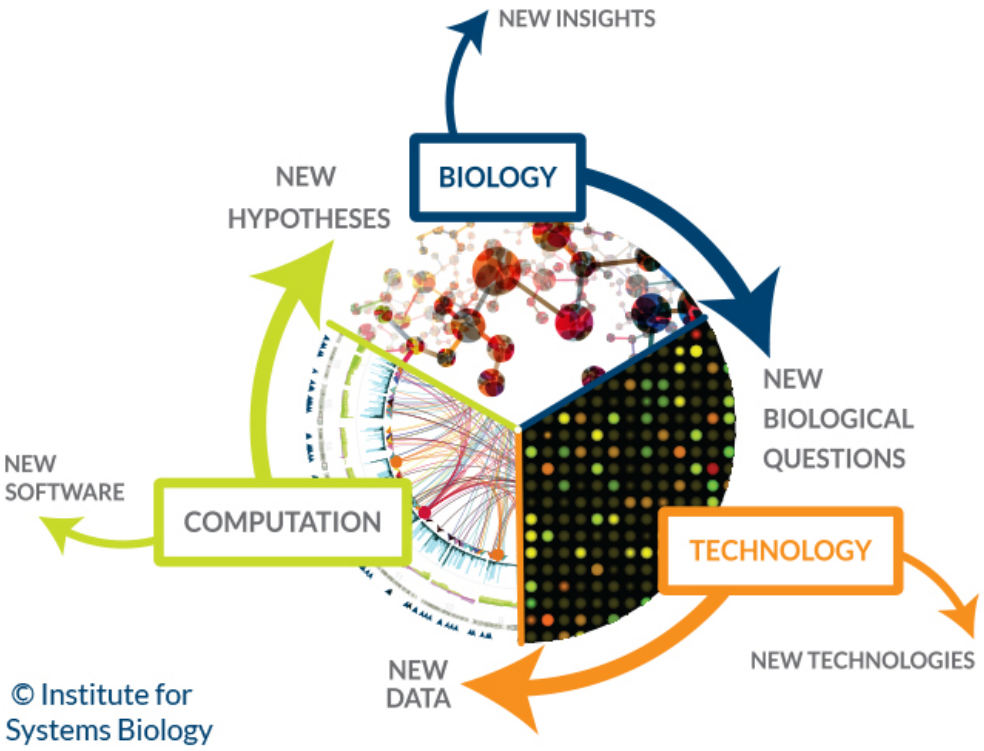


Figure 1: The biological system and its integrated disciplines

2.5.1 *molecular biology and bioinformatics*

Molecular biology is the study of living things at the level of the molecules which control them and make them up. While traditional biology concentrated on studying whole living organisms and how they interact within populations , molecular biology strives to understand living things by examining the components that make them up . Both approaches to biology are equally valid, although improvements to technology have permitted scientists to concentrate more on the molecules of life in recent years.[18]

2.5.1.1 *DNA*

DNA, or deoxyribonucleic acid, is the hereditary material in humans and almost all other organisms. Nearly every cell in a persons body has the same DNA. Most DNA is located in the cell nucleus (where it is called nuclear DNA), but a small amount of DNA can also be found in the mitochondria (where it is called mitochondrial DNA or mtDNA).

The information in DNA is stored as a code made up of four chemical bases: adenine (A), guanine (G), cytosine (C), and thymine (T). The order, or sequence, of these bases determines the information available for building and maintaining an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences.

DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a nucleotide. Nucleotides are arranged in two long strands that form a spiral called a double helix. The structure of the double helix is somewhat like a ladder, with the base pairs forming the ladders rungs and the sugar and phosphate molecules forming the vertical sidepieces of the ladder.

An important property of DNA is that it can replicate, or make copies of itself. This is critical when cells divide because each new cell needs to have an exact copy of the DNA present in the old cell.[42]

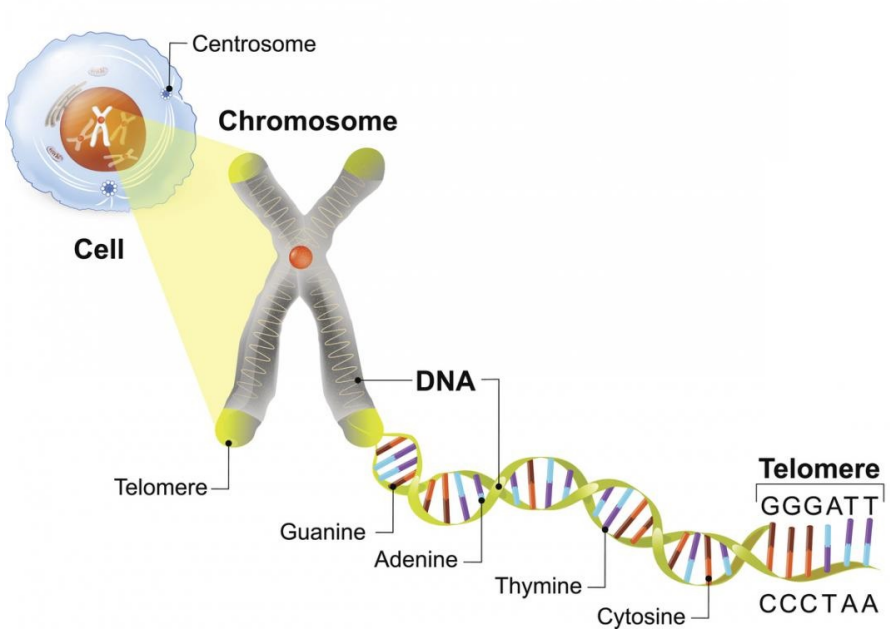


Figure 2: The structure of DNA

2.5.1.2 *RNA*

RNA is typically single stranded and is made of ribonucleotides that are linked by phosphodiester bonds. A ribonucleotide in the RNA chain contains ribose (the pentose sugar), one of the four nitrogenous bases (A, U, G, and C), and a phosphate group.

The subtle structural difference between the sugars gives DNA added stability, making DNA more suitable for storage of genetic information, whereas the relative instability of RNA makes it more suitable for its more short-term functions. The RNA-specific pyrimidine uracil forms a complementary base pair with adenine and is used instead of the thymine used in DNA.

Even though RNA is single stranded, most types of RNA molecules show extensive intramolecular base pairing between complementary sequences within the RNA strand.[48]

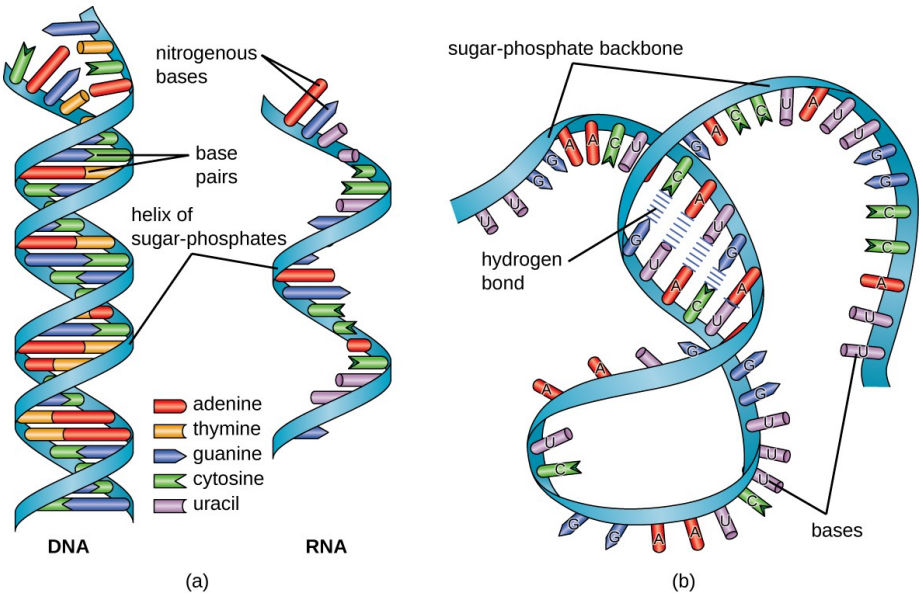


Figure 3: The structure of RNA

2.5.1.3 *GENE*

A gene is the basic physical and functional unit of heredity. Genes, which are made up of DNA, act as instructions to make molecules called proteins. In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases. The Human Genome Project has estimated that humans have between 20,000 and 25,000 genes.

Every person has two copies of each gene, one inherited from each parent. Most genes are the same in all people, but a small number of genes (less than 1 percent of the total) are slightly different between people. Alleles are forms of the same gene with small differences in their sequence of DNA bases. These small differences contribute to each persons unique physical features.[42]

We can also define the gene as a fragment of the genetic information (DNA) corresponding to a protein. We can recapitulate this mechanism as "central dogma" of biology molecular: DNA = RNA = protein = phenotype. where transcription is the property of passing from DNA to RNA and the translation is the process of moving from RNA to protein.

2.5.1.4 *Proteins*

Proteins are one of the most abundant organic molecules in living systems and have the most diverse range of functions of all macromolecules. Proteins may be structural, regulatory, contractile, or protective; they may serve in transport, storage, or membranes; or they may be toxins or enzymes. Each cell in a living system may contain thousands of different proteins, each with a unique function. Their structures, like their functions, vary greatly. They are all, however, polymers of amino acids, arranged in a linear sequence.[28]

The functions of proteins are very diverse because there are 20 different chemically distinct amino acids that form long chains, and the amino acids can be in any order. For example, proteins can function as enzymes or hormones. Enzymes, which are produced by living cells, are catalysts in biochemical reactions (like digestion) and are usually proteins. Each enzyme is specific for the substrate (a reactant that binds to an enzyme) upon which it acts. Enzymes can function to break molecular bonds, to rearrange bonds, or to form new bonds. An example of an enzyme is salivary amylase, which breaks down amylose, a component of starch.[28]

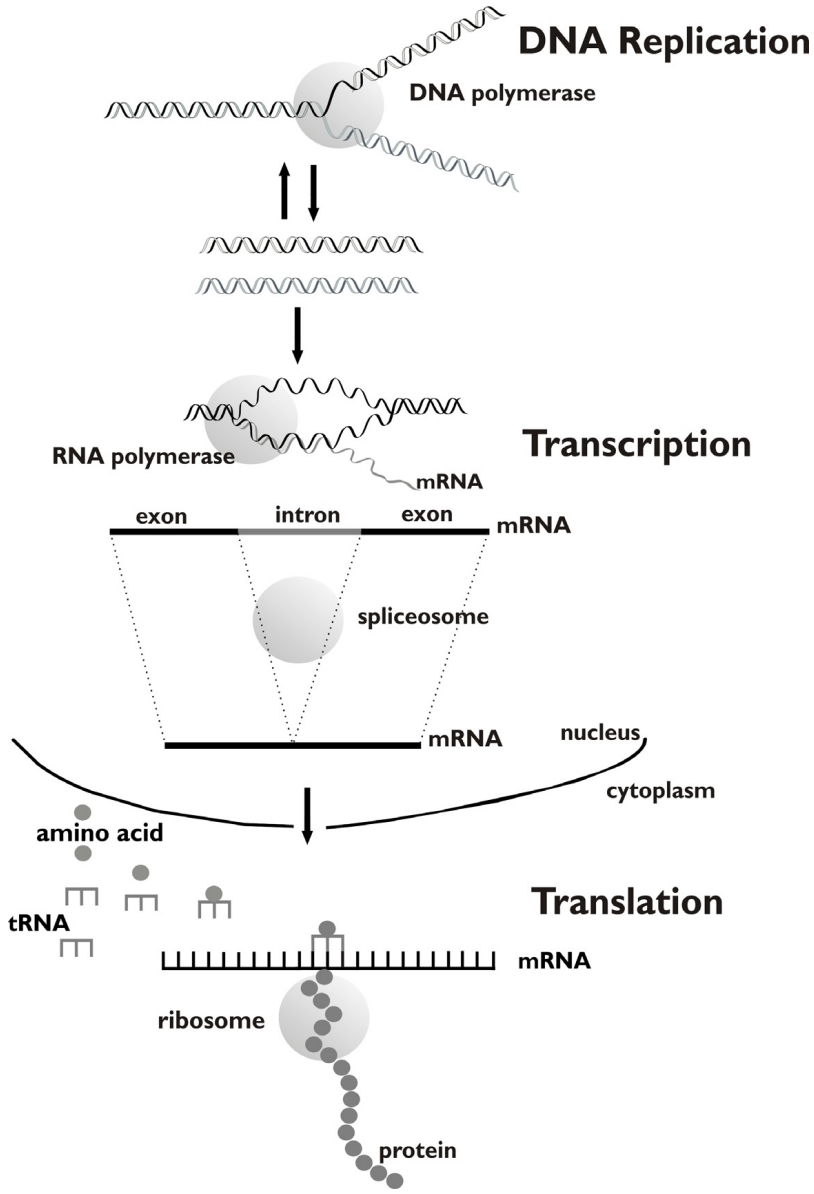


Figure 4: The central dogma of molecular biology (diagrammatic) DNA is transcribed into mRNA that is translated into protein. In addition, DNA is replicated during cell division with the help of DNA polymerase. Transcription is catalyzed by the RNA polymerase. The mRNA is processed by the spliceosome, before translated into a chain of amino acids in the ribosome. tRNA helps the translation by transporting the right amino acids to the right positions as given by the mRNA.[46]

2.6 GENETICS AND GENOMICS

The terms sound alike, and they are often used interchangeably. But there are some important distinctions between genetics and genomics.

Genetics is the study of heredity, or how the characteristics of living organisms are transmitted from one generation to the next via DNA, the substance that comprises genes, the basic unit of heredity. it is involves the study of specific and limited numbers of genes, or parts of genes, that have a known function. In biomedical research, scientists try to understand how genes guide the bodys development, cause disease or affect response to drugs.

Genomics, in contrast, is the study of the entirety of an organisms genes called the genome. Using high-performance computing and math techniques known as bioinformatics, genomics researchers analyze enormous amounts of DNA-sequence data to find variations that affect health, disease or drug response. In humans that means searching through about 3 billion units of DNA across 23,000 genes.

Genomics is a much newer field than genetics and became possible only in the last couple of decades due to technical advances in DNA sequencing and computational biology.[21]

2.7 BIOINFORMATICS TASKS

Due to the complexity of the field of bioinformatics, many tasks has been lunched to deal with different kind of biological problems.

In this section we are going to talk about some of this tasks.

2.7.1 *Sequence Analysis*

In bioinformatics, sequence analysis is the process of subjecting a DNA, RNA or peptide sequence to any of a wide range of analytical methods to understand its features, function, structure, or evolution. Methodologies used include sequence alignment, searches against biological databases, and others. Since the development of methods of high-throughput production of gene and protein sequences,

the rate of addition of new sequences to the databases increased exponentially. Such a collection of sequences does not, by itself, increase the scientist's understanding of the biology of organisms.

However, comparing these new sequences to those with known functions is a key way of understanding the biology of an organism from which the new sequence comes. Thus, sequence analysis can be used to assign function to genes and proteins by the study of the similarities between the compared sequences. Nowadays, there are many tools and techniques that provide the sequence comparisons (sequence alignment) and analyze the alignment product to understand its biology.

Sequence analysis in molecular biology includes a very wide range of relevant topics:

- The comparison of sequences in order to find similarity, often to infer if they are related (homologous)
- Identification of intrinsic features of the sequence such as active sites, post translational modification sites, gene-structures, reading frames, distributions of introns and exons and regulatory elements
- Identification of sequence differences and variations such as point mutations and single nucleotide polymorphism (SNP) in order to get the genetic marker.
- Revealing the evolution and genetic diversity of sequences and organisms
- Identification of molecular structure from sequence alone.[35]

2.7.1.1 *Motif recognition*

The motif recognition problem takes as input a set of known patterns or features that in some way define a class of proteins. The goal is then to search in an unsupervised or supervised way for other instances of the same patterns. As well known, the known motifs in biological sequences Internet. For example, the PRINTS database contains protein fingerprints, where a fingerprint is composed of a group of motifs that characterize a given set of protein sequences with the same molecular function. In contrast, the PROSITE and ELM databases contain

single motifs that correspond to known functionally or structurally important amino acids, such as those involved in an active site or a ligand binding site. The motifs contained in these resources are generally manually curated and the entries in the databases include extensive documentation of the specific biological function associated with the sites.[40]

- Motif representation

Over the years, a variety of motif representation models have been developed to take into account the complexity of Protein and DNA 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 of Protein and DNA sequences.

The motif representation can be divided into two main categories:

1. Deterministic models

   Consensus sequences are the simplest model for representing protein motifs. They can be constructed easily by selecting the amino acid found most frequently at each position in the signal. The number of matches between a consensus and an unknown candidate sequence can be used to evaluate the significance of a potential functional site. However, consensus sequences are limited models, since they do not capture the variability of each position. To support some degree of ambiguity, regular expressions can be used. Regular expressions are typically composed of exact symbols, ambiguous symbols, fixed gaps, and/or flexible gaps. For example, the IQ motif is an extremely basic unit of about 23 amino acids, whose conserved core can be represented by the regular expression:

   [FILV]Qxxx[RK]Gxxx[RK]xx[FILVWY]

   Where x signifies any amino acid, and the square brackets indicate an alternative

2. Probabilistic models

   Although deterministic models provide useful ways to construct human-readable representations of motifs, their main drawback is that they lose

some information. For instance, in the IQ motif discussed above, the first position is usually I and both [RK] are most often R.

Probabilistic models can be used to overcome such loss of information. The position -specific scoring matrix (PSSM, 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 i th position in the sequence motif.[40]

- Motif detection

The models described in the previous section can be applied to the task of scanning a user submitted sequence for matches to known motifs, thus providing evidence for the function of the protein and contributing to its classification in a given protein family. Ideally, a motif model would recognize all and only the members of the family. Unfortunately, this is seldom the case in practice. In the case of deterministic models including consensus sequences and regular expressions, the models are often either too specific leading to a large number of false negative predictions, or too degenerate resulting in many false positives. The statistical power of such models can be estimated using standard measures, such as the positive and negative predictive values (PPV and NPV, respectively).

Given a set of functionally related sequences, the main aim is to find new and a priori unknown motifs that are frequent, unexpected, or interesting according to some formal criteria by used Motif discovery, the methods used to discover such motifs follow the same general schema, as shown in Figure 5. [40]

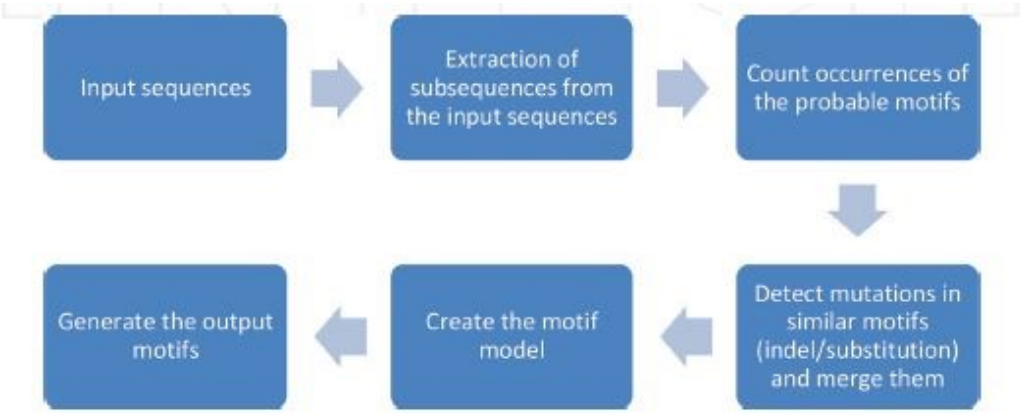


Figure 5: General motif discovery process

2.7.2 *Sequence alignment*

In bioinformatics, a sequence alignment is a way of arranging the sequences of DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Aligned sequences of nucleotide or amino acid residues are typically represented as rows within a matrix. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns. Sequence alignments are also used for non-biological sequences, such as calculating the edit distance cost between strings in a natural language or in financial data.[45]

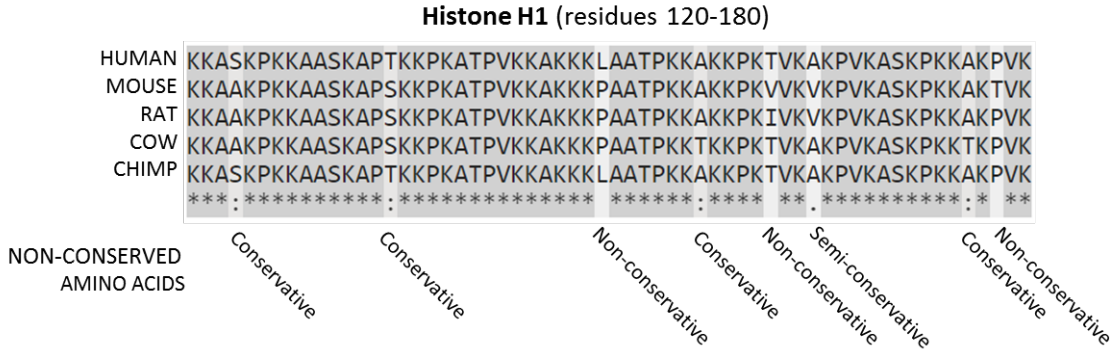


Figure 6: A sequence alignment, produced by ClustalO, of mammalian histone proteins.

1. Types of Sequence Alignment

   Sequence Alignment is one of two types, namely:

   a) Global Alignment : is a matching the residues of two sequences across their entire length, global alignment matches the identical sequences. A general global alignment technique is the NeedlemanWunsch algorithm, which is based on dynamic programming.

   b) Local Alignment: is a matching two sequence from regions which have more similarity with each other . Local alignments are more useful for dissimilar sequences that are suspected to contain regions of similarity or similar sequence motifs within their larger sequence context. The SmithWaterman algorithm is a general local alignment method also based on dynamic programming.[25]

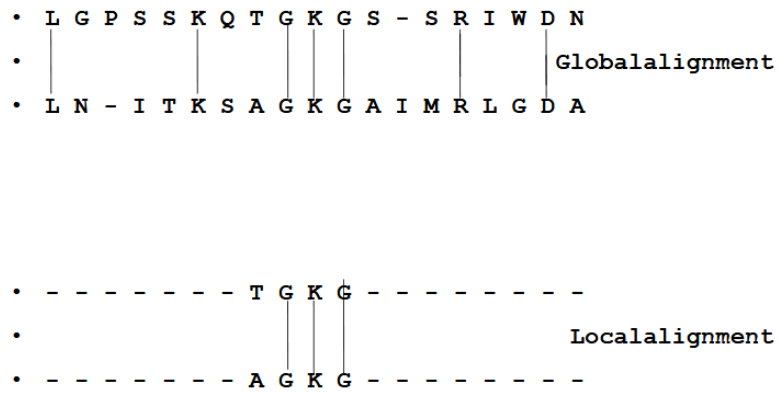


Figure 7: glabal and local alignment in protein sequence

2. Methods of sequence alignment

   a) Dot matrix method

   The dot-matrix approach, which implicitly produces a family of alignments for individual sequence regions, is qualitative and conceptually simple, though time-consuming to analyze on a large scale. In the absence of noise, it can be easy to visually identify certain sequence featuressuch as insertions, deletions, repeats, or inverted repeatsfrom a dot-matrix plot. To construct a dot-matrix plot, the two sequences are written along the top row and leftmost column of a two-dimensional

matrix and a dot is placed at any point where the characters in the appropriate columns matchthis is a typical recurrence plot. Some implementations vary the size or intensity of the dot depending on the degree of similarity of the two characters, to accommodate conservative substitutions. The dot plots of very closely related sequences will appear as a single line along the matrix's main diagonal.

Dot plots can also be used to assess repetitiveness in a single sequence. A sequence can be plotted against itself and regions that share significant similarities will appear as lines off the main diagonal. This effect can occur when a protein consists of multiple similar structural domains.[25]

b) Word or k-tuple methods

Word methods, also known as k-tuple methods, are heuristic methods that are not guaranteed to find an optimal alignment solution, but are significantly more efficient than dynamic programming. These methods are especially useful in large-scale database searches where it is understood that a large proportion of the candidate sequences will have essentially no significant match with the query sequence. Word methods are best known for their implementation in the database search tools FASTA and the BLAST family. Word methods identify a series of short, non-overlapping subsequences ("words") in the query sequence that are then matched to candidate database sequences. The relative positions of the word in the two sequences being compared are subtracted to obtain an offset; this will indicate a region of alignment if multiple distinct words produce the same offset. Only if this region is detected do these methods apply more sensitive alignment criteria; thus, many unnecessary comparisons with sequences of no appreciable similarity are eliminated.[25]

c) The dynamic programming (DP) algorithm

The technique of dynamic programming can be applied to produce global alignments via the Needleman-Wunsch algorithm, and local alignments via the Smith-Waterman algorithm. In typical usage, protein alignments use a substitution matrix to assign scores to amino-acid

matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other. DNA and RNA alignments may use a scoring matrix, but in practice often simply assign a positive match score, a negative mismatch score, and a negative gap penalty. (In standard dynamic programming, the score of each amino acid position is independent of the identity of its neighbors, and therefore base stacking effects are not taken into account. However, it is possible to account for such effects by modifying the algorithm.) A common extension to standard linear gap costs, is the usage of two different gap penalties for opening a gap and for extending a gap. Typically the former is much larger than the latter, e.g. -10 for gap open and -2 for gap extension. Thus, the number of gaps in an alignment is usually reduced and residues and gaps are kept together, which typically makes more biological sense. The Gotoh algorithm implements affine gap costs by using three matrices.[25]

2.7.3 *Gene prediction*

In computational biology, gene prediction or gene finding refers to the process of identifying the regions of genomic DNA that encode genes. This includes protein-coding genes as well as RNA genes, but may also include prediction of other functional elements such as regulatory regions. Gene finding is one of the first and most important steps in understanding the genome of a species once it has been sequenced.[10]

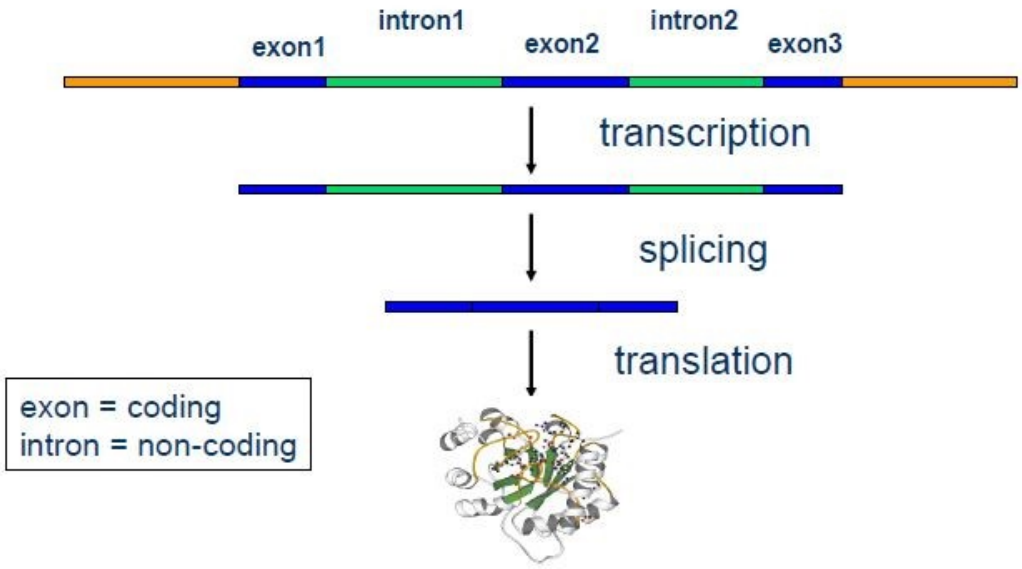


Figure 8: Central Dogma and Splicing

- Approaches to gene finding:

1. Statistical or abinitio methods: These methods attempt to predict genes based on statistical properties of the given DNA sequence Programs are e.g. Genscan, GeneID, GENIE and FGENEH.

2. Comparative methods: The given DNA string is compared with a similar DNA string from a different species at the appropriate evolutionary distance and genes are predicted in both sequences based on the assumption that exons will be well conserved, whereas introns will not. Programs are e.g. CEM (conserved exon method) and Twin scan.

3. Homology methods: The given DNA sequence is compared with known protein structures. Programs are e.g. TBLASTN or TBLASTX, Procrustes and Gene Wise.[41]

2.7.4 *Genome Annotation*

DNA annotation or genome annotation is the process of identifying the locations of genes and all of the coding regions in a genome and determining what those

genes do. An annotation (irrespective of the context) is a note added by way of explanation or commentary. Once a genome is sequenced, it needs to be annotated to make sense of it.[9]

- Annotation Approaches

1. **Nucleotide annotation** The first step of nucleotide annotation is to find sequence that has the features of a gene. Many eukaryotic genes contain specific features, such as introns that separate exons, that can serve are markers for the discovery process. Therefore, it is important to develop a software program that properly recognizes such features. A number of programs are available that perform these searches. A key feature of each of these programs are sensor algorithms that identifies the key structural features.

2. **Naming the genes** The software tool most often used to annotate (or name) a gene is BLAST. This stands for Basic Local Alignment Search Tool. This series of computer programs looks for sequence similarities. Basically, it consists of a query (the sequence to which you are looking for a match) and a database.

3. **Non-gene RNA sequences** Programs are also available that search for non-gene RNA sequences that are important components of the genome. These sequences include the ribosomal RNAs and tRNAs that are essential for protein translation. In addition, the small nuclear RNAs important to processes such as RNA splicing are necessarily components of the genome. These sequences exhibit a high degree of conservation, and therefore, are easily recognizable. Finally, the recent discovery of microRNAs has added another sequence class. These RNAs act as suppressors of gene expression by binding to the mRNA of specific genes.

4. **A novel search for controlling element motifs** All genes are controlled by sequences Upstream of the transcriptional start site. A number of the sequences are important because they represent the site to which transcription factor, proteins that control gene expression, bind. A major goal of annotation would be to describe those sequences, and eventually determine

how universal those sequences are in the promoter of specific genes. The first step is to describe such sequences in a reference species and use that information for further comparative analyses.[43]

2.7.5 *Comparative Genomics*

Comparative genomics is an exciting field of biological research in which researchers use a variety of tools, including computer-based analysis, to compare the complete genome sequences of different species. By carefully comparing characteristics that define various organisms - including the genomes of organisms ranging from humans to chimpanzees to yeast - researchers can pinpoint regions of similarity and difference. This information can help scientists better understand the structure and function of human genes, and develop new strategies to combat human disease.[39]

- Results the field of comparative genomics produced
  - A study discovered that about 60 percent of genes are conserved between fruit flies and humans, meaning that the two organisms appear to share a core set of genes.
  - comparative genomics analysis of six species of yeast prompted scientists to significantly revise their initial catalog of yeast genes and to predict a new set of functional elements that play a role in regulating genome activity, not just in yeast but across many species.
  - Scientists have found genes that increase muscling in cattle by twofold; they found the same genes in racing dogs, and such results may foster human performance studies.
  - In recent years, researchers in the National Human Genome Research Institute (NHGRI) intramural program also have studied the genomics of various cancer types in dogs, including common cancers and other diseases, to try to develop new insights into the human form of the condition. In some cases, they have mapped genes contributing to these disorders. [39]

2.8 APPLICATION OF BIOINFORMATICS

Bioinformatics is the use of information technology in biotechnology for the data storage, data warehousing and analyzing the DNA sequences. There is a tremendous application of bioinformatics in the field of homology and similarity tools, protein function analysis, personalized medicine, Gene therapy, Drug development, Comparative Studies and also climate change studies. Computational methodologies have turn into a noteworthy piece of structure based medication outline. Structure-based medication outline uses the three dimensional structure of a protein focus to plan hopeful medications that are anticipated to tie with high natural inclination and selectivity to the objective. In this survey computational systems for expectation of the protein structure are depicted and their utilization towards the medication outline.[37]

- Heath and Drug Discovery

  The developing world suffers the major burden of infectious disease, yet the range of drugs available for the treatment of many infectious diseases is limited. Moreover, some currently available drugs are difficult to access or administer in developing country settings, while others remain unaffordable at the patient or health facility level; meanwhile, there is increasing resistance to some drugs. Many new public-private partnerships are developing new drugs for the diseases of poverty and they need new leads to work from[50]

2.9 RELATION OF OTHER FIELDS

Bioinformatics is a science field that is similar to but distinct from biological computation, while it is often considered synonymous to computational biology. Biological computation uses bioengineering and biology to build biological computers, whereas bioinformatics uses computation to better understand biology. Bioinformatics and computational biology involve the analysis of biological data, particularly DNA, RNA, and protein sequences. The field of bioinformatics experienced explosive growth starting in the mid-1990s, driven largely by the Human Genome Project and by rapid advances in DNA sequencing technology.

Analyzing biological data to produce meaningful information involves writing and running software programs that use algorithms from graph theory, artificial intelligence, soft computing, data mining, image processing, and computer simulation. The algorithms in turn depend on theoretical foundations such as discrete mathematics, control theory, system theory, information theory, and statistics.[14]

2.10 CONCLUSION

With the confluence of biology and computer science, the computer applications of molecular biology are drawing a greater attention among the life science researchers and scientists these days. As it becomes imperative for biologists to seek the help of information technology professionals to accomplish the ever growing computational requirements of a host of exciting and needy biological problems, the synergy between modern biology and computer science is to blossom in the days to come.

This is only a quick overview of things that are done in computational biology and bioinformatics.

3

PATTERN RECOGNITION

3.1 INTRODUCTION

Pattern recognition has become more and more popular and important to us and it induces attractive attention coming from a wider areas. In this chapter Pattern recognition was introduced including concept, method, application and . At the same time, System and process of pattern recognition were summarized. On the end, we talked about some of the Pattern recognition algorithms and their evolution over time, as well as the challenges facing the process of Pattern recognition .

3.2 PATTERN RECOGNITION

Since its birth in the 1950s, a number of different views on pattern recognition have been taken, focusing on syntax, structure or statistics of the data studied. Statistical pattern recognition has become the predominant paradigm and is mainly concerned with developing theory and methods for Figure 9 illustrates these three problems. Regression, similar to classification but concerned with the prediction of real-valued outputs, is not considered as widely.[30]

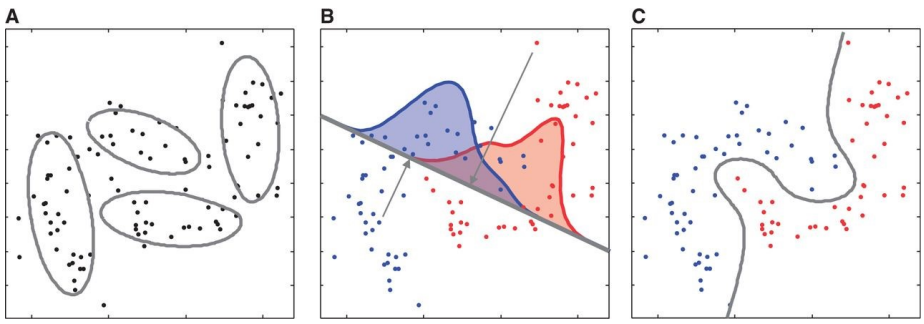


Figure 9: A three types of analysis illustrated on a dataset with two classes and two measurements

Three types of analysis illustrated on a dataset with two classes and two measurements. (A) Clustering: using a mixture of Gaussians, four clusters are fitted to the data. (B) Dimensionality reduction: in linear discriminant analysis, the data are projected on a line such that the classes are separated as well as possible. (C) Classification: the decision boundary of a support vector classifier (3rd degree polynomial kernel) separates the two classes.[30]

3.3 WHAT IS A PATTERN RECOGNITION?

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning.

Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning).[12]

3.4 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.

3.4.1 *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 convert original information into vector which can be dealt with by computer. Pattern analysis task is to process the data (vector), 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 / description showed in fig1 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.[52]

3.5 THE PROCESS OF PATTERN RECOGNITION

Pattern recognition involves making sense or identifying the objects we see, this technique is known as the template matching hypothesis and the feature detection model.

A template is a pattern used to produce items of the same family. The template matching hypothesis suggests that input patterns are compared with templates. If there is a match, the input patterns are identified. While the Feature detection model, suggests that the input patterns are broken down into their component parts for identification.

In addition to this, the process of pattern matching depends on the type of the output, on whether learning is supervised or unsupervised.[27]

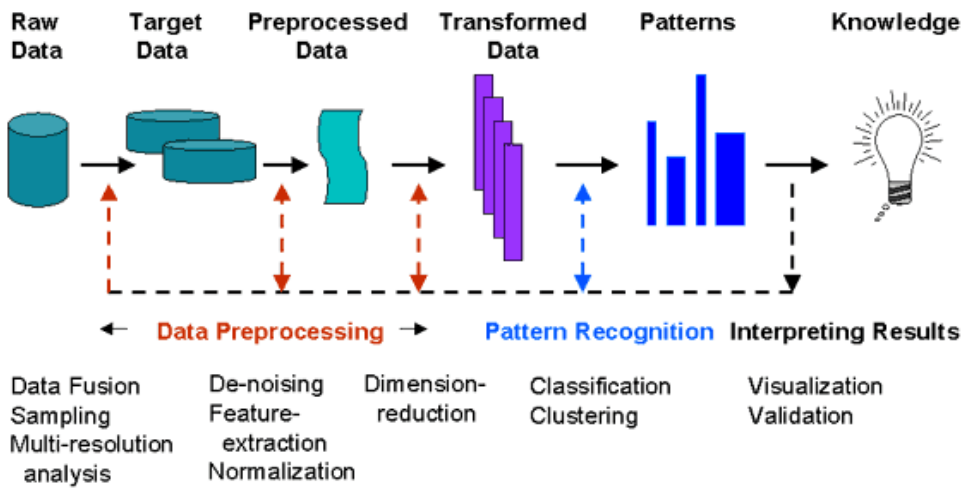


Figure 10: Steps in Pattern Recognition [49]

3.6 TYPES OF PATTERN RECOGNITION

There are many types of patterns recognition exist, some of which can be summarized as follows:

3.6.1 *string matching*

Also known as string matching, string searching, text searching. The fundamental string searching (matching) problem is defined as follows: given two strings a text and a pattern, determine whether the pattern appears in the text. The problem is also known as the needle in a haystack problem.

We can distinguish two types of string searching:

3.6.1.1 *The Exact String Matching*

For the exact string matching, we are given a text $T=t1t2„tn$ and a pattern $P=p1p2„pm$. Our mission is to find whether P appears in T and if it does, where it appears.[47]

Example

We are given *T=aaccgtcacc ggt* and *acc* . We would find *P* does appear in *T* as shown below:

T=aaccgtcaccggt

3.6.1.2 *The Approximate String Matching*

For the approximate string matching problem, we first edit distance which measures the similarity between two strings. Given a string s1 and a string s2, we can transform s2 to s1 by three operations: deletions, insertions and substitutions.

Throughout this sequence, without losing generality,

So we have to decide where to delete and insert characters in pattern and text.[47]

- Deletion Operation

  Let *S1=aactg* and *S2=aaccggt* We may delete *c* in location 4 and *g* in location 5 from *S2* . Then after these two deletion operations as illustrated below, we can transform *S2* to *S1* :

  ```
     1 2 3 4 5 6 7 8
  = a a c - - g t
  = a a c c g g t
  ```

- Insertion Operation

  Let *S1=aactg* and *S2=actt* We may insert *a* and *g* at locations 2 and 5 respectively into *S2* to transform *S2* to *S1* as shown below:

		1	2	3	4	5	6
S_1	=	a	a	c	t	g	t
S_2	=	a	-	c	t	-	t
			a			g	

- Substitution Operation

Let *S1=aactg* and *S2=abctatt* Then we may transform*S2* to *S1* by the following substitution operations at locations 2 and 5. Note that in location 2, *b* is substituted by *a* and in location 5, *a* is substituted by *g*.

		1	2	3	4	5	6	7	8
S_1	=	a	a	c	t	g	t	t	
S_2	=	a	b	c	t	a	t	t	
			a			g			

Example

Let *S1=aagttctattagacg* and *S1=aacgtatatttatag* . *S2* can be transformed into *S1* through the following operations:

		1	2	3	4	5	6	7	8	9	10	1	12	13	14	15	16	17
												1						
S_1	=	a	a	-	g	t	t	c	t	a	t	t	-	a	g	a	c	g
S_2	=	a	a	c	g	t	-	a	t	a	t	t	t	a	t	a	-	g
							t	c							g		c	

The operations are as follows:

1. Deleting at location 3
2. Inserting at location 6

3. Substituting at location 7 by g
4. Deleting t at location 12
5. Substituting t at location 14 by g
6. Inserting c at location 16

Thus there are totally 6 operations, including 2 deletions, 2 insertions and 2 substitutions.[47]

3.6.2 *Handwriting matching*

Handwriting recognition (or HWR) is the technique of receiving the computer for handwritten input from various sources such as paper documents, photographs, as well as touch devices and other devices, and interpret them, the image of the text written by scanning or intelligent recognition of the words.

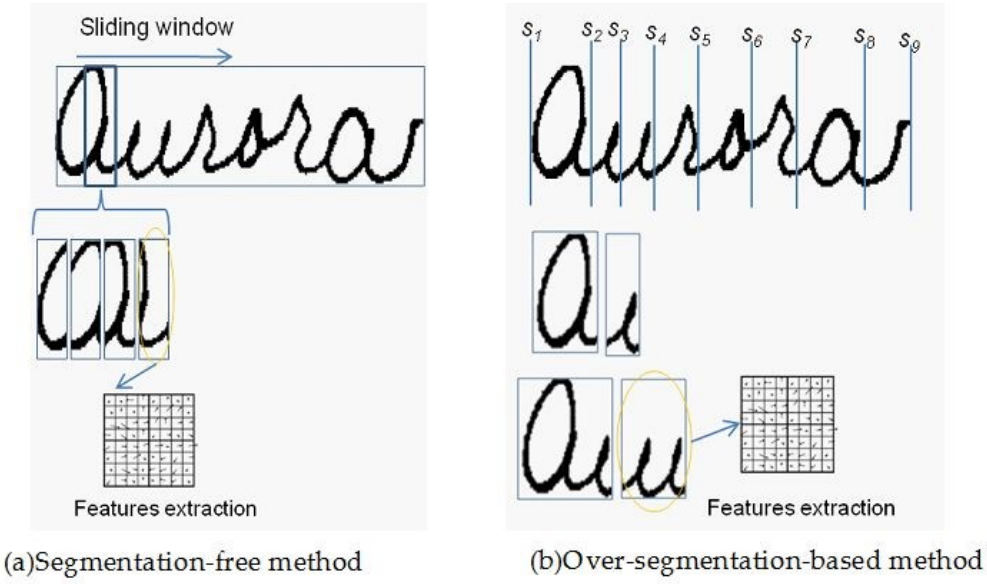
Alternatively, the motion of the pen tip may be sensed "on-line", for example by a pen-based computer screen surface, the task is generally easier as there are more clues available.

- Handwriting matching methods

Handwritten character pattern recognition methods are generally divided into two types: online recognition and offline recognition. Online recognition recognizes character patterns captured from a pen-based or touch-based input device where trajectories of pen-tip or finger-tip movements are recorded, while offline recognition recognizes character patterns captured from a scanner or a camera device as two dimensional images.

In freely written string recognition, we need to consider whether we should select segmentation-free or over-segmentation-based methods. Character segmentation of cursive handwriting is difficult due to the fact that spaces between characters are not obvious. Without character recognition cues and linguistic context, characters cannot be segmented unambiguously. A feasible way to overcome the ambiguity of segmentation is called integrated segmentation and recognition, which is classified into segmentation-free and over-segmentation-based methods

as shown in Figure 11.Segmentation-free methods, mostly combined with hidden Markov model (HMM)-based recognition, simply slice the word pattern into frames (primitive segments) and label the sliced frames, which are concatenated into characters during recognition. Such methods do not sufficiently incorporate character shape information. On the other hand, over-segmentation-based methods attempt to split character patterns at their true boundaries and label the split character patterns Character patterns may also be split within them, but they are merged later. This is called over-segmentation.[56]



(a)Segmentation-free method (b)Over-segmentation-based method

Figure 11: Freely written string recognition

3.7 PATTERN RECOGNITION APPLICATIONS

Pattern recognition is used in any area of science and engineering that studies the structure of observations. It is now frequently used in many applications in manufacturing industry, healthcare, and the military. Examples include the following.[52]

- ***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, such as: X-ray mammography Highlighting potential tambours on a mammogram.
- ***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 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.

3.8 PATTERN RECOGNITION ALGORITHMS

There are many algorithms used patterns recognition this Algorithms depend on the type of label output, on whether learning is supervised or unsupervised like which used the technique of data mining , and on whether the algorithm is statistical or non-statistical in nature. Statistical algorithms can further be categorized as generative or discriminative. in this chapter we focused on the algorithms based on string matching .

3.8.1 *The Naïve Algorithm*

The naive algorithm is the simplest and most often used algorithm. It uses a linear and sequential character-based comparison at all positions in the text between yo and yn-m-1, whether or not an occurrence of the pattern x starts at the current position. In case of success in matching the first element of the pattern xo, each element of the pattern is successively tested against the text until failure or success occurs at the last position. After each unsuccessful attempt, the pattern is shifted exactly one position to the right, and this procedure is repeated until the end of the target is reached. The naive search algorithm has several advantages. It needs no preprocessing of any kind on the pattern and requires only a fixed amount of extra memory space.[29]

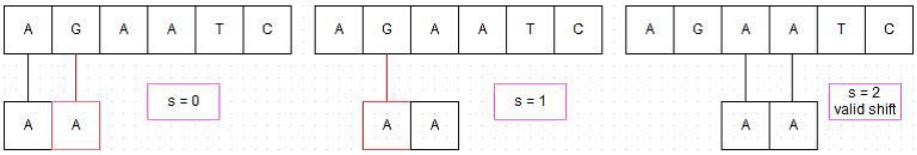


Figure 12: Example of operation of the naive string matcher in a DNA string[33]

3.8.2 *The Karp-Rabin Algorithm*

In theory, hashing functions provide a simple way to avoid the quadratic number of symbol comparisons in most practical situations. These functions run in constant time under reasonable probabilistic assumptions. Hashing technique was introduced by Harrison and later fully analyzed by Karp and Rabin. The Karp-Rabin algorithm is a typical string-pattern-matching algorithm that uses the hashing technique. Instead of checking at each position of the target to see whether the pattern occurs, it checks only whether the portion of the target aligned with the pattern has a hashing value similar to the pattern. To be helpful for the string-matching problem, great attention must be given to the hashing function. It must be efficiently computable, highly discriminating for strings, and computable in an incremental manner in order to decrease the cost of processing. Incremental hashing functions allow new hashing values for a window of the target to be computed step by step without the whole window having to be recomputed. The

time performance of this algorithm is, in the unlikely worst case, O(mn) with an expected time performance of O(m + n).[29]

3.8.3 *Boyer-Moore Algorithm (BM)*

The basic idea behind this solution is that the match is performed from right to left. This characteristic allows the algorithm to skip more characters than the other algorithms, for example if the first character matched of the text is not contained in the pattern P[0...m-1], we can skip m characters immediately. As the KMP algorithm, this algorithm preprocesses the pattern to obtain a table which contains information to skip characters for each character of the pattern. But BM algorithm use also another table based on the alphabet. It contains as many entries as there are characters in the alphabet .In the example below, we can easily persuade the advantage of BM algorithm over KMP and the naive one, we only need four attempts to find the valid shift. In this case, the time complexity of the BM algorithm is sub linear :O(N/M).[33]

3.8.4 *Knuth-Morris-Pratt Algorithm (KMP)*

The KMP algorithm is a linear time algorithm, more accurately O(N +M). The main characteristic of KMP is each time when a match between the pattern and a shift in the text fails, the algorithm will use the information given by a specific table, obtained by a preprocessing of the pattern, to avoid re-examine the characters that have been previously checked, thus limiting the number of comparison required. So KMP algorithm is composed by two parts, a searching part which consists to find the valid shifts in the text, where the time complexity is O(N), obtained by comparison of the pattern and the shifts of the text, and a preprocessing part which consists to preprocesses the pattern.

The goal of the preprocessing of pattern consist to obtain a table that gives the next position in the pattern to be processed after a mismatch. For a pattern P[0:::m-1], the table of result of the preprocessing will give for each character j contained in the pattern a value which is defined as the substring that is in the same time the longest prefix of the pattern and the suffix of the substring of

pattern Po[o:::j]. The complexity of the preprocessing part is O(M), applying the same searching algorithm to the pattern itself.

In Figure 13 there is an example where we need three attempts to find a valid shift, whereas with the naive solution, we need four attempts, we could not skip the shift at the position one.[33]

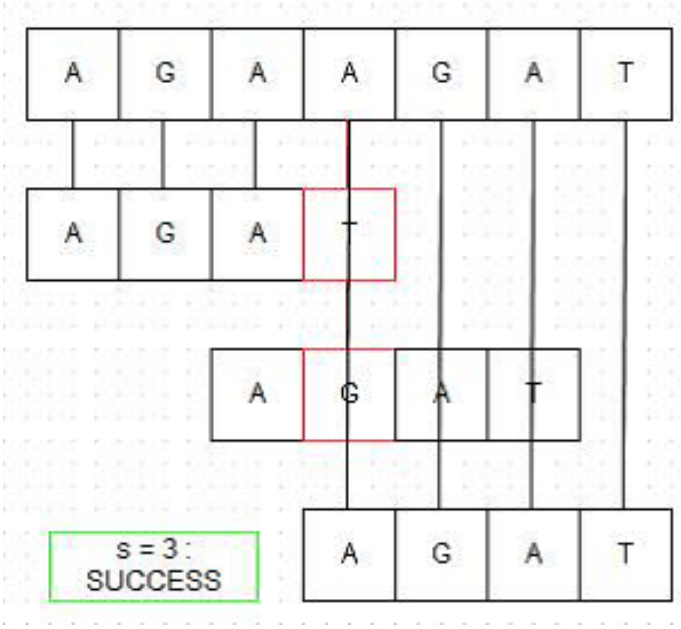


Figure 13: Example of operation of the naive string matcher in a DNA string

3.9 STATE OF THE ART

3.9.1 *New models and algorithms*

Recent developments in pattern recognition and machine learning focus not as much on novel algorithms for existing problems as on finding solutions to (slightly) differently posed problems. Many of these algorithms are potentially useful in bioinformatics, but have not yet been extensively explored in literature. Examples are as follows:

Bag-of-instances representations of objects. In multiple instance learning, the problem is considered where such a bag is labeled positive if at least one instance is labeled positive. This has already found application in drug discovery, relating possible structure of molecules to activity and in predicting protein binding sites.

Semi-supervised learning, for cases where a large number of unlabeled objects are available besides a small set of labeled objects. This has already been exploited in prediction of gene function, in expression-based clustering, prognosis and in the prediction of transcription factor binding sites. A related development is that of positive unlabeled learning, assuming that some objects have a (single positive) label and the remainder is unlabeled, useful for proteinprotein, genetic interaction data, etc. This has been used for predicting disease genes and delineating regulatory networks.

Structured learning, predicting arbitrarily shaped output rather than a single label. Methods include HMMs and, more recently, structured support vector machines and conditional random fields. These have been applied for predicting gene structure (introns/exons) , protein secondary structure , drug activity, enzyme function and interaction networks . A particular case is multi label learning, where several (hierarchically and ontologically) related labels are output, such as in predicting Gene Ontology annotations.

In active learning, a classifier is used to decide which unlabeled object should be labeled next to best improve the classifier. Applications already explored include diagnosis, gene expression sampling, drug discovery and predicting protein interactions and transmembrane helices. Active learning is often used implicitly, when classifier predictions are ranked and the most confident ones are verified experimentally first. Dedicated techniques could further enable current models to guide further experimentation.[30]

3.10 CHALLENGES OF PATTERN RECOGNITION

Next to the developments in pattern recognition itself, new challenges in biology came with new demands on models and algorithms:[30]

It will be increasingly challenging to tie together various heterogeneous data sources in a single application. Pattern recognition algorithms will have to be more robust to missing data, better able to deal with various types of data and scalable to many more objects. Given limited storage and bandwidth, algorithms may have to be able to work on compressed or summarized data.

As tools for measuring and particularly manipulating the cell become more widely available, pattern recognition should help close the systems biology loop by supporting researchers in setting up experiments. Given a limited experimental budget, which interventions together with which measurements are likely to increase our knowledge most? Active learning may prove very useful.

It becomes increasingly important to help users to interpret why a pattern recognition algorithm predicts what it does. This calls for a move from black-box to grey-box models, which allows for a gain in biological knowledge. This requires a shift in emphasis from the performance of a trained predictor to its make-up, stability and uniqueness.

3.11 CONCLUSION

In this chapter, we presented the basic elements of pattern recognition. Data representation process. We focused on major issues and algorithms used as well as potential risks in pattern recognition in general.

4

MOTIF EXTRACTION

4.1 INTRODUCTION

DNA sequences and protein contain different types of information (genes, RNA structures, active sites, organizational structure ...), this information can lead to the discovery of many useful knowledge in biology such as the function of a particular protein sequence, another example of the classification of proteins On different families based on this information. In this chapter we focus on the existed motif in the nucleic acid sequences . Before going further, it is useful to review the concepts and terms associated with this study.

The element is a short structural element that can be found in all members of the protein family. It contains basic remnants of the saved function, not necessarily consecutive, but closes to the 3-D structure, because the reason involves the same function (active location, link site ...). While the pattern or profile is a deteriorating sequence and / or composed of different shapes that can be separated by variable regions.[38]

4.2 MOTIF EXTRACTION

Extracting motifs from sequences is a mainstay of bioinformatics. Analyzing and interpreting sequence data is an important task in bioinformatics. One critical aspect of such interpretation is to extract important motifs (patterns) from sequences. The challenges for motif extraction problem are two-fold: one is to design an efficient algorithm to enumerate the frequent motifs; the other is to statistically validate the extracted motifs and report the significant ones.[55] .

4.3 WHAT'S A MOTIF?

A motif is an idea, object, or concept that repeats itself throughout a text.[11] In bioinformatics Motifs often refers to important functional regions in proteins or DNA, Such as DNA binding sites, and can be used to characterize a protein family, like the signature motifs. Recently Years, the number of sequenced proteins is explosively increased. To automatically classify proteins and predict the functions of proteins rapidly and efficiently becomes an emergent task. The extraction of motifs in proteins can help classify protein families and predict protein functions. Moreover, it provides valuable information on evolution of species.[36]

4.4 WHY SEARCH FOR MOTIFS?

- to find homologous sequences.
- apply existing information to new sequence.
- to find functionalities of important sites.
- to find templates for homology modelling.[38]

4.5 TYPES OF MOTIFS

We can distinguish three specific types of motif as follows :

4.5.1 *Deterministic motifs*

To extract deterministic motifs, we consider a sequence motif as the common subsequence of a set of protein sequences. Therefore the most simple and straightforward way to represent a motif is to use a consensus string that is a single string composed of most likely residues on each position with substitutions and wildcards. However, a motif may occur in each protein in different lengths, and gaps may also appear within the motif. To increase the expressive power, the regular expression is a natural choice.

The regular expression is a powerful notational algebra describing strings and sequences.

The Prosite database is a well-established source of proteins motifs. It records a larg number of proteins classified into different families. For each family, some motifs are given to characterize the family. Prosite uses regular expressions to represent the motifs.[36]

4.5.2 *Probabilistic motifs*

Deterministic motifs are simple and easy to interpret. However the expressive power of the deterministic motifs limited. Compared to the deterministic motif, the probabilistic motif has more expressive power, though it is harder to understand and explain. Because of the advantages of the probabilistic motif, the Prosite database also adopted probabilistic motifs in a later version.

A basic probabilistic model to describe a motif is the Position Weighted Matrix (PWM). In the PWM, it lists the possibilities for all symbols of which they may appear in specific positions. To obtain the possibility that a sequence S is a motif, just multiple the corresponding possibility of each symbol of S at each position. Models like this are also called profiles.[36]

4.5.3 *Combining deterministic and probabilistic motifs*

The boundaries between the two types of representations are not very strict. Proposes a probabilistic regular expression language_Stochastic Regular Expressions (SRE), which has a probabilistic nature but is based on the regular expression, to represent motifs. The syntax of SRE is as follows:
E ::= a | E : E | E*p | E+p | E1(n1) + . . . + Ek(nk) where integers ni, k > 0, 0 < p < 1, E represents the terms of SRE and a represents atomic actions. The above terms represent atomic actions, concatenation, iteration (* iteration and + iteration) and choice . For each term Ei(ni) in the choice, it is chosen with the probability ni/ k ? i=1 ni. In E* and E+, each iteration of E happens with the probability p. It can be seen that each expression of SRE defines a probability function. Thus any subsequence that can be recognized by an SRE expression has a probability score.

The subsequences that cannot not be recognized by the expression have probability scores of zero. An example of the SRE expression is (a : b * 0.7)(2)+ kc * 0.1(3) . For a given string c, it can be recognized with the probability score 0.054 which is computed with above rules.[36]

4.6 MOTIF REPRESENTATION

Consider the N-glycosylation site motif mentioned above: Asn, followed by anything but Pro, followed by either Ser or Thr, followed by anything but Pro This pattern may be written as NP[ST]P where N = Asn, P = Pro, S = Ser, T = Thr; X means any amino acid except X; and [XY] means either X or Y. The notation [XY] does not give any indication of the probability of X or Y occurring in the pattern. Observed probabilities can be graphically represented using sequence logos. Sometimes patterns are defined in terms of a probabilistic model such as a hidden Markov model.[3]

4.6.1 *Motifs and consensus sequences*

The notation [XYZ] means X or Y or Z, but does not indicate the likelihood of any particular match. For this reason, two or more patterns are often associated with a single motif: the defining pattern, and various typical patterns. For example, the defining sequence for the IQ motif may be taken to be:

[FILV]Qxxx[RK]Gxxx[RK]xx[FILVWY]

where x signifies any amino acid, and the square brackets indicate an alternative Usually, however, the first letter is I, and both [RK] choices resolve to R. Since the last choice is so wide, the pattern IQxxxRGxxxR is sometimes equated with the IQ motif itself, but a more accurate description would be a consensus sequence for the IQ motif. [3]

4.7 MOTIFS DESCRIPTION LANGUAGES:

to defines what kind of motifs you can find we can use which called motifs Description Languages among them:

4.7.1 *Profiles*

Profiles are used to model protein families and domains. They are built by converting multiple sequence alignments into position-specific scoring systems (PSSMs). Amino acids at each position in the alignment are scored according to the frequency with which they occur, as represented in Figure 14. Substitution matrices (such as BLOSUM matrices) can be used to add evolutionary distance weighting these scores.[13]

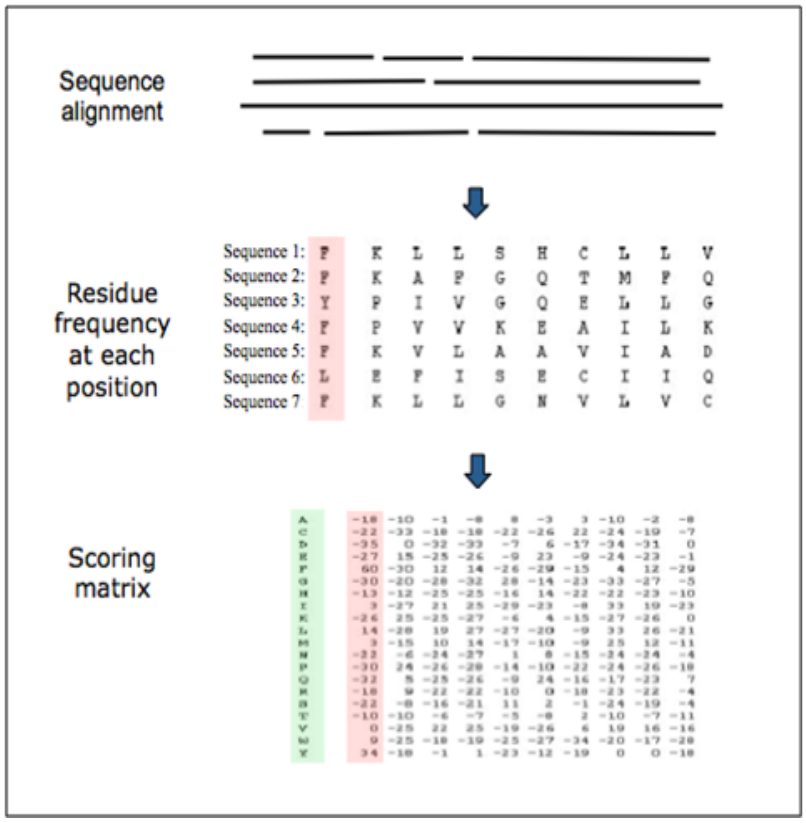


Figure 14: Representation of a scoring matrix based on a multiple sequence alignment.[13]

4.7.2 *Regular expressions*

- Regular expressions can be used to describe sequence motifs.
- They use a simple syntax to describe patterns.
- An example protein pattern:[53]

[DENG]-x-[DEN]-x(0,2)-[DENQK]-[LIVFY]

1. Basic rules for regular expressions

    - Each position is separated by a hyphen -
    - A symbol X is a regular expression matching itself
    - x means any residue
    - surround ambiguities [ ] - a string [XYZ] matches any of the enclosed symbols
    - A string [R]* matches any number of strings that match
    - surround forbidden residues
    - ( ) surround repeat counts.[53]

4.7.3 *Hidden Markov Models(HMMs)*

- HMMs generalize the idea of a profile.
- They can model insertions and deletions in the sequence as well as the letters at conserved positions.
- Profiles can be seen as simple HMMs.
- contains statistical information on observed and expected positional variation - platonic ideal of protein family.[53]

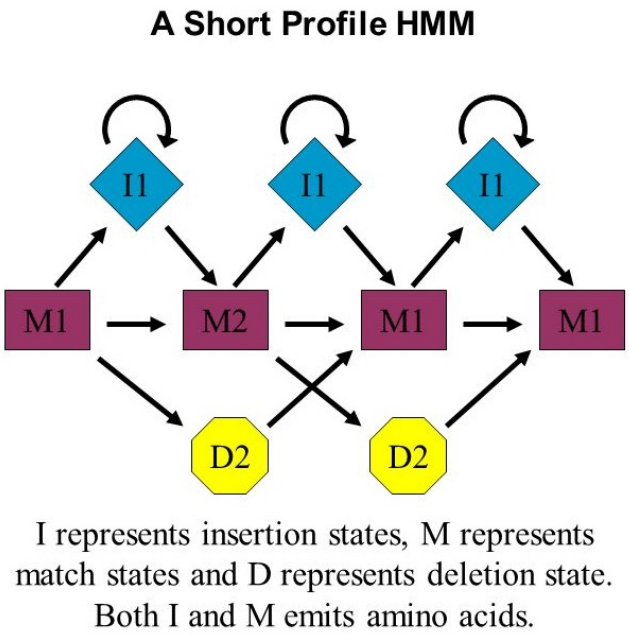


Figure 15: A short profiles HMM

4.8 MOTIF EXTRACTION ALGORITHMS

As there are a growing interest in the regulatory component that can lead to understanding some of the functions of viruses, discovering a new drug, classifying spices, or acquiring many other useful new knowledge in biology. Researchers have developed many algorithms to detect or predict this small part of the biochemical molecule. Each of these algorithms has a different concept in the way data are represented for the discovery of results.

4.8.1 *Steps of building motif discovery algorithm*

The field of motif discovery brings together researchers from several domains. To build algorithms, you have to know about at least three domains. First, biology, through knowing how the system of transcriptional regulation works, coding region, and regulatory binding site...etc. Statistics is the second domain, where building discovery algorithms depends on knowing how to use training data and cleaning noise. The last, and most important domain, is informatics, by knowing

how to design and build efficient algorithms taking into count all techniques from previous domains.[27]

Brazma et al and others have defined some criteria that can interfere in building the kernel of the algorithm:

1. Training set, is the backbone of any discovery algorithm. Choosing the right type of data can make a huge effect on the results that the algorithm delivers.

2. Pattern model can be as simple as a Boolean type (return true or false), and it can also be more complicated, such as a model returning probabilistic results.

3. Design the algorithm. For this purpose, we are inspired by the work of Brazma et al and others , who concluded that the last criterion in building a motif discovery algorithm is designing the algorithm taking into account the two previous criteria, by carrying training set on a sophisticated data structure and developing a matching process for the pattern model.

   Measuring the performance of a motif discovery algorithm is an important task after developing it, due to the sensitivity of applications that implement these kinds of algorithms. There are different ranking methods depending on the pattern model and the type of training set. The quality of the training set may have some influence on the ranking process because, in reality, the sequences coming from biological experiments may contain errors

4.8.2 *Categories of Motif Discovery Algorithm*

4.8.2.1 *Exact string matching algorithms*

Also known as word-based matching algorithms consists of finding one, or more or generally all the occurrences of a string of length m patterns within a text of the total length n characters. We can show some widely used Word-Based Algorithms:

- *MaMF*

  MaMF, or Mammalian Motif Finder, is an algorithm for identifying motifs to which transcription factors bind. The algorithm takes as input a set of promoter sequences, and a motif width(w), and as output, produces a ranked list of 30 predicted motifs(each motif is defined by a set of N sequences, where N is a parameter).

  The algorithm firstly indexes each sub-sequence of length n, where n is a parameter around 4-6 base pairs, in each promoter, so they can be looked up efficiently. This index is then used to build a list of all pairs of sequences of length w, such that each sequence shares an n-mer, and each sequence forms an ungapped alignment with a substring of length w from the string of length 2w around the match, with a score exceeding a cut-off.

  The pairs of sequences are then scored. The scoring function favours pairs which are very similar, but disfavours sequences which are very common in the target genome. The 1000 highest scoring pairs are kept, and the others are discarded. Each of these 1000 'seed' motifs are then used to search iteratively search for further sequences of length which maximise the score(a greedy algorithm), until N sequences for that motif are reached. Very similar motifs are discarded, and the 30 highest scoring motifs are returned as output.[8]

  MaMF's search algorithm is deterministic, and it depends on a simple yet effective indexing strategy to optimize performance. Indexing techniques to speed searches have been used widely and are nicely described in the original report of the BLAST algorithm (Altschul et al., 1990). In the case of MaMF, we can create an index of all n-mers (defined as a short sequence of length n, typically 46 bp long) found per input sequence, which makes identifying locations within a sequence that have a given n-mer a constant time operation. Given indices of two sequences and an n-mer, we can identify all alignments between the two sequences that share that particular n-mer in constant time.[34]

- *MoTeX-II*

  MoTeX-II, is a word-based high-performance computing tool for structured Motif extraction from large-scale datasets. Similar to its predecessor for sin-

gle motif extraction, it uses state-of-the-art algorithms for solving the fixed-length approximate string matching problem. It produces similar and partially identical results to state-of-the-art tools for structured motif extraction with respect to accuracy as quantified by statistical significance measures. Moreover, we show that it matches or outperforms these tools in terms of runtime efficiency by merging single motif occurrences efficiently. MoTeX-II comes in three flavors: a standard CPU version; an OpenMP-based version; and an MPI-based version. For instance, the MPI-based version of MoTeX-II requires only a couple of hours to process all human genes for structured motif extraction on 1056 processors, while current sequential tools require more than a week for this task. Finally, we show that MoTeX-II is successful in extracting known composite transcription factor binding sites from real datasets.[44]

4.8.2.2 *Approximate string matching algorithms*

- *PhyloGibbs*

  The idea of the Gibbs sampler is to sample the space of all possible "binding site configurations" that can be assigned to the input data. A binding site configuration is an assignment of a set of non-overlapping "windows" to input sequences, together with a assignment of "colors" to each of the windows. All windows of the same color are considered sites for the same "motif". Binding site configurations are scored by assuming that all sites belonging to the same motif, i.e. that are colored the same, were drawn from a common weight matrix. All parts of the sequences not in colored windows are scored according to a "background model".

  The program assumes that all binding sites have a fixed length w that must be specified on the command line via the -m option. The input sequence set is parsed into a set of all possible "windows". In the phylogenetically-unrelated case, a window is just a set of w adjacent bases on a sequence. In the phylogenetically aligned case, a window can extend across multiple sequences; this is discussed below. Having built up a list of all possible windows, as well as detailed pointer structures telling us how windows can block other windows (due to overlapping sites), we proceed to "select"

and "unselect" these windows by giving them "colors": "color 0" means not selected, and colors 1, 2, ... correspond to distinct motifs. For example, given a sequence as follows,[5]

```
> Seq 1
ACGATAGATGCGTGATGATATGCCCACAATAATACCCATGTG
 ^^^^^^^^  ^^^^^^^^        ~~~~~~~~~~   ~~~~~~~~~~
```

The sequence is 56 bases long and the chosen motif width w is 8: thus we have 49 possible windows, starting at sites 0 through 48 inclusive. The windows underlined with ^^^^^^ have color 1, the windows underlined with ~~~~~~ have color 2, and all other possible windows have color 0. Thus, the above is an example of a binding site configuration containing 4 sites in total for 2 motifs. Each "configuration" has a "score", which is the posterior probability of the configuration given the sequence; by Bayes theorem, assuming a flat prior on configurations, this is proportional to the probability of seeing the sequence given the configuration. This is the probability that the windows were sampled from weight matrices, multiplied by the probability that the rest of the sequence was sampled from background. We normalize by the probability that the entire sequence was sampled from background. We sample the space of all configurations. An initial simulated-anneal phase seeks the best-scoring configuration, that is, the configuration that best explains the sequence. A tracking phase then assesses the significance of the "clusters of sites" found in the simulated anneal, by keeping statistics of how often each window is a part of each cluster).[5]

4.9 ISSUES IN PROTEIN SEQUENCE MOTIF EXTRACTION

One important issue to consider is the sensitiveness of the algorithm. Most sequence motif extraction algorithms are suitable for discovering motif from closely related proteins, like proteins in one family. But many of them fail to find sparse and subtle sequence motifs from distantly related proteins. Besides, many algorithms are not sensitive enough to extract the sequence motifs accurately if the

motifs are not very conserved in the sequence data set, i.e. having many mismatches and gaps. Another important issue is the robustness of the algorithms. Most algorithms require their input data set carefully selected, consisting only related proteins. They do not perform well on noisy data. However, in certain circumstances,the input sequence data set may be large and noisy, including unrelated proteins not containing the motifs. Therefore, the true motifs will become more subtle, or buried by the background random motifs. A robust algorithm should be able to find the true motifs in such situations. Hence, the robustness of an algorithm need to be considered when we are to design new algorithms or improve existing algorithms.[36]

4.10 DATA BASES OF MOTIF

A number of databases have been constructed that attempt to describe particular protein motifs in terms of patterns and profiles. They allow you to search for patterns or profiles that are indicative of particular functional motifs within a query protein. [19]

Some examples of such databases include:

- **PROSITE** - a collection of patterns and profiles
- **Pfam** - A collection of Profiles generated using hidden Markov models
- **PRINTS** - provider of fingerprints (groups of aligned, un-weighted motifs)
- **BLOCKS** - a database of weighted profiles or blocks

These databases all have different areas of optimum application its difficult to tell which one will give the best results. They all have particular strengths and weaknesses. A new database was recently created called INTERPRO, which collected information from PRINTS, PROSITE, ProDom and pfam, the latter have been used to provide a lot of work so that we can search for multiple databases in one batch.

4.11 CONCLUSION

In this chapter we presented some basic elements about the motif and its types as we focused on some Methods for defining motifs . At the same time we mentioned some of the algorithms based string matching. In this context we summarized how to build the algorithm .on the end the most commonly used databases were summarized.

Part III

OUR CONTRIBUTION

This part has one chapter and will be provided to help the reader to understand our work. The chapter is about realization and implementation of a tool that use a motif extraction algorithm called ps-scan.

5

REALIZATION AND IMPLEMENTATION

5.1 INTRODUCTION

The realization is the last phase in any development process of a system or software. In this chapter, we aim to present briefly the tools and the means used to implement an extraction algorithm called ps-scan. In particular, the chosen design approach,the chosen programming environment, and the interface generated by our application.

5.2 PS_SCAN ALGORITHM

To predict protein function, assign family identity or detect remote homologues, searches against signature databases, also known as secondary databases, are essential.

Ps-scan ScanProsite is a new and improved version of the web-based tool for detecting PROSITE signature matches in protein sequences. For a number of PROSITE profiles, the tool now makes use of ProRule context dependent annotation templates to detect functional and structural intra-domain residues. The detection of those features enhances the power of function prediction based on profiles. Both user-defined sequences and sequences from the UniProt Knowledgebase can be matched against custom patterns, or against PROSITE signatures. To improve response times, matches of sequences from UniProtKB against PROSITE signatures are now retrieved from a pre-computed match database. Several output modes are available including simple text views and a rich mode providing an interactive match and feature viewer with a graphical representation of results.[31]

we can defined ps_scan like a perl program used to scan one or several patterns, rules and /or profiles from PROSITE against one or several protein sequnces in Swiss-Prot or FASTA format.

5.2.1 *Perl language*

Perl, a cross-platform, open-source computer programming language used widely in the commercial and private computing sectors. Perl is a favourite among Web developers for its flexible, continually evolving text-processing and problem-solving capabilities.[51]

In general, Perl is easier to learn and faster to code in than the more structured C and C++ languages. Perl programs can, however, be quite sophisticated. It is often used for developing common gateway interface (CGI) programs because it has good text manipulation facilities, although it also handles binary files.

When compiled, a Perl program is almost as fast as a fully precompiled C language program. A plug-in can be installed for some servers, such as Apache, so that Perl is loaded permanently in memory, thus reducing compile time and resulting in faster execution of CGI Perl scripts.[51]

Perl was ported to non-UNIX operating systems, such as Apple Incs Mac OS and Microsoft Corporations Windows OS, during the 1990s, though it remains more popular in the UNIX community.[6]

5.2.2 *PROSITE database*

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. PROSITE was created in 1988 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 Ioannis Xenarios.

PROSITE's uses include identifying possible functions of newly discovered proteins and analysis of known proteins for previously undetermined activity.

Properties from well-studied genes can be propagated to biologically related organisms, and for different or poorly known genes biochemical functions can be predicted from similarities. PROSITE offers tools for protein sequence analysis and motif detection . It is part of the ExPASy proteomics analysis servers.[2] The database is accessible at

http://www.expasy.org/prosite/.

5.2.3 *CMD command:*

Command Prompt, also known as cmd.exe or cmd (after its executable file name)[4] is a command line interpreter application available in most Windows operating systems. it is used to execute entered commands. Most of those commands are used to automate tasks via scripts and batch files, perform advanced administrative functions, and troubleshoot and solve certain kinds of Windows issues. Command Prompt is officially called Windows Command Processor but is also sometimes called the command shell or cmd prompt, or even referred to by its filename, cmd. it is sometimes incorrectly referred to as "the DOS prompt" or as MS-DOS itself. Command Prompt is a Windows program that emulates many of the command line abilities available in MS-DOS but it is not actually MS-DOS.[7]

5.2.4 *Parameters*

5.2.4.1 *Input Sequence*

The sequence can be in 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 simplicity of FASTA format makes it easy to manipulate and parse sequences using text-processing tools and scripting languages like the R programming language, Python, Ruby, and Perl. [1] The fasta format is based on a simple

text. Each sequence starts with a > followed by the sequence name, an space and, optionally, the description.[16]

```
>seq_1 description
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
>seq_2
ATCGTAGTCTAGTCTATGCTAGTGCGATGCTAGTGCTAGTCGTATGCATGGCTATGTGTG
```

5.2.4.2 *Output format*

In interactive mode, once the analysis is completed, the results will be directly displayed in the selected output view mode inside user's interface.

1. Simple text
   results in simple text format are view without graphical representation of hits and feature prediction.

2. Graphical view
   ps_scan tool displays for each hit within a protein sequence: the hit sequence, the score (for hits against a profile), the PROSITE description and link. In addition, if predicted; biological features associated with each matched sequence are also indicated. While the results are separated into different kinds of hits: hits by 'profiles' , 'patterns', or 'user-defined patterns'[22].

5.3 ARCHITECTURE OF OUR APPLICATION

our application is based on ps_scan algorithm to analysis a given sequence of protein to find one or all motifs exists in this sequence. this is a brief description of a our tool, so the architecture in the Figure 16 describe overview of the work of our tool.

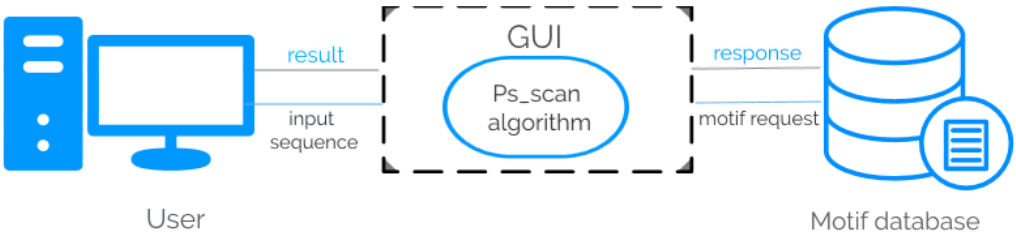


Figure 16: Architecture of the application

5.3.1 *GUI (Graphic User Interface)*

It is a simple graphical interface that allows users to interact with the application. Even if the algorithms are not intended for the general public, we have tried to give a visual, attractive and user-friendly aspect to our application through an interface ergonomic graphic.

5.4 IMPLEMENTATION

Ps_scan was implemented in Perl ,we use Strawberry Perl as a perl environment for MS Windows , containing all we need to run and develop perl applications.[23]

5.4.1 *Development and Design*

for development and design our application interface we relied on NetBeans as an integrated development environment based on java programming, Its a great tool for large-scale projects and makes it easier to bring on new developers because the structure is so visible, also We adopted on the GUI(Graphic User Interface) design-tool that enables us to prototype and design Swing GUIs by dragging and positioning GUI components.

As a programming language, we chose the **Java** language to develop our application interface. This choice of language is motivated by the following reasons:

- Platform Independent: Most of the systems have a built-in Java Runtime Environment, the only prerequisite for running an application that has been designed in Java. As a result, no setups or dependencies have to be injected into a system before executing a Java appliction.[20]
- Portable: Code written in Java can be taken from one computer to the other without having to worry about system configuration details.
- Multithreading: Synchronization and multitasking come as a complimentary gift thanks to Java's multithreading features. These are particularly useful in multimedia and other real-time applications.[20]

5.4.2 *Experimental results*

In what follows, we will present the various tests and evaluations carried out as well as the results obtained by the algorithm used to prove their importance from a point of view of the search for an exact solution to the problem of our study. Then and in order to evaluate the performance of the algorithm,a comparison is conducted between the results on instances of different size of Proteins sequences with one patterns and with all prosite.

5.4.2.1 *Used Data*

To tests the algorithm with our application with the presence of the database of motifs " Prosite " that consists of documentation entries describing protein domains, families and functional sites as well as associated patterns and profiles ,we chose instances of different protein sequences in terms of length ,The first sample consists of 50 to 600 amino acids. The second ranges from 1000 to 2000 and the last one from 3000 and above.

5.4.2.2 *Execution time*

In order to evaluate the performance of the algorithm we calculated the time to analyze each protein sequences with one pattern compared with all the prosite patterns.the results obtained are as follows:

- with one prosite pattern :

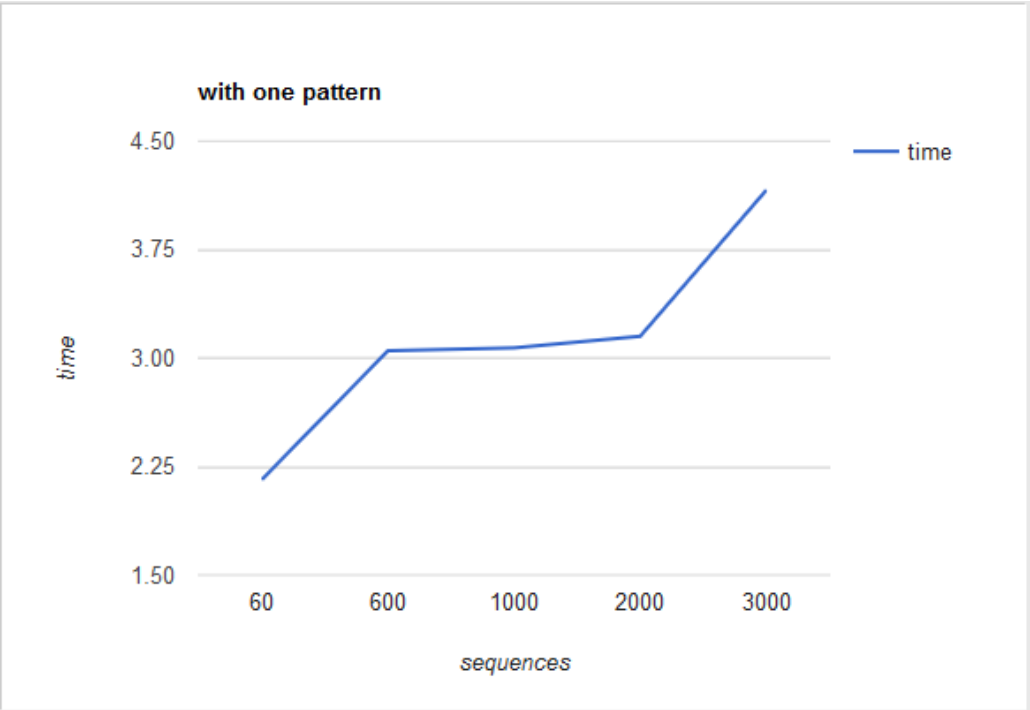


Figure 17: The time taken to analyze protein sequence samples with one prosite pattern

- with all prosite :

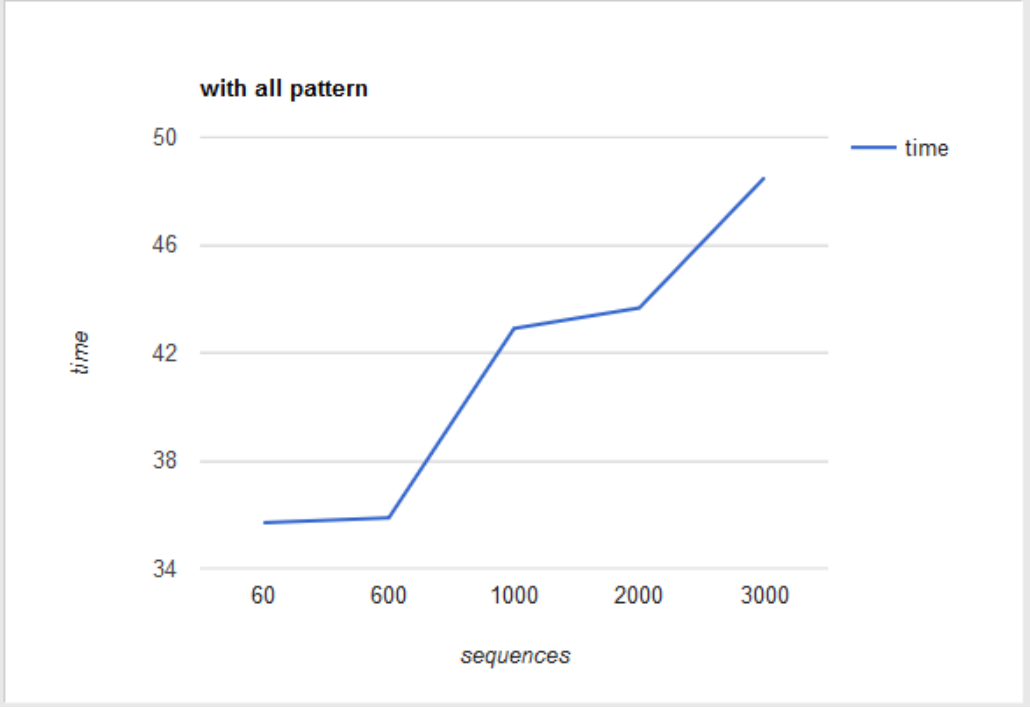


Figure 18: The time taken to analyze protein sequence samples with all prosite pattern

5.4.3 *Our perspective*

We look forward to using more than one database and make our tool works with multiple databases in the same time instead of just looking at the PROSITE database , we look to implementing the Pfam or suiss-prot database.

5.4.4 *The Interfaces of application*

The main interface of our system, from this page can choose the database implements , the file that load or can enter the sequence text in the assigned space, from this interface we can also control the options search and the outputs results format.

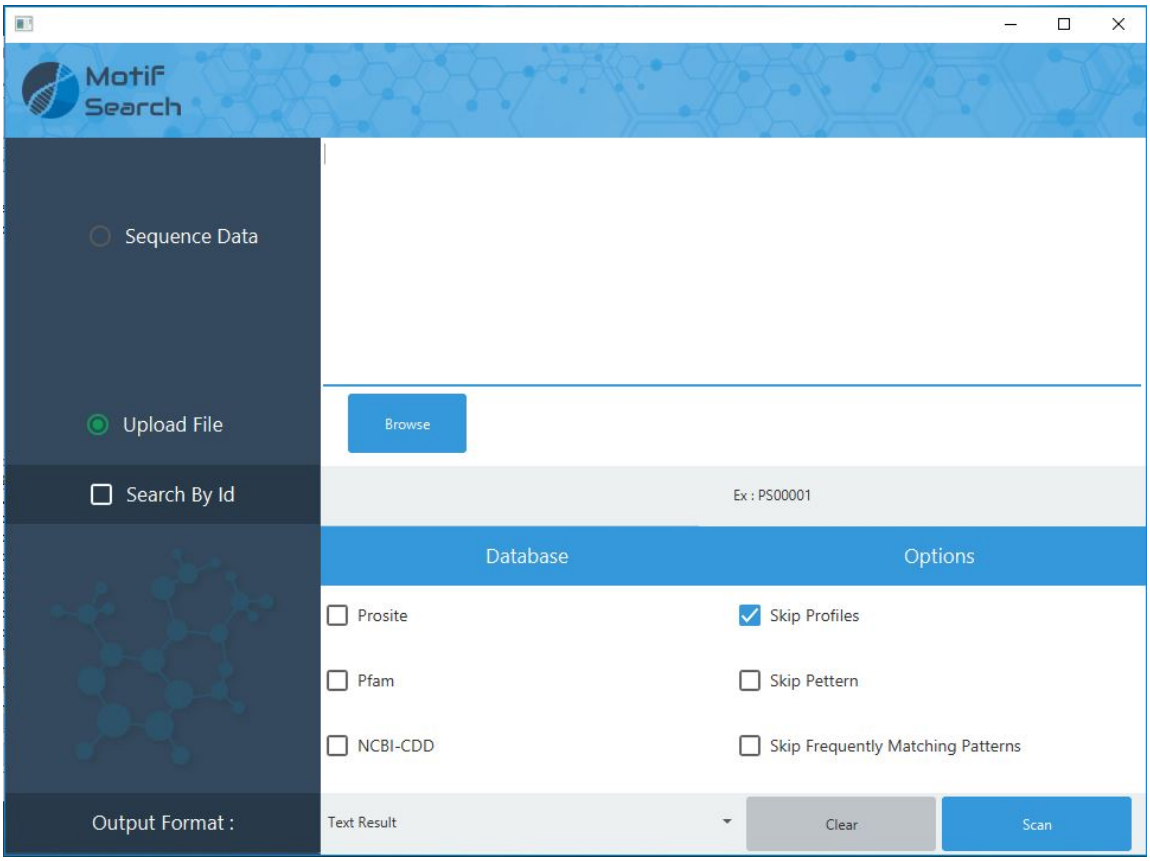


Figure 19: The main interface

5.4.4.1 *The results*

Our tool displays two types of results as the user chooses.

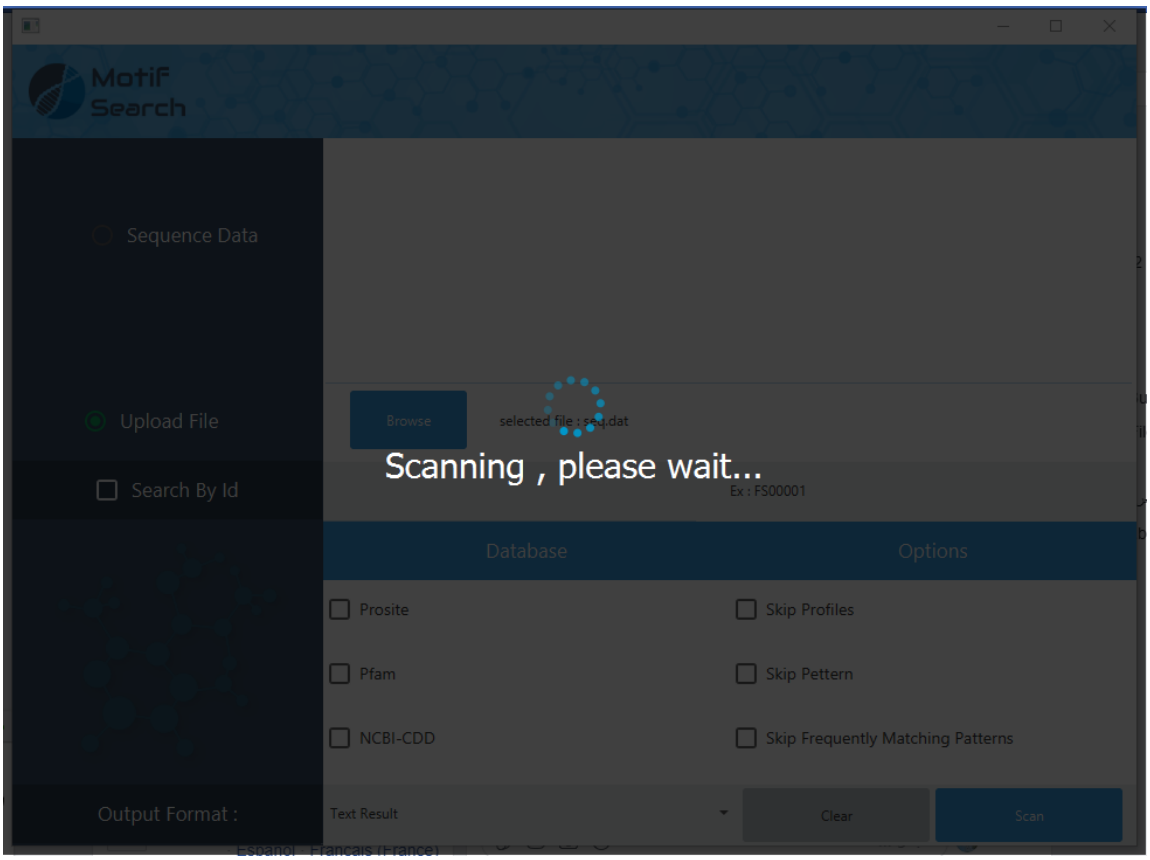


Figure 20: Scanning of sequence

The results obtained with a simple text shown as Figure 21:

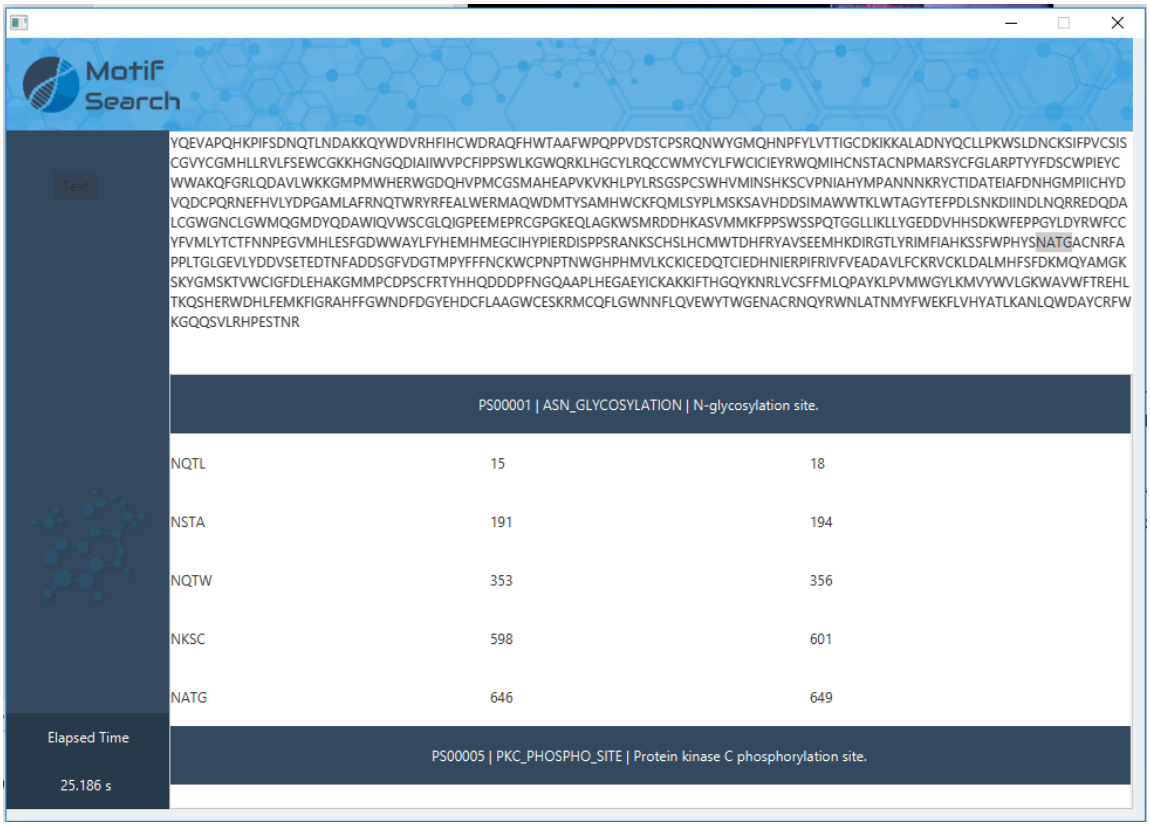


Figure 21: Results of simple text

The results obtained with the graphical view shown as Figure 22:

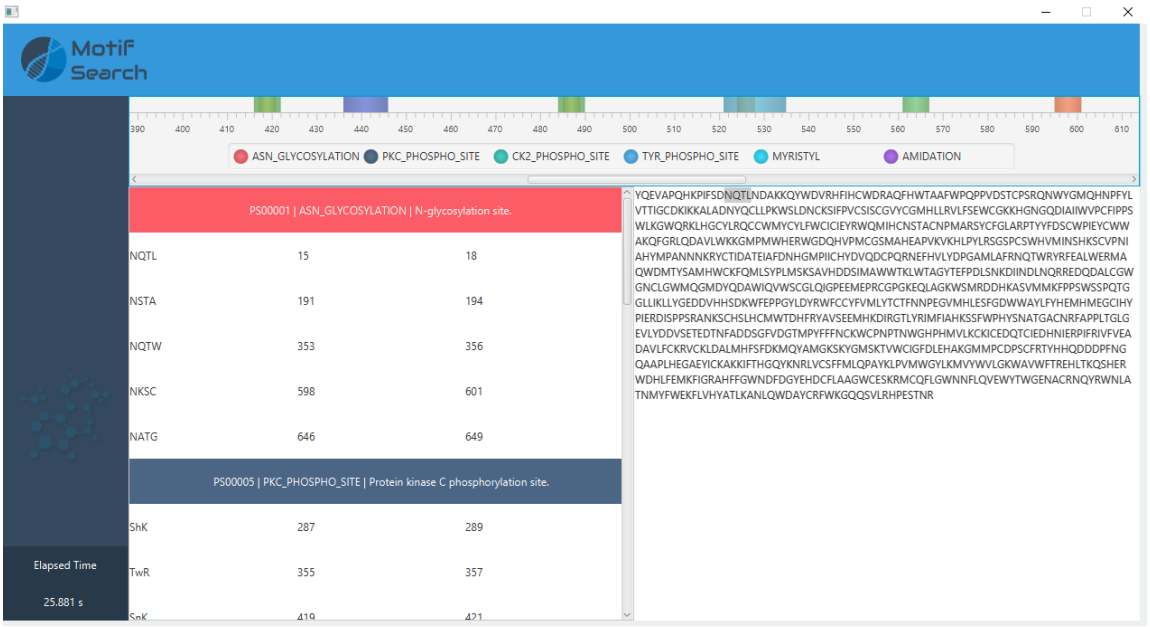


Figure 22: Results of graphical view

5.5 CONCLUSION

In this chapter, we first introduced the algorithm we implemented with its package and Parameters. In addition to the implementation environment, then we presented the general architecture of the application with Some of the aspirations we aspire to it, and in order to evaluate the algorithm we presented the data used with various tests , in the end the results obtained from the algorithm and the application interfaces.

Part IV

CONCLUSION

6

CONCLUSION

6.1 CONCLUSIONS

Motif extraction is one of the most popular problems, which has many applications. It is the process of identifying meaningful patterns in DNA sequences, DNA, or proteins. Motives vary in length, position, repetition, direction and rules. Finding these short sequences (motifs or signals) is a fundamental problem in molecular biology and computer science with important applications. Building the best tools to solve this problem helps biologists resolve many biology issues.

Analyzing and extracting the data represented by the text is an important task in the bioinformatics ,the nature of this field has made it interesting for scientists to implementing and improving Pattern matching algorithms with a view to the development of computational biology, A great number of algorithms has been proposed in the relative literature for inferring motifs in biological sequences. The majority of algorithms relies on either statistical or string machine approaches for solving the inference problem.

In our work we present the basic information that we must first understand for any computer scientist who wants to build algorithms that can help solve biological problems. We also present some of the works that have been addressed in the context of extracting the stimulus, with an explanation of how they work.

A many string matching algorithms has been lunched in context of extracting motifs from biological sequence whether the exact or the approximate , in this work we chose an exact string matching algorithm based on perl language and depend on prosite data base , it is improved version of the web-based tool for detecting PROSITE signature matches in protein sequences, we have converted it into a tool implemented in an offline environment and in graphical view instead of the cmd view As well , we worked on design the user interface that contains the basic commands to control the features of this algorithm.

In addition, our experimental results show high accuracy and good running time in having chains of different lengths, so we have reached to solve the problems that led us to do this work. As future work we aim to make this algorithm work with many databases instead of Prosite alone to make it more accurate and for richer results .

BIBLIOGRAPHY

[1] (1985). FASTA format. https://en.wikipedia.org/wiki/FASTA_format. Accessed:2018-03-26.

[2] (1988). Prosite. https://en.wikipedia.org/wiki/PROSITE. Accessed:2018-03-26.

[3] (2000). Sequence motif. https://en.wikipedia.org/wiki/Sequence_motif. Accessed:2018-04-26.

[4] (2002). cmd.exe. https://en.wikipedia.org/wiki/Cmd.exe. Accessed:2018-03-26.

[5] (2002). Phylogibbs algorithm. https://www.imsc.res.in/~rsidd/phylogibbs/phylogibbs_algorithm_7.html. Accessed:2018-03-26.

[6] (2005). Perl. https://whatis.techtarget.com/definition/Perl. Accessed:2018-03-25.

[7] (2006). Command prompt: What it is and how to use it. https://www.lifewire.com/command-prompt-2625840. Accessed:2018-03-28.

[8] (2006). Protein classification: An introduction to embl-ebi resources. https://en.wikipedia.org/wiki/MaMF. 2018-03-25.

[9] (2010). DNA Annotation. https://en.wikipedia.org/wiki/DNA_annotation. Accessed:2017-12-23.

[10] (2010). Gene Prediction. https://en.wikipedia.org/wiki/Gene_prediction. Accessed:2018-04-20.

[11] (2014). Definition, examples of motifs in literature. https://writingexplained.org/grammar-dictionary/motif. Accessed:2018-04-26.

[12] (2014). Pattern Recognition. https://en.wikipedia.org/wiki/Pattern_recognition. Accessed:2018-04-21.

[13] (2015). Protein classification: An introduction to embl-ebi resources. https://www.ebi.ac.uk/training/online/course/introduction-protein-classification-ebi/what-are-protein-signatures/signature-types/what-are-. Accessed:2018-04-26.

[14] (2016). Bioinformatics. https://en.wikipedia.org/wiki/Bioinformatics. Accessed: 2018-03-15.

[15] (2016). Definition of Bioinformatics. https://www.medicinenet.com/script/main/art.asp?articlekey=16836. Accessed:2018-04-20.

[16] (2016). Ngs file formats. https://bioinf.comav.upv.es/courses/sequence_analysis/sequence_file_formats.html. Accessed:2018-03-26.

[17] (2016). What is Systems Biologys. https://www.systemsbiology.org/about/what-is-systems-biology. Accessed:2018-03-15.

[18] (2017). Introduction to Molecular Biology. https://di.uq.edu.au/community-and-alumni/sparq-ed/cell-and-molecular-biology-experiences/dna-restriction-and-electrophoresis/introduction-molecular-biology. Accessed:2018-04-20.

[19] (2017). Protein motifs, patterns and profiles. http://homepages.cs.ncl.ac.uk/anil.wipat/home.formal/Bioinfoweb/section3/aboutMotifs.htm. Accessed:2018-03-26.

[20] (2017). Why is java preferred to other languages as a building block? https://www.techopedia.com/2/28705/development/programming-languages/why-is-java-preferred-to-other-languages-in-building-technological-blocks. Accessed:2018-05-11.

[21] (2018). Genetics vs Genomics. https://www.jax.org/personalized-medicine/precision-medicine-and-you/genetics-vs-genomics. Accessed:2018-04-20.

[22] (2018). Scanprosite. https://prosite.expasy.org/scanprosite/scanprosite_doc.html. Accessed:2018-05-09.

[23] (2018). The perl for ms windows. http://strawberryperl.com/. Accessed:2018-05-11.

[24] ABDUL-RAHMAN.S (2017). *Binformatics and its Importance*. online.

[25] Ammar, K, A. (2016). Computational methods of sequence alignment.

[26] Ana .T, F. (2006/2007). *Bioinformática*. Eng.Biomédica.

[27] B, L. (2016). *DISCOVERY AND EXTRACTION OF PATTERNS IN BIOLOGICAL SEQUENCES*. PhD thesis.

[28] Charles.M, J. (2017). *Biological Molecules*.

[29] Christian Lovis, M. and Robert H. Baud, P. (2000). Fast Exact String Pattern-matching Algorithms adapted to the characteristics of the medical language.

[30] Dick de Ridder, Jeroen de Ridder, M. J. T. R. (2013). Pattern Recognition in Bioinformatics.

[31] Edouard de Castro, 1, C. J. A. S. . A. G. . V. B. . P. S. L.-G. . E. G. . A. B. . . and Hulo1h, N. (2006). Scanprosite: detection of prosite signature matches and prorule-associated functional and structural residues in proteins. *PMC Journals*, page 5.

[32] G, M. (2006). *BIOINFORMATICS Introduction*. Yale University gerstein-lab.org/courses/452, last edit in spring '09 edition.

[33] GOU, M. (2014). Algorithms for String matching.

[34] Hon LS1, J. A. (2006). A deterministic motif finding algorithm with application to the human genome.

[35] Jalil, A. (2015). *Bioinformatics*. Number 2.

[36] Jingyi Yang, Jitender S. Deogun, Z. S. (2005). A new scheme for protein sequence motif extraction. page 8.

[37] Lindsey Barron, A. J. R. B. (2017). Applications of Bioinformatics. *Journal of Proteomics Bioinformatics*.

[38] Lounnas. B, Bouderah. B, M. (2017). Biological Motif discovery Algorithm based on mining tree Structure. *International Journal of Computer Applications*.

[39] Mahin Gh., H. K. (2015). Comparative Genomics. *International Human Genome Research Institute*, 1.

[40] Mourad .E, Julie .D, A. E. H. M. S. (2017). Motif Discovery in Protein Sequences. 17.

[41] Neil, J. (2015). Bioinformatics Approaches for Gene finding. 1.

[42] of Medicine, U. N. L. (2018). *Your Guide to understanding Genetic condition*.

[43] Phil, M. . (2015). Genome Annotation. 1.

[44] Pissis, S. (2014). Motex-ii: structured motif extraction from large-scale datasets. *BMC Bioinformatics*, 15(1):235.

[45] .R, G. (2011). Sequence Similarity Method.

[46] R, H. T. (2004). Introduction to Molecular biology and bioinformatics-methods for functional genomics.

[47] R, C, T. L. K. H. C. C. W. L. C. S. O. Y. K. S. (2012). Exact and Approximate String Matching Algorithms.

[48] Rich.A (2009). The Era of RNA awakening: Structural biology of rna in the early years. 42.

[49] Sajida Mohammad, S. N. (2013). *Pattern Recognition and its Applications*.

[50] Sarah .K, Wooller, G. B.-H. X. C. Y. A. F. M. P. (2017). Bioinformatics in translational drug discovery.

[51] Thinley K, B, A. T.-L. W. W. L. H. (2015). Perl computer programming language. *Encyclopaedia Britannica*, page 5.

[52] Vinita Dutt, Vikas Chaudhry, I. K. (2012). Pattern Recognition: an Overview. *American Journal of Intelligent Systemss*.

[53] Z., C. Y. Motifs and methods for generating motifs. page 8.

[54] Z., R. *Biological Medical Informatics*. SAN DIEGO STATE UNIVERSITY. http://informatics.sdsu.edu/bioinformatics/.

[55] Zhang, Y. and Zaki, M. J. (2006). Exmotif: efficient structured motif extraction. page 92.

[56] Zhu, B. and ., M. N. (2012). Online Handwritten chinese/japanese character recognition. *SIGACT News*, page 18.