12-2011

ALIGNMENT-FREE METHODS AND ITS APPLICATIONS

Ramez Mina
University of Nebraska at Omaha

ALIGNMENT-FREE METHODS AND ITS APPLICATIONS

A Thesis

Presented to the

Department of Computer Science

And the

Faculty of the Graduate College

University of Nebraska

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

University of Nebraska at Omaha

by

Ramez Mina

December 2011

Supervisory Committee:

Name: Prof. Hesham Ali

Name: Prof. Dhundy Bastola

Name: Prof. Guoqing Lu

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

ALIGNMENT-FREE METHODS AND ITS APPLICATIONS

Ramez Mina, Master of Science

University of Nebraska, 2011

Advisor: Hesham Ali

Comparing biological sequences remains one of the most vital activities in Bioinformatics. Comparing biological sequences would address the relatedness between species, and find similar structures that might lead to similar functions.

Sequence alignment is the default method, and has been used in the domain for over four sdecades. It gained a lot of trust, but limitations and even failure has been reported, especially with the new generated genomes. These new generated genomes have bigger size, and to some extent suffer errors. Such errors come mainly as a result from the sequencing machine. These sequencing errors should be considered when submitting sequences to GenBank, for sequence comparison, it is often hard to address or even trace this problem.

Alignment-based methods would fail with such errors, and even if biologists still trust them, reports showed failure with these methods.

The poor results of alignment-based methods with erratic sequences, motivated researchers in the domain to look for alternatives. These alternative methods are alignment-free, and would overcome the shortcomings of alignment-based methods.

The work of this thesis is based on alignment-free methods, and it conducts an in-depth study to evaluate these methods, and find the right domain's application for them. The right domain for alignment-free methods could be by applying them to data that were subjected to manufactured errors, and test the methods provide better comparison results

with data that has naturally severe errors. The two techniques used in this work are compression-based and motif-based (or k-mer based, or signal based). We also addressed the selection of the used motifs in the second technique, and how to progress the results by selecting specific motifs that would enhance the quality of results.

In addition, we applied an alignment-free method to a different domain, which is gene prediction. We are using alignment-free in gene prediction to speed up the process of providing high quality results, and predict accurate stretches in the DNA sequence, which would be considered parts of genes.

Keywords: sequence alignment, alignment-free, compression complexity, Lempel-Ziv complexity, Kolmogorov complexity, biological signals, motifs, phylogeny, comparative genomics, and gene prediction.

In memory of my mother, Marta Guirguis.

ACKNOWLEDGMENTS

My sincerest thanks to Professor Hesham Ali, an advisor and a mentor, for his continued support, encouragement, guidance and inspiration, which made this work complete.

My sincere gratitude also to professor Dhundy Bastola and professor Guoqing Lu, for serving on committee and providing detailed feedback on improving this thesis.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

1.1 Bioinformatics and computational biology

Research in biology is very important for human life. Discoveries in this domain help physicians to improve their treatments techniques, and provide better quality for the health care domain. The primary research has been done in wet lab over hundreds of years. Biomedical engineering along with other similar disciplines came to the domain to speed up the research, and improve the quality of the treatment. Computational biology specifically came to provide biologists with fast and accurate tools to look into the human DNA sequences, and provide analysis for the digital format of the sequences. These tools would analyze the sequences and provide evidences for natural phenomena, where these tools are computational and are applied to digital data. Types of tools would be to find specific patterns or signals in the sequences, searching for stretches that could be genes, or identifying relationships between species. The enhancements to provide such tools happen in another discipline called Bioinformatics. This science integrates biology with other sciences to provide solid tools for the biologists, and give them faster and more accurate results compared to those obtained in a regular wet lab. Basically computational biologists take advantages of the bioinformatics tools to help them enhance their biology research.

1.2 Sequence analysis

Sequence analysis is an important field of bioinformatics domain, and it deals with the analysis of sequences. Sequence analysis is the heart of bioinformatics; and it is essential in almost any biology research. A biologist would undergo major analysis on the sequences in preference, before they decide about applying their methodologies. Sequence analysis deals with both; DNA and protein sequences, and it is mainly a tool for data mining in the sequences; to obtain information that would be essential in the decision needed for the biologist. This information would let the biologist proceed on with their research, and apply the right solution for the problem in favor. An example of this process would be a research to find new signals in the sequences, so a biologist would apply a motif finding approach, to find words with potential strength, and then compare these words to others from other sequences. This would allow them to conduct a deeper research on the reported ones and measure their biological strength. The first step for such a work would be applying some sequence analysis tools, like motif-finding tools, before proceeding to solve the actual problem.

Another example would be gene prediction, and finding genes is very important problem, especially for drugs companies. The drugs' industry is built on understanding the nature of genes, and their main research is to search for stretches in the DNA sequences. These stretches might be genes, where one way to find genes in sequences, would come from the fact that similar species carry similar functions. The first task to carry on such research; is to find similar species that might carry similar functions and structures, hence applying sequence comparison tools to find such groups of species is the first step in this kind of research, which would need sequence comparison.

1.2.1 Sequence comparison and sequence alignment

Sequence comparison deals with comparing biological sequences. That could happen in either pair-wise fashion or multiple sequence comparison fashion. By pair-wise; we mean comparing two sequences and find the relationship between them. So if we need to compare 3 sequences together, we compare each 2 separately, and then relate the 3 sequences accordingly. While multiple sequence comparison is also based on pair-wise comparison, but the end result deals with the relationship of several sequences at once. Sequence alignment is the default method by choice; it came to the literature in 1970 as a digital solution to solve the problem of comparing sequences. The basic interpretation for sequence alignment; is how many steps are needed to convert one sequence to another. This process involves aligning two similar nucleotides (a nucleotide is the basic unit of a DNA sequence), aligning dissimilar nucleotides, or inserting a gap in one sequence. The cost of this process would provide a numerical value; which would represent the similarities between the sequences.

1.2.1.1 Limitations of sequence alignment

Sequence alignment came in the frustration time, when there was no computational tool to speed up the process of comparing biological sequences. Biologists were really excited to see a fast tool to provide them with numerical results; which they can analyze; and build a sense on how closely related are the sequences. With the new advancement in the domain, reports showed failure of sequence alignment, especially with longer sequences that were the results of the new sequencing technologies. Also the tool became relatively slow, and would take long time before it reports results. Improvements and algorithmic solutions were provided to deal with the speed issue, and there was success in this

direction to some extent, but the problem of quality still existed, and biologist started to request other solutions to the problem.

The limitations of alignment-based methods come mainly in two categories:

1. Accuracy, although the methods provided the biologists with graphical representation, but in many cases it forced the sequences to be aligned, even if they are closely related. This shortcoming would mislead the biological research, and provide incorrect results.
2. Complexity, although alignment-based methods were fast when they first came; as the available sequences length were short at this time, it started to be considered slow with more longer sequences, and it is now unreliable method with whole genomes. Alignment-based methods also failed with longer sequences as they consume large amount of memories.

1.3 Alignment-free methods

Alignment-free methods came to the literature in the last two decades, as a solution for the shortcomings of the sequence alignment methods. These methods are not based on alignment, but they are built on different concepts and computational foundations. Sometimes these methods integrate biological facts to provide better way to compare the sequences, some of them are based on statistical and stochastic models; like Hidden Markov Model, some of them are based on finding special signals in the sequences like shortest unique substring, or find genes that would be considered in a weighing function and would indicate the relationships between species. But the major two categories that alignment-free methods fall in; are compression-based techniques and k-mers methods.

1.3.1 Compression-based methods

Compression-based techniques came to the literature recently, and they got a lot of attention from researchers because of their heuristic speed, and because of the concept

they were built on. They are based on compression complexity; which basically deals with the similarity between the compared strings. The basic idea falls in the concept of compressing one string in terms of another string, so if an algorithm is capable of compressing string S_1 in terms of string S_2, then the two strings have similarities, and the compressibility ratio would provide an estimate of how closely the two sequences are. That would happen mainly by appending S2 to S1, and compress them as one string, and then append S_1 to S_2 and compress them as one string, and use the resulting values in some mathematical equations to measure the relatedness between the sequences. That would provide a numerical value that is also normalized for the relatedness between two strings, by normalized we mean that the value is scaled in a range of 0 to 1. This concept was borrowed to the sequence comparison domain, and two strings would be replaced with two biological sequences, and lots of experiments and tests were applied to measure the viability of these methods and how good results they would provide. In this work; two different compression complexities were used, Kolmogorov complexity and Lempel-Ziv complexity. Kolmogorov complexity deals with different compression algorithms, and uses the resulting compression values as the seeds for its unique equations, giving by this flexibility for the researcher to use their own compression algorithm. While Lempel-Ziv has its own compression technique that is a dictionary-based and would provide special class of unrepeated parsed words. The number of these words would express the complexity, and would be the main seeds for a group of equations provided by the technique, and the results of these equations would provide numerical values to express the relatedness between 2 sequences as well. Besides testing this hypothesis, our work was beyond the point of testing the method, and went beyond that to find applications for

these methods, the application is mainly for sequences that have natural changes, and these changes would mislead alignment-based methods, and would make it reports uncertain results. Examples of these changes would be sequences with mutations that were developed over years, sequences with high repetitions of subsequences, sequences that are not complete, sequences that are not complete and were assembled from incomplete fragments, or sequences that are incomplete and were assembled from incomplete fragments that were out of order.

We applied the methods on such erratic datasets to evaluate the methods performance in such cases, and to report them for better use when datasets are highly subjected to errors.

1.3.2 k-mers methods

K-mers methods is the second main category for sequence comparison, it deals mainly with the probabilistic model for all possible words of length k. For example all possible words of length 3 for a DNA sequence would be 64 (4^3), where 4 stands for the number of letters in alphabet (4 nucleotides for a DNA sequence) and 3 is the length of the word. The algorithm then would scan the sequence for all the occurrences for each word, and reports its probability in a vector for each sequence. Then it applies a distance measure between each pair of vectors to be an indication for the relatedness between two sequences. The algorithm could be applied to protein sequences as well.

The algorithm has several improvements that deals with using different values for k, and even append several vectors representing different values for k in one vector. The main contribution to the algorithm was in providing lots of suggested measures for the distances between the vectors were proposed; and these distance measures would maximize the quality of the results.

In this work we are using a different approach to use the k-mers concept, and this approach considers the occurrence, frequency and order of the k-mers, it also consider the usage of random signals that might carry special information in the sequences, and signals that have biological relevance; like restriction enzymes or signals obtained from coding regions. The main motivation for such an approach; is based on the fact that some signals would differentiate between sequences, and sometimes such signals would be unknown, and hidden within the sequences, hence integrating an algorithm to reveal their strength (and not revealing the signals themselves) would lead to better results for sequence comparison. That would happen by running the approach using random words of length k, and measures if these random signals would provide better results than those of alignment-based or not. If the results are better than alignment-based methods, then this would prove that there exist some signals with better strength. Another issue was addressed in the k-mers work, is to use signals that have biological relevance, and measure if they would provide better measurement for sequence comparison; relative to measurements obtained by alignment-based methods. Such words could be restriction enzymes, or words that occur within a region of the DNA with biological relevance, like coding region (CDs).

1.3.3 Sequence comparison assessment

Evaluating sequence comparison was never an easy task, the default method is to cluster the species based on the reported distances, draw the phylogenetic tree of these distances, and leave it to the biologist to decide the correctness of the resulting tree, which in turn would reflect the correctness of the reported distances. Unfortunately this method lacks accuracy, especially with a big number of species, which would make the biologist not

able to read the tree and give an in-depth report about its quality. In addition to that, this method would not provide a numerical value to be used to evaluate the correctness of several trees, for example if we are evaluating the results of 5 different methods, and 3 of them sound to have very close tree's topology, it will be difficult for the biologist to report the best of them, but if there is numerical values for the correctness of each tree, then a sorted order of best trees would be reported, which in return would reflect the quality of each experiment and the used approach.

We evaluated the results using a modified approach of the above, and we overcome the shortcomings of this one, by first compare the resulting trees of our methods; to a reference tree that would be provided by the biology society. This tree has the correct topology for a group of species, and it is usually provided by a trusted organization like NCBI, this tree has the correct topology for a group of species. Secondly we measure the resulting trees computationally and not using visual measurement, and this computational measurement would provide values to be considered for evaluating the approaches. The basic idea is to first run the experiment for the method/technique in favor, report the resulting distances of this technique, use these distances to cluster the species and construct their phylogenetic tree, then we compare the resulting tree to a reference tree that has the gold standard topology. The comparison occurs computationally using an algorithm called path-length-difference. This algorithm measures the relative positions of the species to each other, and evaluates how much deviation happened in the resulting tree compared to the reference tree. This would occur by giving a penalty for any species that move from its original position, and report the penalties as numerical values. These numerical values would be considered as the measurable values for the difference

between the resulting tree and the reference tree. Finally the evaluation for the resulting trees and their differences to the reference tree would be reported as a way of comparing different sequence comparison approaches. For example when we compare Lempel-Ziv distance measures which are compression-based, to Multiple Sequence Alignment (MSA) which is alignment-based, this would happen by generating the trees of all the proposed methods, measure their distances to the reference tree, and evaluate which distances are smaller. That would reflect which tree is closer to the reference tree, which in turn would be an indication for better results, as the good results indicate which trees are closer to the reference tree.

1.4 Comparative genomics and gene prediction

Comparative genomics is an application of sequence comparison, it deals with comparing the entire genomes, and we are considering it in this wok to predict genes. The main motivation of using comparative genomics to search for genes in closely related genomes, which would share similar structures and functions, these structures would probably be the proposed genes (sometimes the similarities are not usually genes). So in case of unknown genes, the similar structures might be unknown genes; and hence using comparative genomics would lead to find new genes in the sequences.

1.4.1 Using comparative genomics to find genes

The proposed algorithm works by first splitting the sequences to relatively short fragments, and these fragments would overlap, and then use a fast and an accurate alignment-free technique to compare these fragments, and report the closely related fragments to be considered for the next step. The next step would be local alignment, to

measure how closely related are these fragments, and finally report the fragments that pass the proposed thresholds, and combine those which are in order to have them as predicted genes.

1.4.2 Gene prediction assessment

Evaluation of the method occurs by using the sensitivity and specificity measures, where sensitivity reports how many true positive divided by true positive and false negative, and specificity reports how many true positive divided by true positive and false positive, where true positive are the number of predicted nucleotides that happen to be within a gene region, false negative is the number of unpredicted nucleotides that happen to be within a gene region, and false positive is the number of predicted nucleotides that happen not to be within a gene region.

Understanding the work of this thesis needs understanding of some foundations in biology, and the following subsection would discuss these foundations briefly.

1.5 Overview of biology

The work of this thesis is considered to be “biological sequence analysis”; hence it is important to understand some basic definitions and concepts of biology, which are needed for the reader to understand both the biological terminology, as well as the nature of the sequences and their problems that motivated the work of this thesis.

The following brief headers cover the definitions and the nature of genomes, genes, DNA, gene expression, transcription, gene structure, splicing, translation, synteny and homology.

1.5.1 Genomes

Genomes are the genetic material of a species, hence it is the total amount of DNA in the entire cell. It occurs as a set of chromosomes, and each chromosome has a long chain of DNA that is highly condensed.

1.5.2 DNA

DNA sequences are usually a pair of anti-parallel chains, held together by complementary base pairs that form the double helix. Each DNA sequence is composed of a molecule called a nucleotide, which is composed of hydrogen, oxygen, nitrogen, carbon, phosphate and a base of four bases. The four bases are Adenine, Guanine, Cytosine and Thymine.

The structure of the DNA model reveals some information that would be needed in the sequence analysis.

- The nature of the sequences could be interpreted as a digital nature, by considering the main difference between nucleotides, which is the base. The bases are denoted digitally as A, G, C and T. It is worth mentioning that T is replaced with U (for Uracil) in an RNA sequence.
- The two chains are complementary to each other, A faces T, C faces G and vice versa, hence to analyze the sequences, a mean of definitions is needed for forward or positive (+) and reverse or negative (-) strands, and elements that specify the sequences like genes, exons and introns. Although genes are transcribed from both chains, most research deals with only the forward sequence, which is also known as 5’ to 3’.

- The complementary nature allows computers to just deal with the forward sequence, as the reverse one could be digitally constructed using the forward.
- The forward sequence could be read as from 5' to 3', where the 5' is the upstream region, and 3' is the downstream region.

1.5.3 Gene Expression: from DNA to RNA to Protein

Gene expression is a process in which the gene in form of DNA is converted to protein. It is also known as the central dogma of biology, this process includes transcription, splicing and translation steps, and each step is triggered by some signals. In general the process of gene expression starts with the DNA (gene), and it converts to RNA and finally it becomes a protein sequence. Figure 1 shows the central dogma flow.

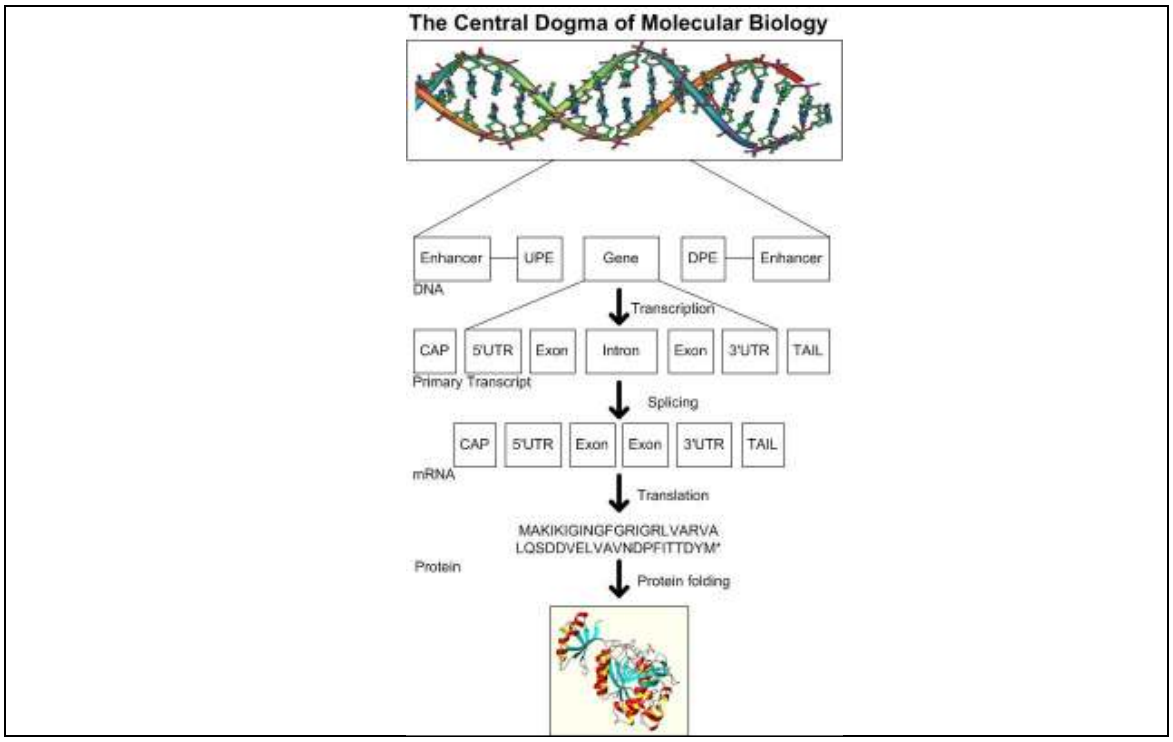

Figure 1 Central dogma of molecular biology.

1.5.4 Transcription

Is the copying of the gene sequence in form of DNA (template strand) to RNA (pre-mRNA). Transcription starts when an upstream region of the gene (promoter region) is activated (bound) by transcription factors, and these regions control and initiate a gene transcription from either a forward or reverse strand. The strand which gets transcribed is called the template or sense, and the other is called nonsense or antisense strand.

When analyzing mRNA, cDNA or EST data, the mRNA to be translated would be identical to the coding strand, where coding refers to translation and not transcription. This means that mRNA is transcribed from the strand that has its complementary sequence.

There are three main types of transcript data:

1. mRNA: messenger RNA.
2. cDNA: a double-stranded copy, usually a fragment, of an mRNA molecule
3. EST: expressed sequence tag. A short single-pass sequencing of a cDNA clone. It is typically a fragment from the 5' or the 3' end of the cDNA.

1.5.5 Gene Structure

In Eukaryotes, genes are short stretches of DNA within a genome of peculiar and discrete structure, and gene prediction techniques take advantage of this structure to predict genes. The main characteristics of this structure to consider are:

- Coding and non coding exons (UTRs)
- Introns
- Translation start site (ATG)
- Splice sites (GT, donor and AG, acceptor)

- Translation termination site (STOPs: TAG, TGA and TAA)

1.5.6 Splicing

Splicing is the process where introns are removed from the RNA, and only exons would remain. Splicing are denoted by splicing signals like GT (donor) and AG (acceptor) in the intron region, and are used to delimit exon-intron boundaries, hence exons (whether coding or non-coding) are joined together as an open reading frame from 5' to 3'.

1.5.7 Translation

The mature mRNA sequence is translated into a protein; the process is guided by signals along the mRNA sequence to find the right open reading frame (ORF), hence the process would start, and would be terminated by a stop signal (stop codon).

1.6 Structure of this thesis

The rest of this thesis is as follows, chapter 2 presents the problem statement, chapter 3 is overview on the literature review, chapter 4 presents the compression-based methods, and the results of applying them to datasets with different level of errors, chapter 5 presenting a modified method of using k-mers, and application of this method on genomic datasets using random motifs or motifs that have biological relevance, chapter 6 discusses the use of comparative genomics in gene prediction, and present the obtained results, chapter 7 is the conclusion and future work.

CHAPTER 2

Problem Statement

2.1 Sequence comparison importance

The importance of sequence comparison comes in every aspect in biology, as it is a necessary step to solve most of the biological problems. Sequence alignment is the main tool to solve this problem, as biologists use this method to compare and align sequences and measure their relatedness, as the method provides an easy graphical tool to visualize the aligned nucleotides. This trust for sequence alignment was developed as it was the only computational method for long time, and biologists trusted it, especially with its graphical representation.

2.1.1 Limitations of sequence alignment

But sequence alignment has limitations; these limitations result in failure with sequences of erratic nature. This erratic nature would be mutation, inversion, translocation, repetition or sequencing errors. Such errors would mislead the results of sequence alignment, and would report misunderstanding of the relationships between species for the researchers. In fact it might reports dissimilarities between species, while they are similar and their similarity is not recognized by sequence alignment. Hence the urge need for alternative methods to catch such similarities comes to the domain, and researchers are looking for alternatives to overcome the shortcomings of sequence alignment. To do so, a need to understand the nature of the errors is very important, and an extensive

research and lookout for methods that would address these errors, and provide alternative solutions is very important.

2.1.2 Features of errors in biological sequences

Biological sequences undergo evolution events over time; these events could be mutation, inversion, translocation, repetition, also the sequences would be subjected to sequencing errors. It is very important for the reader to know that if events exceed certain limit, then the species would evolve, hence our focus is on events that would not cause evolution of the species.

Mutation events happen over years in an amount that would still maintain the functionality of the sequence. They happen mostly as point mutation, where a nucleotide changes to a different one in the same position. Mutations happen also as deletion of a nucleotide, or as insertion of a new nucleotide.

Inversion is an event that sequences undergo, and it is basically an inversion that happens to a substring in the sequence, but still this inverted substring would carry the needed information for the sequence to maintain their functionality. Translocation is the transfer of some substrings from their original locations; to different locations within the sequences. Repetition is the process where a substring would copy itself to another location; this location could be after the same position of the original substring, or in a different location.

Sequencing errors are errors that happen from the sequencing machine, and those might be incomplete sequences, which are composed of fragments in or out of order.

Looking deeper into the nature of these events and errors, we would get a sense on how poor sequence alignment would perform, and that a different method which would address and catch these errors would be needed.

2.1.3 Addressing a proposed solution

The previous discussion on the errors would lead us to look into alternative algorithms and techniques, which would overcome the erratic nature of the sequences. This nature would consider mutations, inversion, translocation, repetitions and sequencing errors. Hence the proposed technique should not look at the sequences with their order, but should go beyond that and address the similar structures, even if they are not in order (because of translocation), or do not carry the exact same structure (because of the mutation), or are inverted, or have repetitions that would be extra unneeded information. One or two main approaches needed to be addressed by this algorithm, the first is to have the algorithm looking for these errors, and compression would address such errors, the second is an algorithm that would look for special signals carried by the sequences, and these signals would conserve the species even if they have undergone some errors, and that would be using k-mers approach.

2.1.4 Alternatives for sequence alignment

Compression would overcome such shortcomings, as compression algorithms would look for similar structures within strings, and would consider little changes that could be mutations. It will also catch the repetition of strings; consider them by ignoring them in the final output, and as compression looks at the sequences in a linear way, it will also address any translocation of the sequences. In general, an efficient compression algorithm

would catch such errors, and the compressible output of two compressed strings would indicate the relationship between them.

On the other hand, by considering special signals in the sequences, we would identify the relationships between species, as some signals are consistent and needed for the species to maintain their functionality. These signals don't change over time, and they might be hidden within these errors, and a good algorithm would reveal the similarity between species based on these signals.

A k-mers approach is suggested in this work, to address the relatedness between species based on mutual signals. The main motivation in this work; is to take advantage of unknown signals to address the relatedness between them, this could be done by applying all possible signals of length k, random sets of them, or signals with biological relevance.

2.2 Gene prediction problem

Gene prediction is an important problem for biologist, and it provides a major contribution to the drug industry. But gene prediction is not an easy problem to solve, as reports showed poor results with several tools. The main problem in searching for genes is that the nature of DNA is very random, and the DNA mechanism is not known. In fact the functionality of a lot of subsequences for the DNA is not known. Although a lot of research has been applied to look for genes, report showed that the generated tools are domain specific, and even in some cases they fail within the same domain. Hence the need for a tool that would overcome these problems comes, and a need to address the possible features that can be used to find genes is important. Based on these features we would design the appropriate approach.

2.2.1 Features of biological sequences and comparative genomics

As the work for gene prediction is a continuation for our work on sequence comparison, we are addressing the features of DNA that would be relative to comparative genomics, and that would help in providing a model to predict genes.

The DNA sequences carry similar functions, and these functions are the results of gene expression. As these functions are similar, their protein sequence would be similar, and in turn their DNA sequences would be similar. Hence the idea of using sequence comparison would be relevant and would identify such stretches in DNA, which might be genes.

2.3 Terminology

Sequence comparison is a sub science of bioinformatics. It deals with comparing biological sequences, to find the relationship between a group of species. This relationship is represented by a metric matrix. Sequence comparison has two main categories, the first one is called sequence alignment, and deals with methods that are alignment-based, and the second is alignment-free, and this one does not use sequence alignment for comparison, but use other techniques to catch the similarities of the sequences based on pattern recognition and/or biological relevance.

Alignment-free methods have two main categories, first one is based on compression techniques, and it takes advantage of the compression algorithms' ability to search similar patterns for compression purposes. The second one is based on k-mers, which deals with signals within the sequences, and find the relatedness between species based on these signals.

Compression techniques use compression complexity to identify the relationships between species, and this complexity is the main tool to identify similarities between species. The main two compression complexities are Kolmogorov and Lempel-Ziv complexity. Kolmogorov complexity uses any compression algorithms, and takes its compressibility value as the seeds for proposed equations. The results of these equations would be the normalized distances between species. While Lempel-Ziv complexity has its own compression algorithm, and its results would be the seeds for a group of proposed equations to evaluate the distances between the sequences. In both complexities, a measure of compressibility for each sequence, and each sequence appended to the other sequence is needed.

The k-mers method is similar to the use of a group of motifs, to identify the relatedness between species. By Motif, we mean a special word that is composed of a group of nucleotides, and might have special biological nature. These motifs works as the main source of signals to trigger the similarities between species, and these signals could be all possible motifs of length k, some random sets of motifs, or sets with biological relevance, like restriction enzymes which cut the sequences in specific positions, or motifs from biological relevant regions like coding regions. Restriction enzymes are special structures of DNA sequences, that cut the sequences in specific positions of the sequences, and Coding Regions or CDs are the regions of DNA that carry genic information.

The k-mers method transfers the nucleotide level sequences to motifs sequences, and a comparison between the sequences happen using some comparison methods, like Longest Common Subsequence or a compression technique.

The resulting pair-wise distances are the scores needed to construct a phylogenetic tree, a tree that would show the relationships between species. This relationship is presented as a topology with the branch length between species and ancestors. There are few datasets for species that their right topology are known, and would be considered as the gold standard trees for datasets.

The results of the alignment-free techniques need to be compared to alignment-based, to measure their performance against the well known method; hence the use of multiple sequence alignment scores to build a phylogenetic tree is important. Multiple sequence alignment is based on sequence alignment, but it considers all the sequences together for alignment, and provides a scoring matrix to be used as the main seeds for phylogeny.

The main reason to use phylogeny in this research; is to find how close would be the resulting trees to the gold standard tree. That would reflect the quality of the resulting scores from other methods. Two famous phylogeny algorithms were used in this work, UPGMA and Neighbor-Joining (NJ).

Back to the discussion of the k-mers (motifs) method, and after identifying the signals in the sequences, a way of comparing these signals is needed. The use of longest common subsequence algorithm for comparison is in favor, where this algorithm would provide a numerical value that represents the relationship between the species, and this value represents to how many mutual signals that are in order between the two sequences.

Local alignment is a method of sequence alignment. As its name explains, it deals with defining the best local relationship that could be found between two sequences, and would have the maximum score. This method is used in our work for gene prediction as the last step to verify the relationship between fragments.

CHAPTER 3

LITERATURE REVIEW

The literature review covers all the previous work that has been conducted in sequence comparison, alignment-free, compression-based, k-mers approaches and gene prediction. It is important for the reader to understand where the researchers' steps are, so they would be able to evaluate the work of this thesis accordingly.

3.1 Sequence comparison

Sequence comparison is an essential tool for biologists. In the past, and before introducing computers and computational methods, biologists used to compare biological sequences using biological observations. These observations were mainly focused on the species natural behavior. With the advent of microscopes and other analog tools, biologist started to look in-depth at the DNA and protein structures, but these methods were very slow and to a lot of extent were not accurate, and the work needed several runs for verification purposes. Hence the need for automated and computational methods was necessary.

Sequence alignment was brought to biologists by Saul B. Needleman and Christian D. Wunsch in 1970[1], this method was built on dynamic programming, and provided a graphical tool for biologists to map the relationships between species. This method has a name of global alignment, as it aligns the entire sequences, and maps the base level relationships between a pair of sequences for the entire sequences. Later in 1981 Temple

F. Smith and Michael S. Waterman [2] modified the method, and had it to find the best score for subsequences from the pair of sequences, and this score would be maximal. Biologists counted on both methods for their research and their sequence comparisons problems, but slowly errors were reported for sequence alignment, and with a deeper investigation, researchers found that sequence alignment fails with some natural events or errors in the sequences. Such natural events could be mutations, inversion, translocation and repetitions. Hence the need for other methods that would overcome such errors became demanding, these methods are not based on alignment and are called alignment-free methods.

3.1.1 Alignment-free methods

The main strength about alignment-free methods; comes from their algorithmic nature. That nature does not consider the order of the nucleotides and/or the subsequences; hence they would overcome natural errors like inversion, translocation and repetitions. They would also overcome errors like in mutation, although sequence alignment might work fine with some limited point-mutations.

Alignment-free methods came to the literature in the last 25-30 years, a comparative study by Susana Vinga and Jonas Almeida [3] shows that alignment-free methods fall mainly in two main categories. The first category is based on compression techniques [3][4][5][6][7][8][10][11]; and takes advantage of the pattern recognition searching algorithms, which compression-based techniques are built on. While the second one is based on the use of k-mers [3][12], by generating vectors of the probabilities of these k-mers, then measure the distances between these vectors using several distance measures, and the numerical result would be the biological distance between species.

In addition to these two categories, other papers in the literature discussed alignment-free methods, like using shortest unique substrings [13] or Local Decoding of Order N [14]. The basic idea behind using shortest unique substrings, is to find these shortest substrings, and use them as an identity for the genomes, and would provide a way of clustering for the species, but the researchers did not provide numerical measures.

While the basic idea of Local Decoding of Order N falls in the use of some statistics. These statistics deals with the change of a certain amount of nucleotides in a word of length *N*, and by scanning the similar words that has a little of mutation on the nucleotide level.

Unfortunately the first method did not provide a numerical value for the distances, and the second one did not use a biological fact, or at least consider any.

Beside these two methods that were provided in the literature, the main two methods for alignment-free are the compression-based and the k-mers based techniques, and we are providing more details about the work that has been done for each one.

3.1.2 Compression-based techniques

Compression-based techniques provided a new way to compare biological sequences. The concept that compression is built on is that strings would be compressible if they have repetitions, and these repetitions could be replaced by pointers which would save space. If two sequences share similar structures, and they are combined into one sequence, then this similar structure would be repetitions, and hence the new sequence would be compressible. This concept is used to identify relationships between sequences, and according to the compressibility values, the distances between species would be identified.

But the use of compression with biological sequences started for the sake of compression, and to save space; and give lighter weight for the biological data when they are transferred over networks.

When research in bioinformatics and computational biology was born, compression of biological sequences was introduced for the sake of saving space as well. The properties of DNA and protein sequences are suitable to apply compression techniques. Chen [8] stated that standard text compression tools like Compress, Gzip and Bzip2 cannot compress DNA sequences efficiently [8], while Behzadi and Le Fessant [9] discussed that DNA sequences have structures that are not random; which would make them possible for compression using only two bits. That led Chen et al. [8] to design a compression algorithm that would take advantages of the previous facts.

Compression is based mainly on compression complexities, and these complexities are the main core to use compression in comparing sequences. Abraham Lempel and Jacob Ziv introduced the concept of compression complexity in 1976 [15] which was the core foundation to introduce the LZ compression technique in 1977 [16]. Another angle and a little different concept of compression complexity was introduced by Kolmogorov [17], and his concept was independent from the compression algorithm, which means that the user can apply any compression algorithm.

We started to learn about the integrated information of the biological sequences. This integrated information comes in different forms of similarities within the same sequence, or among several sequences, and it became convenient to use compression to detect the relatedness between biological sequences. Chen et al. [8] and Rivals et al. [10] discussed that biological sequences have tandem repeats in higher eukaryotes, and multiple copies

of genes which make them relevant to the compression techniques. In addition to these properties, DNA sequences are rich with other biological features that are hidden within the sequences, and these features could be detected using compression, such features are random mutations, translocation, cross-over and reversal events.

Also Rivals et al [10] discussed how compression would address such properties, and would use these properties to provide high compression values. Which in turn would reflect the relatedness between the sequences, so by concatenating two sequences we would be able to compress them effectively if they have some common information.

Compression complexity is a powerful tool to address the relatedness between strings, as strings in general are the base for text. Compression complexities would address how much similarity does several text has in common.

In the application of this research, the strings are the biological sequences, and the complexity would address if two sequences are related according to the compressibility level of each sequence, and each sequence in terms of the other (concatenate first sequence to the second, and the second to the first). For any pair of sequences [5], we would measure the compression complexity of each sequence, also for each sequence concatenated to the other sequence. We then introduce the compressibility values to distance measures that would give us good estimates for the relatedness between this pair of sequences. Two compression complexities were used in previous research, and are used in this work, Lempel-Ziv complexity as in the work of Otu et al. [4] and Burstein D. et al. [7], and Kolmogorov complexity as in the work of Ming Li et al [6] and E. Rivals et al. [10].

This researchers' work was mainly focusing on the assessment of the methods, and how

viable they could be, but they ignored searching for a good domain for these methods, and although these methods are very effective, we believe they would be even better with datasets that have errors.

3.1.3 k-mers based techniques

k-mers based techniques were introduced to magnify the importance of hidden signals within the sequences. These signals are not known in the biology domain, but they do exist, and researchers want to take advantage of their existence, and use them to identify the relatedness between species. The basic idea for this approach is to parse the sequence into words of length k, and these words overlap with k-1 period, and then measure the probability of each word in the sequences. This would lead to a feature vector that has probabilities for 4^k cells words; each one is in a separate cell. These probabilities would identify the biological distance between species. The idea itself is basic, and the only added improvement was to consider different vectors for different values of k, and append them together to result in one vector, as was shown by Guoqing Lu et a. [12]. But the main contribution to this area occurred within advances in the suggested distance measures, as in the work of Guoqing Lu et al, they used the cosine angle between the vectors [12]. Also as in the work of Qi Dai et al., as they used Euclidian distance, cosine of the angle between the vectors, Standardized Euclidean distance, Kullback-Leibler discrepancy, a protein matrix noted as W-metric and some suggested statistical measures. They also provided novel statistical distance measures like the generalized relative entropy and gapped similarity measures [18].

3.2 Gene prediction

The work for gene prediction in this thesis; focuses on the use of comparative genomics, and is a continuation for the work that has been done before by Rong Chen et al. But there is a lot of work in the gene prediction that was mainly based on digital signal processing and other statistical and machine learning approaches, and we are summarizing the previous work in the following section.

Mahmoud Akhtar et al. used digital signal processing (DSP) models to find genes in eukaryotic [19], as they proposed several DSP models to predict genes and exons, and compared them to other existed models. They also contributed a DSP-statistical hybrid technique for acceptor splice site detection. While P.P. Vaidyanathan et al. [20] used specifically digital filters on DNA sequences to predict patterns for the codons, which would be 3 nucleotides and would be translated into proteins. T. Efstetol et al [21] used Fourier transform to detect genes in the DNA sequences based on the framework of Bayes classification. On the other hand Yang Weng et al. [22] combined several work and programs using Desmpster-Shafer theory to find evidence for gene prediction, their basic idea is to use several reliable programs, and take their results as the seeds for their approach to maximize the results. Other approaches included using resampling-based spectral analysis to improve the gene prediction process, as in the work of C.Q. Chang et al. [23], they suggested that any small improvement in the process of finding genes would contribute to the literature, as these improvement could be combined together to provide a better solution. Statistical combination and classification in terms of gene characteristics was introduced as well, as in the work of Qing Tong et al. [24]. The focus of their work was to identify special features of the genes, and identification of these

features would happen by using statistics that are applied to different gene prediction approaches. The different approaches and methodologies discussed above resulted in a group of gene prediction packages/software like genescan, genemark.hmm, HMMgene, Fgenes and MZEF [25].

Another domain for gene prediction focused on finding genes based on comparative genomics as in the work of Rong Chen et al. [25]. The main strength of this work focuses on identifying similar stretches from closely related species, and considers that these species would share similar functions, and hence would also share similar structures of DNA. Their work showed promising results.

CHAPTER 4

COMPRESSION-BASED TECHNIQUES

Comparing biological sequences remains one of the major activities in the bioinformatics domain. Sequence alignment is the default method by choice, but reports showed limitations for the method, on quality and speed levels. Hence the need for alternative methods became essential, and compression-based techniques are a main alternative for replacing sequence alignment, especially with data that suffers errors. This chapter discusses compression-based techniques and how to apply them on DNA and protein datasets, it also discusses their application to different types of datasets that might have erratic nature, like DNA datasets, protein datasets, or genomic datasets that have errors.

4.1 Background on Compression Complexities

4.1.1 Kolmogorov Complexity

For any two sequences x and y, we define the conditional Kolmogorov complexity $K(x|y)$ [9] as the shortest binary program that compute x in terms of y. We define Kolmogorov complexity of a sequence x as $K(x)$ and is also defined as $K(x|\lambda)$, where λ stands for an empty string. We also define the information distance ID between two sequences x and y as:

$$ID(x, y) = \max \{K(x|y), K(y|x)\}$$

Kolmogorov theory is a concept more than a measure; therefore it does not offer a metric value that could be used in constructing a phylogenetic tree, and provide an application of

clustering. Hence Universal Similarity Metric (*USM*) was suggested and implemented to measure the complexity of Kolmogorov.

Three practical approximations to Kolmogorov were suggested; Universal Compression Distance/Dissimilarity (UCD), Normalized Compression Distance/Dissimilarity (NCD) and Compression Distance/Dissimilarity (CD), and have the following formulas:

- $UCD(x, y) = \frac{\max\{|C(xy)| - |C(x)|, |C(yx)| - |C(y)|\}}{\max\{|C(x)|, |C(y)|\}}$
- $CD(x, y) = \frac{\min\{|C(xy)|, |C(yx)|, |C(x)| + |C(y)|\}}{|C(x)| + |C(y)|}$
- $NCD_1(x, y) = \frac{\{|C(xy)| - \min\{|C(x)|, |C(y)|\}}{\max\{|C(x)|, |C(y)|\}}$

Then $NCD(x,y) = \min\{NCD_1(x,y), NCD_1(y,x)\}$, while $C(l)$ is the length of the compressed sequence.

4.1.2 Lempel-Ziv complexity

Consider the sequence S = AACGTACC, its history [3] is defined as:

H(S) = A.A.C.G.T.A.C.C,

H(S) = A.AC.G.T.A.C.C or

H(S) = A.AC.G.T.ACC

The exhaustive history [3] is defined as the history where no substring has a repetition, and no substring can be found in the whole sequence before this substring. This means if a substring is chosen at the i^{th} position, then the sequence of characters before the i^{th} position will not contain an occurrence for the following parsed substring. By examining the histories in the previous example, the first two cannot be exhaustive histories because the ‘A’ and ‘C’ are repeated, but the third one is.

LZ complexity is defined as the least exhaustive history of a sequence and noted as c(sequence)

Example:

Consider the following three sequences:

S =AACGTACCATTG

R =CTAGGGACTTAT

Q=ACGGTCACCAA

The exhaustive histories for these sequences would be:

$H_E(S)$ = A · AC · G · T · ACC · AT · TG

$H_E(R)$ = C · T · A · G · GGA · CTT · AT

$H_E(Q)$ = A · C · G · GT · CA · CC · AA

c(*S*) = c(*R*) = c(*Q*) = 7

And the exhaustive histories for *SQ* and *RQ* are:

$H_E(SQ)$ = A·AC ·G· T ·ACC · AT·TG·ACGG·TC ·ACCAA

$H_E(RQ)$ = C · T ·A·G·GGA·CT T · AT·ACG·GT ·CA·CC ·AA

c(*RQ*) = 12 and c(*SQ*) =10

Which means that *S* is closer to *Q* than *R* is to *Q,* and that would be visualized as in the following colored sequences:

S =AACGTACCATTG

Q =ACGGTCACCAA

Q =ACGGTCACCAA

R =CTAGGGACTTAT

LZ complexity itself is not a distance measure between sequences, but instead the following distance measures are used:

Distance measure 1:

$$d(S, Q) = max\{c(SQ) - c(S), c(QS) - c(Q)\}$$

Distance measure 2:

$$d^*(S, Q) = \frac{\max\{c(SQ) - c(S), c(QS) - c(Q)\}}{\max\{c(S), c(Q)\}}$$

Distance Measure 3:

$$d_1(S, Q) = c(SQ) - c(S) + c(QS) - c(Q)$$

Distance Measure 4:

$$d_1^*(S, Q) = \frac{c(SQ) - c(S) + c(QS) - c(Q)}{c(SQ)}$$

These distances would be same as the scoring values of any sequence alignment method, and would be used in building the phylogenetic tree of the dataset.

We referred to phylogeny and phylogenetic trees in the compression complexities, as they are major tool for the evaluation and assessments of the experiments.

4.2 Methodology

The purpose of this section is to have a way to evaluate the hypothesis, which starts with designing the flow of each experiment, moving to collecting the datasets, then applying the steps for each experiment to finally evaluate our hypothesis.

4.2.1 Experimental Design

The experiments progressed in four phases (Figure 2): dataset assembly, scoring matrices compilation, construction of the phylogenetic trees, and then evaluation of the hypothesis. Lempel-Ziv-Welch and Huffman compression algorithms [26] were the seeds for

Kolmogorov complexity metrics. Lempel-Ziv complexity has its own algorithm to measure the complexity, before seeding it to a group of distance measures, and finally introduce the results as the compilation matrices for phylogeny. Lempel-Ziv complexity was implemented with a modified algorithm published by Borowska et al. [27].

After obtaining the datasets, and generating the scoring matrices using compression complexities and multiple sequence alignment, the evaluation step would come next. As the scores generated by the compression algorithms are subjective according to different datasets, we evaluated them by measuring the consistency of the constructed trees of these scores. The correctness of the topologies of these trees would be an indication for the quality of the used scoring algorithms (compression complexities), and hence evaluating the quality of these trees would reflect the quality of the compression techniques used in this research.

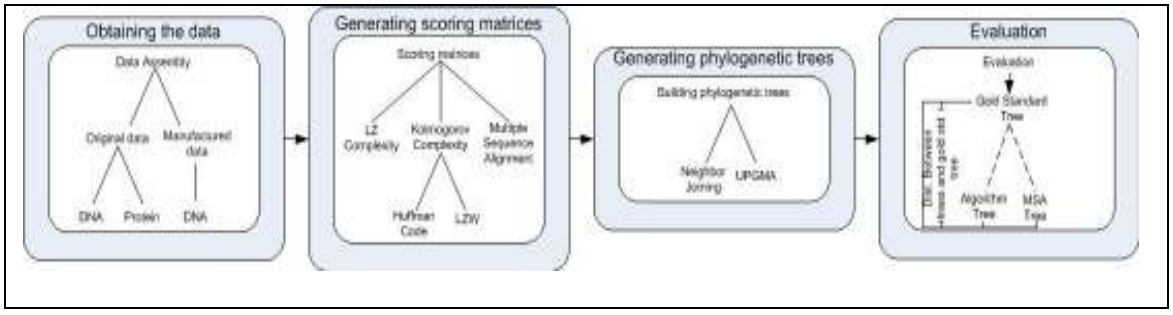

Figure 2 Experiment steps, starting from collecting data, and moving towards compiling the scoring matrices for the sequences, then clustering the results using phylogeny, and finally evaluating the correctness of the resulting trees.

4.2.2 Dataset Collection

Datasets varied according to the experiment, the first experiment used a protein dataset and a mitochondrial whole genomes dataset. The protein dataset is a Chew-Kedem data

set for 36 protein sequences, drawn from PDB entries of three classes (alpha-beta, mainly-alpha, mainly-beta), and the mitochondrial dataset is for Apostolico whole mitochondrial genomes.

These two datasets were used to test the viability of compression techniques in comparing biological sequences. The datasets obtained are as follows [5]:

CK-36-PDB: Chew-Kedem dataset of 36 protein domains, represented as amino acid sequences in FASTA format.

AA-15-DNA: Apostolico dataset of 15 species, mitochondrial DNA complete genomes.

The second dataset (mitochondrial DNA complete genomes) was used as the source dataset for the last four experiments, and several datasets were manufactured from this dataset, each new dataset was manufactured to serve a specific purpose of the experiments, and hence it has specific parameters.

The second experiment focused on having different percentage of incomplete genomes, ranging from 10% - 90% of the original genomes, with start positions of the incomplete fragments chosen randomly (Figure 3-A). The third experiment evaluated incomplete genomes assembled from separate segments, and the total length contained of 10 - 90% of the whole genomes (Figure 3-B). The fourth experiment explored genomes that are 10 - 100% incomplete fragments containing several shuffled fragments assembled together (Figure 3-C). The fragments were placed in random order using the Fisher–Yates algorithm [28]. The fifth experiment dealt with mutated sequences, these mutations were obtained with different percentages, and were point-mutations. For each experiment multiple sequence alignment was used to measure the sequence alignment values for each

dataset. MUSCLE [29] package was used to generate the multiple sequence alignment scores.

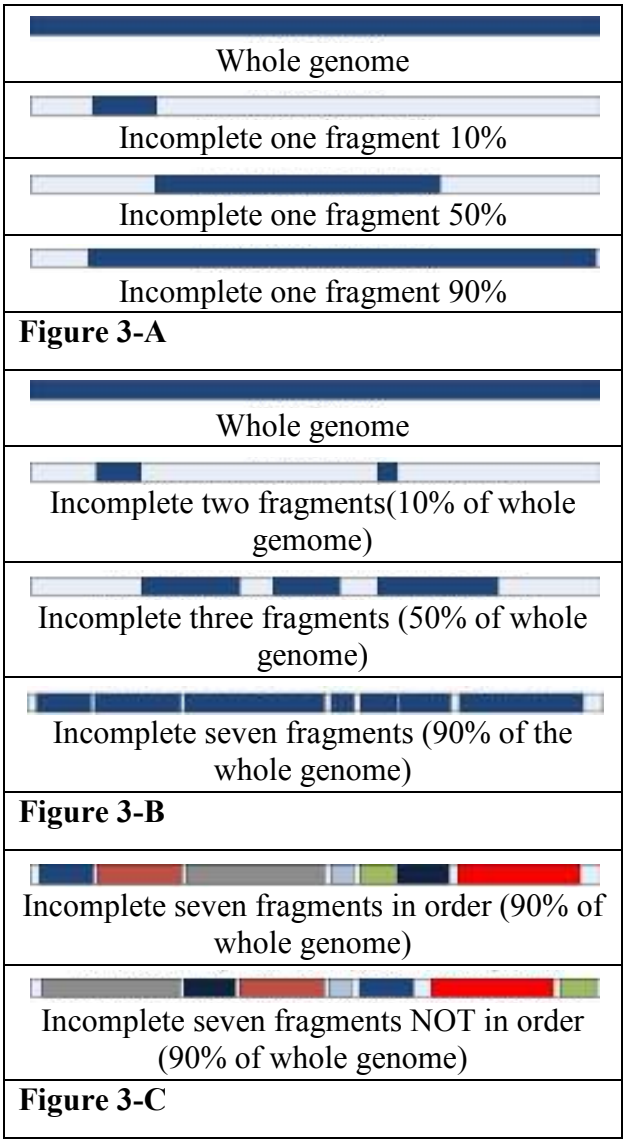

Figure 3, 3-A. Diagram describing the range of completeness of genomes for experiment 2, 3-B. Diagram describing the range of completeness of genomes for experiment 3, 3-C Diagram describing the range of complexity of genomes for experiment 4.

4.2.3 Comparing Phylogenetic Trees

Evaluating the constructed phylogenetic trees was accomplished by measuring the distances of the trees to a gold standard tree. The distance's measure to the gold standard tree is done by estimating the path-length-difference metric as described in Felsenstein [30]. A matrix is constructed for each tree, and the size of the matrix is m^2, where m is the number of tree leaves (the species) and each cell in the matrix has the number of branches that separates the species of the corresponding row and column. For each cell in the matrix and its correspondent in the gold standard tree matrix, the squared of the difference between them is computed. The distance is then calculated by finding the square root of the sum of these square differences, taking into account not to include duplicate values. The distance then was normalized by dividing it by the summation of the distances of the cells in the gold standard tree.

For example, consider the two trees in Figure 4), where the tree on the left represents the gold standard tree (species A, B, C, and D) and the second tree on the right represents the output tree of a tested algorithm (species A', B', C' and D'). The scoring matrices are calculated by summing the edges between two nodes in a tree. The finished scoring matrices are shown in Figure 5

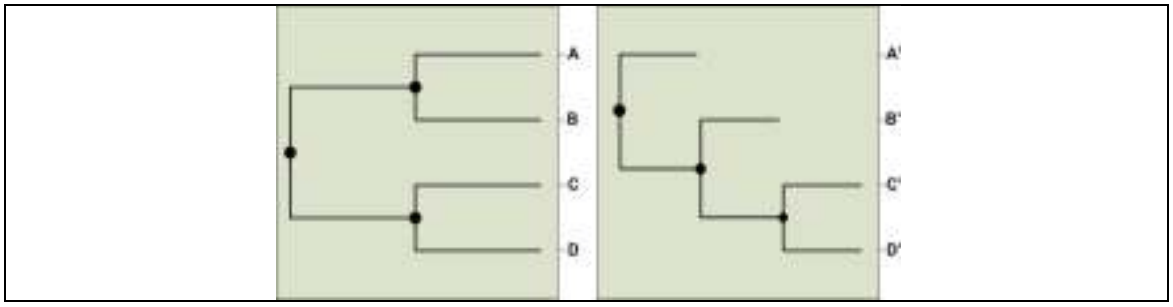

Figure 4 Two possible output trees, the one of the left is the gold standard tree and on the left is the algorithmic tree.

	A	B	C	D
A	0	2	4	4
B	2	0	4	4
C	4	4	0	2
D	4	4	2	0

	A'	B'	C'	D'
A'	0	3	4	4
B'	3	0	3	3
C'	2	3	0	2
D'	4	3	4	0

Figure 5 The scoring matrix for the two trees in Figure 4. The shaded cells represent the distance from one node to another.

The distance between the two trees is calculated by finding the root mean square between the trees:

Distance = $\sqrt{((AB - A'B')^2 + (AC - A'C')^2 + (AD - A'D')^2 + (BC - B'C')^2 + (BD - B'D')^2 + (CD - C'D')^2)}$

$= \sqrt{((2 - 3)^2 + (4 - 4)^2 + (4 - 4)^2 + (4 - 3)^2 + (4 - 3)^2 + (2 - 2)^2)} = (1^2 + 0^2 + 0^2 + 1^2 + 1^2 + 0^2) = (1 + 0 + 0 + 1 + 1 + 0) = \sqrt{3}$

The distance between the two trees is $\sqrt{3}$ or 1.732. To normalize the distance, it is divided by the sum of the distances between the species in the gold standard tree, which is:

(AB + AC + AD + BC + BD + CD) = 20

Normalized distance = $\sqrt{3} / 20 = 8.66\%$.

4.3 Results and Analysis

Results are the heart of this work; they would give an evaluation for the used compression methods on different datasets. In addition to regular datasets that are error-free, applications to datasets with errors that range from basic to severe errors were

conducted. The reason for using such datasets is to evaluate the performance of the methods on datasets that would suffer natural errors.

The evaluation of compression-based techniques is not trivial, as the techniques would provide dissimilarities values, and those values could not be evaluated, as there is no mathematical formula that would provide a value for the quality of results. Instead clustering algorithms for phylogeny were applied, and hence evaluation of the correctness of the resulting trees was conducted. And that turned the problem of evaluating the scoring matrices, to evaluating the correctness of the trees. Fortunately NCBI provided gold standard trees for some datasets, and a gold standard tree would be the reference for the resulting tree of an algorithm.

Although most research evaluates the correctness of a phylogenetic tree based on visual inspection, we do not recommend this approach, but instead we use a method from the literature called path-length-difference. This method provides a numerical value, but this value is still not normalized, so a contribution to the method was suggested to provide normalized values. The same method of evaluation was applied to phylogenetic trees resulting from multiple sequence alignment (MSA), to evaluate if compression-based methods would result in closer trees to gold standard trees over MSA or not.

4.3.1 Evaluating the hypothesis on datasets with no errors

The first experiment determined the feasibility of using compression algorithms for phylogenetic purposes. It tests the methods against regular datasets that are error-free, and its purpose is to evaluate if these methods are capable of measuring the distances of normal datasets. We compare the results obtained from various versions of compression-based techniques to those results obtained from multiple sequence alignment. In this

experiment, two datasets were used, a set of protein sequences and a set of complete mitochondrial genomes. The gold standard trees for both datasets are available to provide the base line comparison.

Table 1 and Table 2 show the results for experiment 1. The shaded cells reveal the compression techniques that surpassed multiple sequence alignment. In the protein dataset, the consistently desirable results were derived from UPGMA clustering using the scoring matrices of both Kolmogorov and Lempel-Ziv complexities. In the mitochondrial dataset, only Lempel-Ziv outperformed multiple sequence alignment.

Table 1 Comparisons of the compression algorithms and multiple sequence alignment for the protein dataset CK-36-PDB. Shaded cells represent cases when compression based

Test		Protein dataset CK-36-PDB	
Algorithm	Variant	Neighbor-Joining	UPGMA
Kolmogorov using Huffman coding	CD	2.395244	3.169468
	NCD	2.328382	2.264505
	UCD	2.328382	2.264505
Kolmogorov using LZW compression	CD	2.176959	2.165911
	NCD	2.210704	2.215544
	UCD	2.305268	2.238781
Lempel - Ziv complexity	Distance 1	2.337454	2.26598
	Distance 2	2.248862	2.192803
	Distance 3	2.244591	2.284809
	Distance 4	2.222918	2.371806
Multiple Sequence Alignment		2.182934	2.371806

Table 2 Comparisons of the compression algorithms and multiple sequence alignment for the Mitochondrial genome dataset in experiment 1. Shaded cells represent outcomes better than multiple sequence alignment algorithms performs better than multiple sequence alignment

Test		Mitochondrial Genome dataset AA-15-DNA	
Algorithm	Variant	Neighbor-Joining	UPGMA
Kolmogorov using Huffman coding	CD	7.871585	7.871585
	NCD	7.871582	7.871582
	UCD	7.871582	7.871582
Kolmogorov using LZW compression	CD	3.034474	3.034474
	NCD	2.797647	2.797647
	UCD	2.878755	2.878755
Lempel - Ziv complexity	Distance 1	1.357058	1.357058
	Distance 2	1.357058	1.357058
	Distance 3	1.357058	1.357058
	Distance 4	1.357058	1.357058
Multiple Sequence Alignment		1.5547053	1.878762

The reported results clearly indicate that compression-based technique provides a valid measure for the dissimilarity of biological sequences. The produced measures are in the same vicinity as the ones produced by multiple sequence alignment, or outperformed alignment-based results in several cases. It is also clear that with a careful selection of the clustering algorithm, the compression methods and associated distance measure can improve the overall results.

Results of Table 2 might be confusing for the reader, as all the LZC distances had the same results. The reader might think that all results are the same, but in fact the resulting trees had different branch lengths, although the algorithm was able to have consistent topological relationships among species. By looking at the following newick strings for the trees of LZC distance 2/NJ and LZC distance 4/UPGMA, we find that they have different branch lengths values, but share the same topology.

((Hylobates_lar:0.35896,(Pongo_pygmaeus_abelii:0.34718,(Gorilla_gorilla:0.29236,(Homo_sapiens:0.25472,(Pan_troglodytes:0.13975,Pan_paniscus:0.14321):0.11433):0.032986):0.050659):0.0128):0.048022,((Mus_musculus:0.36248,Rattus_norvegicus:0.35935):0.045141,((Balaenoptera_musculus:0.22948,Balaenoptera_physalus:0.23363):0.15935,((Ceratotherium_simum:0.35369,Equus_caballus:0.3531):0.023024,(Felis_catus:0.36895,(Phoca_vitulina:0.12961,Halichoerus_grypus:0.12935):0.23825):0.013218):0.0058083):0.004996):0.008113);

((Hylobates_lar:0.82879,(Pongo_pygmaeus_abelii:0.81548,(Gorilla_gorilla:0.73281,(Homo_sapiens:0.6775,(Pan_troglodytes:0.44288,Pan_paniscus:0.44288):0.23462):0.055311):0.082675):0.013308):0.064518,((Mus_musculus:0.83845,Rattus_norvegicus:0.83845):0.049034,((Balaenoptera_musculus:0.62995,Balaenoptera_physalus:0.62995):0.24279

,((Ceratotherium_simum:0.82737,Equus_caballus:0.82737):0.035366,(Felis_catus:0.848 21,(Phoca_vitulina:0.40841,Halichoerus_grypus:0.40841):0.4398):0.014523):0.010005) :0.014747):0.005823);

Figure 6 shows that these trees have the same topology even if they do not share the same branches' lengths.

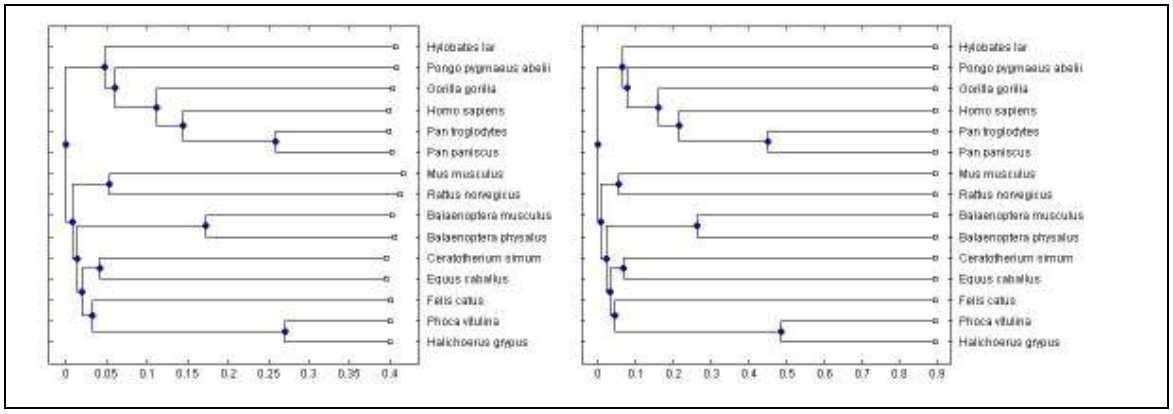

Figure 6 the trees of LZC distance 2/NJ (left tree) and LZC distance 4/UPGMA (right tree), although the two trees have the same topology, they are not exactly the same according to the branches' lengths.

Although it is difficult to compare the two trees by visual inspection, the reader would be able to identify some differences in the two trees, for example the distances of the pair (Pan_troglodytes, Pan_paniscus) to their ancestor in each tree are not the same, same with the pair (Phoca_vitulina, Halichoerus_grypus).

4.3.2 Evaluating the hypothesis on datasets with incomplete fragments

With the success of the first experiment, the second experiment was conducted to discover the capabilities of compression algorithms in clustering incomplete genomes. The purpose of this experiment is to conduct a study on the compression-based methods, and evaluate its performance on datasets with incomplete sequences, and to decide if

these methods are worth using or MSA would be a better solution for such datasets. For this, the mitochondrial genomes were again used, and percentages of the genomes were incrementally removed, then an application of an algorithm to randomly choose the starting position of the remaining genome was provided (Figure 3-A). We eliminated Huffman results from the charts as they did not provide comparative results in the first experiment. In examining Neighbor-joining method (Figure 7 - left) and UPGMA (Figure 7 - right), Lempel-Ziv complexity surpassed multiple sequence alignment in all the trials (with both NJ and UPGMA clustering) except once in both figures. Kolmogorov with LZW had viable results but not competitive to Lempel-Ziv.

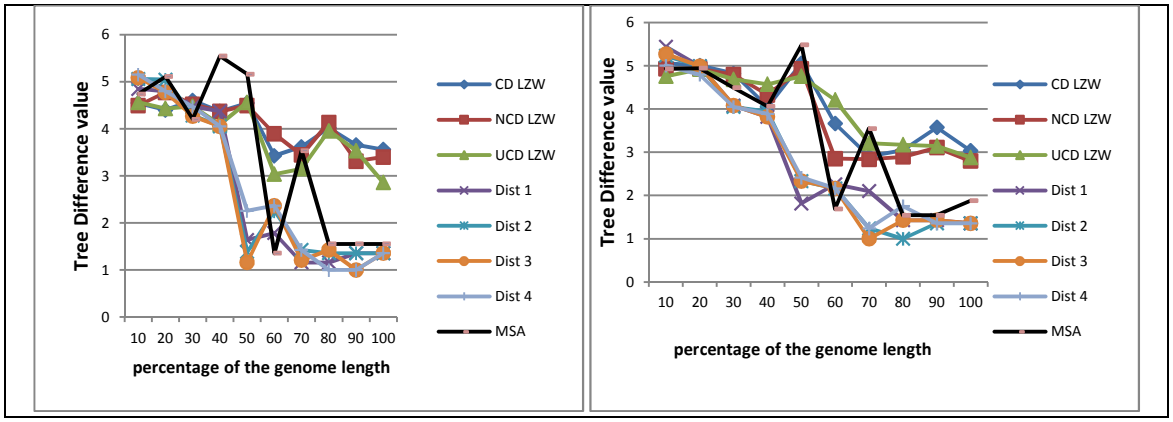

Figure 7 Experiment 2 results using Neighbor-Joining clustering (left figure) results using UPGMA clustering (right figure).

In general, Lempel-Ziv complexity had the best chances in revealing the similarities between the genomes, even while these genomes were not complete, but still Lempel-Ziv was able to address the dissimilarities between the sequences.

4.3.3 Evaluating the hypothesis on datasets with incomplete fragments that are not continuous

This experiment expanded on experiment 2 by breaking the genome into several pieces

and then reducing the total size to the same 10 – 90%, but allowed each fragment to be of different and random size (Figure 3-B). Multiple fragments were then combined together and tested. The results of experiment 3 mirrored that of experiment 2 in both the neighbor joining method (Figure 8 - left) and UPGMA (Figure 8 - right) in that Lempel-Ziv complexity outperformed multiple sequence alignment in almost every percentile. Also Kolmogorov using LZW compression, and Kolmogorov using Huffman coding failed to perform better than multiple sequence alignment (Huffman results were eliminated from the charts as well).

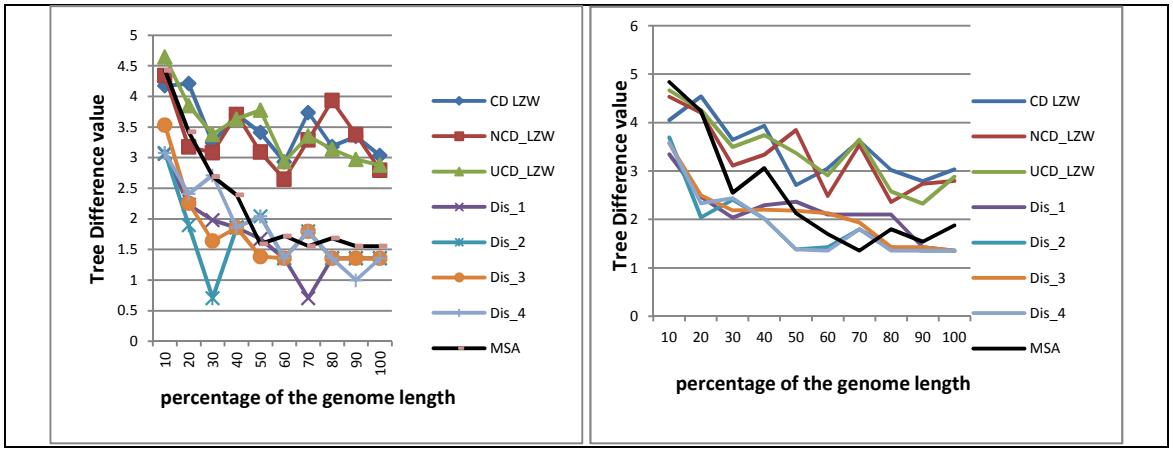

Figure 8 Experiment 3 results using Neighbor-Joining clustering (left figure) and results using UPGMA clustering (right figure).

Again, Lempel-Ziv showed powerful results of detecting similarities among sequences, while these sequences were not complete and not even from the same region within genomes.

4.3.4 Evaluating the hypothesis on datasets with incomplete fragments that are not continuous and not in order

This experiment was designed to "push the envelope" of multiple sequence alignment and the compression algorithms for the erratic datasets. The genomes for this experiment

were cut into multiple fragments, randomly decreased to a total 10 – 100% of the original size, and then rearranged (Figure 3-C). Again we eliminated Huffman results from the charts as they did not have comparative results. The compression algorithms returned results similar to the previous experiments, and multiple sequence alignment performed much worse (Figure 9). For the incomplete genomes less than 50% in length, Kolmogorov using LZW and Lempel-Ziv both surpassed multiple sequence alignment, but Kolmogorov was overtaken by multiple sequence alignment at 60% and above. Huffman still failed to perform as well as the other tests.

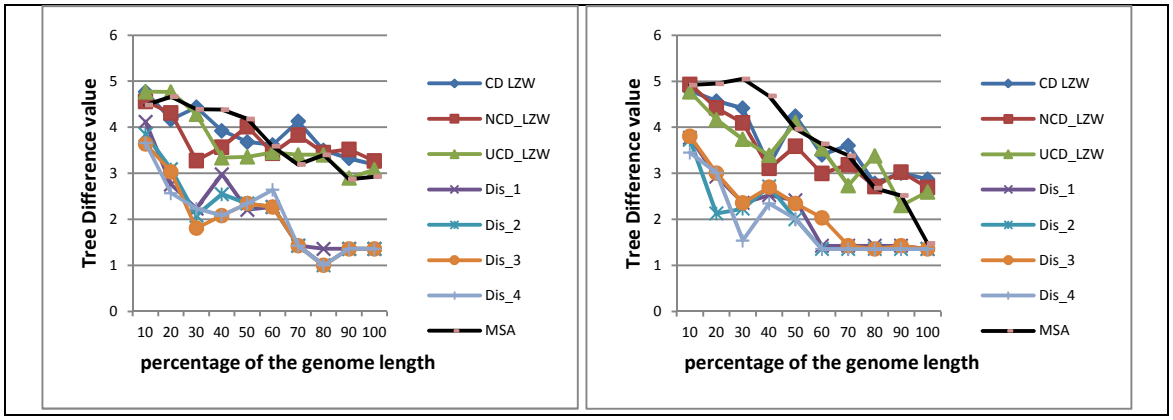

Figure 9 Experiment 4 using Neighbor-Joining clustering (left figure) and using UPGMA clustering (right figure).

In this experiment, Multiple Sequence Alignment had a big failure in detecting the relatedness between the genomes, which shows that alignment-based method would fail with sequences that has translocated segments. The rest of results show that compression-based method of Lempel-Ziv would still detect the relationships among genomes, and would give accurate clustering results compared to Multiple Sequence Alignment. Even LZW was competitive to Multiple Sequence Alignment in finding the right dissimilarities between the genomes.

4.3.5 Evaluating the hypothesis on datasets with mutated nucleotides

This experiment was designed to evaluate the performance of the compression-based methods, on mutated dataset. As the sequences evolve and mutate, it is difficult for methods like MSA to identify the relatedness among species. We have a hypothesis that compression-based methods, which has a nature of looking linearly into sequences; would identify the relatedness between sequences even if they have such kind of errors. Mutations (point mutations) were applied with percentages of 1%, 3%, 5% and 7% to the mitochondrial genomic datasets, and selection of point mutation was applied randomly. Comparison of the results to the multiple sequence alignment was conducted in the same fashion as the previous experiments, by measuring the resulting trees' distances to the gold standard tree.

Table 3 shows the results for this experiment, the shaded cells represent the resulting trees that had closer distance to the gold standard tree, these shaded cells are the values for Lempel-Ziv complexity.

Table 3 Comparison of the performance of compression against Multiple sequences alignment, on a mutated datasets with mutation percentages of 1%, 3%, 5% and 7%, the shaded cells shows the best results, where these results complexity of Lempel and Ziv, were for scoring matrices obtained by compression

		1 percent		3 percent	
		NJ		UPGMA	
Kolmogorov using Huffman coding	CD	7.1835178	7.8715849	7.1835178	7.8715849
	NCD	7.0541659	7.8715818	7.0541659	7.8715818
	UCD	7.0541659	7.8715818	7.0541659	7.8715818
Kolmogorov using LZW compression	CD	3.200615	3.2658798	3.4431587	3.152301
	NCD	3.2717484	2.996303	3.2776065	2.7907814
	UCD	3.4095661	3.0407893	3.1278642	2.9898938
Lempel and Ziv complexity	Dist	1.357058	1.357058	1.357058	1.7737186
	Dist2	1.357058	1.357058	1.357058	1.357058
	Dist3	1.357058	1.357058	1.357058	1.7737186
	Dist 4	1.357058	1.357058	1.357058	1.542317
Multiple Sequence Alignment		1.5547053	1.8787618	1.5547053	1.7737186
		5 percent		7 percent	
		NJ		UPGMA	
Kolmogorov using Huffman coding	CD	7.1835178	7.8715849	7.1835178	7.8715849
	NCD	7.0541659	7.8715818	7.0541659	7.8715818
	UCD	7.0541659	7.8715818	7.0541659	7.8715818
Kolmogorov using LZW compression	CD	3.6434739	3.0970471	3.6957492	3.0090805
	NCD	3.3240997	3.4874455	3.3869859	2.9641184
	UCD	3.2776065	2.7070388	3.5366032	3.0026985
Lempel and Ziv complexity	Dist 1	1.8582282	2.1005195	2.0543486	2.2758555
	Dist 2	1.357058	1.5298284	1.357058	1.357058
	Dist 3	1.357058	2.699943	1.1588089	2.2758555
	Dist 4	2.0166509	1.4259876	1.357058	1.357058
Multiple Sequence Alignment		1.5547053	1.357058	1.6849758	1.8787618

With limited mutations (not exceeding 9%) (which may lead to few changes in function but not in an evolution of the species itself), Lempel-Ziv complexity was able to detect the similarities among the species, and showed better clustering than Multiple Sequence Alignment, while Kolmogorov failed to detect similarities with this kind of sequence errors.

The results of Lempel-Ziv that are similar, are for trees that have the same topology but different branches' lengths.

4.4 Conclusions

Compression-based techniques to compare biological sequences are a viable alternative to multiple sequence alignment. In cases where the datasets contain errors such as incomplete genomes and/or out of order fragments or mutations that happened over time, compression techniques would cluster the dataset more accurately than multiple sequence alignment. Additional benefits of using compression analysis over sequence alignment include much shorter run times and independence of sequence length. Of the three compression techniques examined in this paper, Lempel-Ziv complexity has shown the best propensity in classifying incomplete and malformed datasets. To summarize these results, Lempel-Ziv complexity comes first in performance as alignment-free techniques. It also outperforms multiple sequence alignment, especially with unprocessed DNA datasets, as protein sequences are considered processed biological sequences, and they are rich with information that can be addressed easily using alignment-based techniques. From charts and tables, we can see that compression techniques in general, and Lempel-Ziv specifically were able to catch the relatedness among species, that comes from the

algorithmic nature of compression techniques, as they look linearly into the sequences, and ignores the arrangements of the fragments. Additional application for the compression-based techniques would be trans-located genes, and those would be addressed by compression techniques, while alignment-based would fail. In addition to these applications, compression techniques addressed point mutations in DNA sequences that undergone up to 7%.

CHAPTER 5

MOTIF-BASED TECHNIQUES

The previous chapter showed the usage of compression-based techniques as alignment-free methods. The strength of compression-based comes from their heuristic speed and their strong parsing techniques that would catch similarities between sequences. Though this method does not address the hidden signals within the sequences, some of these signals might provide major information for identifying the relationships between sequences.

This chapter is considering using different signals within the sequences to address the relationships between sequences; the approach is modified from the k-mers approach, and considers the order, occurrence and frequency of the signals.

The work of this chapter is extensive and covers addressing usage of all possible signals, random signals, signals that have biological relevance like restriction enzymes or signals taken from CDs regions.

5.1 Nature of DNA sequence

DNA sequences are not random in their structures, and it is believed that each fragment/subsequence of the DNA sequence carries a message or a signal. The hypothesis used in this research; is that these signals would be similar if the genomes are closely related. For example sequences that carry the same restriction enzymes cut positions [31] might be related and would have similar functions; the same would be with

sequences that carry transcription factor binding sites. Other signals would be motifs of specific nature, unique shortest substrings [13] within the sequences, or just motifs with biological relevance that are unknown to the literature. Another feature that DNA sequences hold, that they carry tandem repeats in their structures, and again these tandem repeats might be signals with significance. All these features are needed to be addressed when comparing the biological sequences.

One way to analyze the comparison problem is based on the fact that similar genomes, share similar structures and functions, and although subsequences with similar functions do not necessarily have similar exact structures; they carry similar signals within these structures. And by identifying these signals, we would be able to classify these genomes, and address better measurement for their relatedness.

Notice that these signals might be hidden, and/or overlap with other signals, also they might be of different lengths.

For the previous reasons we designed an approach that would consider all or a group of prospective signals of specific length k, which would consider the unknown hidden signals. Also would consider the overlapping signals, and could be applied using signals of different lengths.

The question of identifying such hidden and unknown signals is not easy. The focus of this work is to try to identify these signals and their functions, or to take advantage of their existence within the sequences and use them for clustering purposes. The hypothesis in this work is to take advantage of these hidden signals within the sequences; to identify the relatedness between a group of species. This would be done by considering all the

possible chances for the existence of these signals within the sequences, and use them to identify the biological distance between the sequences.

The focus and challenge of this work is to investigate if addressing such signals would improve the clustering process, and reveal a better measurement for the relatedness among species. We consider different signals of different lengths for comparing the sequences, we also consider random groups of these signals to measure the quality of the results in each case, and we measure if some randomly selected signals would have better results than others or not. In addition, this work considered the use of signals that have biological relevance like restriction enzymes, and also signals that occur within specific regions that have biological functionality in the DNA sequence, like those in CDs regions. Finally and as a conclusion of the strength of this approach, applications to datasets with errors were conducted.

5.2 Experimental Design

The design of the experiment should meet the needed requirements to test the hypothesis. Recalling that comparing DNA sequences results in numerical values that represent biological distances between species. These values are subjective with each dataset, and would be meaningless if they are not used to address the relationship for the entire group of species. Verification of the correctness of these distances is not an easy task; looking at these numerical values will not reveal the correctness of the results, and there should be a way to measure the correctness. Clustering the species based on the resulting distances would provide a way to evaluate the correctness of these results; this would be done by evaluating the trees instead of the distances. The clustering would be done using bi-

clustering algorithms for phylogeny. The resulting trees of the phylogeny would be a good way to evaluate the quality of the results, but even with these trees; evaluating their correctness is needed as well; so we can make sure that they represent the correct relations among species. This evaluation happens mostly by visual inspection, which in many cases might be misleading, and have a drawback of not providing a numerical value for the evaluation.

It is also important to have the results compared to some reference, as these numerical results will not provide evidence for any quality improvement unless if they are compared to a known method. As this work is an alternative method for sequence alignment and its drawbacks, then the results would be compared to those obtained by sequence alignment.

The previous discussion addressed the main points needed for the experimental design, and hence a summary for the needed steps to accomplish any experiment in this work is concluded as follows:

1. Generate the list of the k-mers, for example for k = 3, it would be all the possible 3-mers, which would result in 64 words (4^3), or the list could be a random selection of about 20% of all the possible words. Which would be 13 random 4-mers; also it could be a list of biological signals.
2. Convert the DNA sequences according to the input list of k-mers (refer to Figure 10)
3. Generate the scoring matrix based on pair-wise comparison and not multiple sequence comparison, using longest common subsequence (LCS) and Lempel-Ziv Complexity of distance measure 2 (LZC).

4. Build the phylogenetic trees using UPGMA and Neighbor-Joining (NJ) phylogenetic algorithms.
5. Repeat the last step using scoring matrix generated by multiple sequence alignment (MSA).
6. Measure the distance between the generated trees and the gold standard tree (defined in the terminology in chapter 2), the method used to measure this distance is the path-length-difference (discussed in chapter 4).

The first two points are the main core for this work, and the following subsection would explain how to apply them.

5.2.1 Conversion of DNA sequence to sequence of signals (motifs)

To consider all the possible signals of specific length k (all possible k-mers), a production of all the possible combination of the length k is generated, this would result in a words' list of size 4^k, where 4 is the number of the used nucleotides in a DNA sequence(A, C, T and G).

The generated list is used as the main seeds for the signals needed to be identified within the sequences. If a signal exists in the DNA sequence, we substitute it with a unique code for this signal, and this would have conservation of the order for the signals within the sequences. Also this design would save computational time, when the list is small and the sequences are longer.

Figure 10 shows how to identify the existence of these signals in the sequences, and how to convert the DNA sequence to a sequence of signals/words with the proper order of these signals. The used motifs/signals list in this example is on the right side of the

figure; in this list each motif/signal would have a name (code), and the left side of the figure has the original sequence, parsed as words of length k (k = 4 in this case).

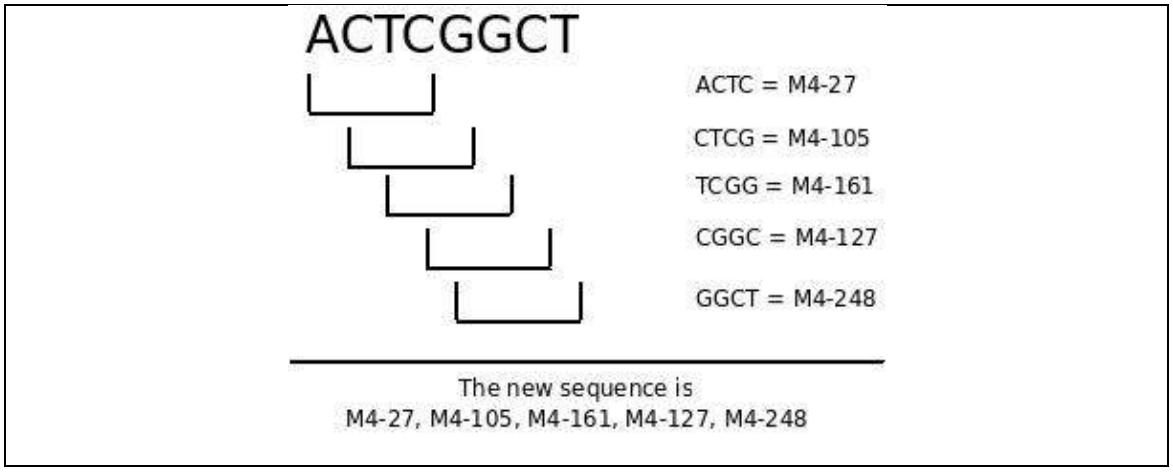

Figure 10: the list on the right side is the preferred signals to be used for the approach and their proper code, the sequence on the left is parsed to subsequences each of the same length as the signals' length on the right. If there is occurrence for any signal from the list within subsequences, this subsequence would be replaced with the matching code; if the subsequence is not listed, then it will be deleted.

The used motifs/signals list in this example is on the right of the figure, and each has a name (code), and on the left side is the original DNA sequence. We identify the signals from this list that exist in the sequence, and if they occur; their codes would be assigned with the proper order to the new sequence of signals. Thus we convert the DNA sequence to a sequence of signals, also notice that this approach consider all the overlapped signals. We also need to mention that sometimes some of these signals do not exist in the sequence, or they occur more frequent, and in either way that would impact the results for the relatedness between the sequences. This would be a major difference between the converted sequences, and would address similarity or dissimilarity among species.

5.2.2 The experimental design steps, discussion on the rest of the steps

The conversion step is the heart of this work, as in this step addressing the signals in preference happens, but the work remains incomplete as long as there is no way of comparing the converted sequences. The nature of the converted sequences carries two main features, the first feature is a new alphabet of preferred signals, and this would motivate us to use similar comparison algorithms/approaches as in regular DNA sequences. Simple and efficient algorithm like longest common subsequence (LCS) [32] would address the distances between the converted sequences. The second feature of the converted sequences is the conservation of the signals' order within the sequences; this was missed by other research that was also based on k-mers [12]. The order of the signals would have a great impact on the results. Although there is some possibility of having few or more mobile subsequences, we would still be able to use an algorithm that would address the order feature, like Lempel-Ziv Complexity (LZC) [4]. LZC is based on compression complexity and has a great success in identifying the relatedness of different strings.

The comparison method would result in numerical values that represent the distances between species, which are needed to cluster the species and identify the correctness of the resulting distances. The clustering would be using hierarchical clustering algorithms like UPGMA and NJ. The results of these algorithms are in the form of trees, and although most researches use visual inspection to evaluate phylogenetic trees, we don't recommend it for the following reasons:

1. Visual inspection uses personal judgment, and personal judgment is not usually accurate, and would mislead the evaluation process, especially if it is not compared against some reference
2. Visual inspection cannot identify the correctness of trees with big number of species. In fact with a big number of species like 1000 species, it would be impossible to find out the relationships between each species and the rest of species.
3. Visual inspection does not provide numerical value for the comparison, and hence no clear decision could be achieved based on it. But using a computational method to measure the distance of the resulting tree to a reference tree, would give a decision for the entire experiment.

For these reasons, a computational approach to measure the distance between the resulting trees to a gold standard tree was used. This approach is called path-length-difference [30], and it was modified to give normalized values.

And finally it is important to compare the trees from our approach to the resulting values from MSA, and evaluate if our approach would have better results on not.

5.2.3 Different algorithms of the experiments

This subsection has a discussion for some of the methods used to verify the hypothesis of this work, specifically methods that are new to the reader or those that have modification to fit into the experiments.

5.2.3.1 Normalizing Longest Common Subsequences:

LCS is based on dynamic programming, and has a well established reputation and implementations, but the problem that the generated scores are not normalized, and these

scores cannot be used to build a phylogenetic tree. To understand this problem, we refer to the following example:

Consider these sequences

S1: GTTAATGCCACCAAAAAAAAA (length 21)

S2: GTTAATGCCACCGA (length 14)

S3: TCCCTAGCT (length 9)

The LCS for all the pair-wise sequences is as follows:

S1: GTTAATGCCACCAAAAAAAAA

S2: GTTAATGCCACCGA

LCS is GTTAATGCCACCA and the score is 13

S1: GTTAATGCCACCAAAAAAAAA

S3: TCCCTAGCT

LCS is TTAGC and the score is 5

S2: GTTAATGCCACCGA

S3: TCCCTAGCT

LCS is TTAGC and the score is 5

The resulting scores of using LCS for these sequences are shown in Table 4.

Table 4 A table represents the scores of using LCS on the example sequences

	S3	S2	S1
S3	9	5	5
S2	5	14	13
S1	5	13	21

Two problems are seen here from Table 4, the first one is that these scores are maximized and not minimized, which means that two closely related sequences have a bigger score, while the clustering algorithms are designed for smaller scores with two closely related sequences (representing shorter distances between sequences). The second problem is that these scores are not normalized, for example the relationship between S1 and S3 is 5, and this 5 is not relative to any value, and that means that several comparisons with this value would mean different relatedness between species, which is not consistent. But if it was 0.43, it would represent a relative distance of 43% for both sequences.

To solve this problem, we normalized the resulting score by dividing it by the length of the shortest sequences of the measured pair; and that would normalize it, then subtract the result from 1; and that would minimize the relationship between the sequences instead of maximizing it, and would also result in a normalized value, as in the following matrix (Table 5).

Table 5 A table shows the results generated after using the normalizing function which was suggested

	S3	S2	S1
S3	0	1-5/9	1-5/9
S2	1-5/9	0	1-13/14
S1	1-5/9	1-13/14	0

The normalization issue was very essential for the clustering step, as the clustering algorithms would take only normalized matrix with zero diagonal.

5.2.3.2 Lempel-Ziv complexity [4]:

Lempel-Ziv complexity of distance measure 2 was used. Please refer to the reference for more details or section 4.1.2.

5.2.3.3 Path-Length-Difference:

The comparison between trees was done by estimating the path-length-difference metric [30]. For more details review section 4.2.3

5.3 Experiments

The experiments are designed and carried to answer proposed and motivation questions for this work, these questions are:

1. Would some motifs/words/signals provide good results for sequence comparison? Would these signals have better comparison results over traditional sequence comparison methods like those that are alignment-based?
2. If the answer for point 2 is yes, is it possible to change the selection of the k-mers for the experiment? Would that enhance the results? In other words, are there certain words that would improve the clustering results?
3. If the answer for point 2 is yes, would we be able to use signals with biological relevance like restriction enzymes; to improve the results?
4. If the answer for point 3 is yes, would it be possible to find hidden signals within the sequence with biological relevance? And use them to have valid results?
5. Finally, if the first four questions have answered positively, is it possible to use the approach on datasets that have errors, and still get better results than with MSA?

To answer these questions, designs of an experiment for each question was proposed.

5.4 Results and Analysis

All the experiments have the same steps that were discussed previously in the Methodology section. The only differences between them are the list of used k-mers, the used dataset in some experiment, and the purpose for using this specific list. In addition to these differences, one experiment had a change in the conversion algorithm, which would be discussed in that experiment (restriction enzymes).

Datasets Collection:

1. The first dataset used was for mycobacterium dataset, and we used it for the first 3 experiments.
2. The second dataset was for a mitochondrial genomic dataset, and was used for experiment 3, 4 and 5.

5.4.1 The First Experiment::Viability of the method

The purpose of this experiment is to evaluate if using such hidden signals within the sequences, would provide good results, and also if those results would be better than results of traditional alignment-based methods. This experiment deals with all the possible k-mers; as some of them might be hidden signals within the sequences and have strength. The used list of k-mers is all possible k-mers.

Figure 11 shows the results of using all possible k-mers, and it shows outstanding results, as all distances of any value for k was less than 1.25%, while with MSA the results were above 1.8%, recalling that smaller values express better distance measures.

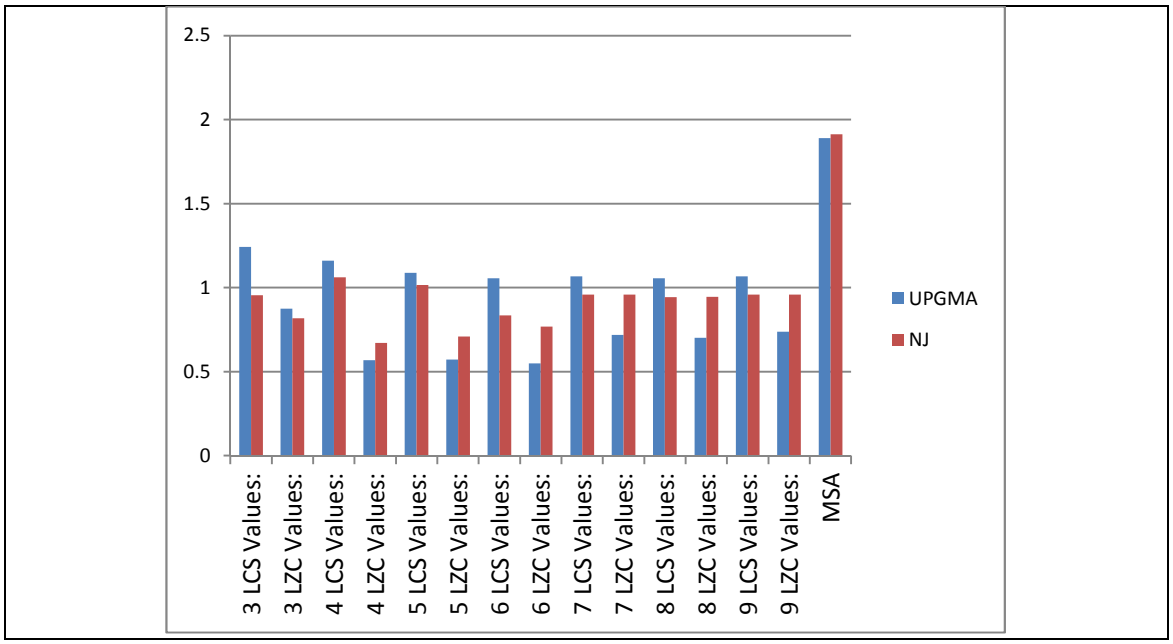

Figure 11 this figure shows the results of using our algorithm with different parameters, here k ranges from 3 to 9, the used methods of comparison are LCS and LZC, and the clustering methods are UPGMA and NJ. The chart shows that in all cases our approach outperformed MSA (multiple sequence comparison), with significant results.

The figure shows significance in the results using our approach compared to those of MSA. That proves our hypothesis that emphasizing such signals would improve the results, and would answer the first question.

5.4.2 The Second Experiment::Selection of random k-mers

With the success of the first experiment, we proceeded with the second one, and used lists of random signals, these signals were selected randomly from all possible k-mers.

The lists were generated randomly by selecting k-mers from all possible k-mers lists, with percentages of 10%, 20%, ..., 90%, and these selections were applied to k values of 3 to 9, and comparison methods LCS and LZC, and clustering NJ and UPGMA.

Expectations for the randomly generated lists are: either the list carries signals with strength, carries weak signals or carries both. The purpose of running this experiment is to test the hypothesis of having better results, worse results or close results to those obtained from first experiment.

Results were presented as charts, each chart would be for one comparison method (LCS or LZC), and one clustering algorithm (NJ or UPGMA), and would include all the different values of k, and at different levels of percentages.

The horizontal axis represents the different percentages used to select the randomly generated lists in the experiment, and the vertical represents the degree of closeness to the gold standard tree.

All charts showed better results compared to MSA, and some of them were even better than results of the first experiment. But some results had lower quality, and some results were very close to the results of the first experiment.

Figure 12 shows the results of applying the approach using LCS and NJ Figure 13 shows the results for the same experiment using LZC and NJ, Figure 14 shows the results for LCS and UPGMA, and Figure 15 shows results of LZC and UPGMA.

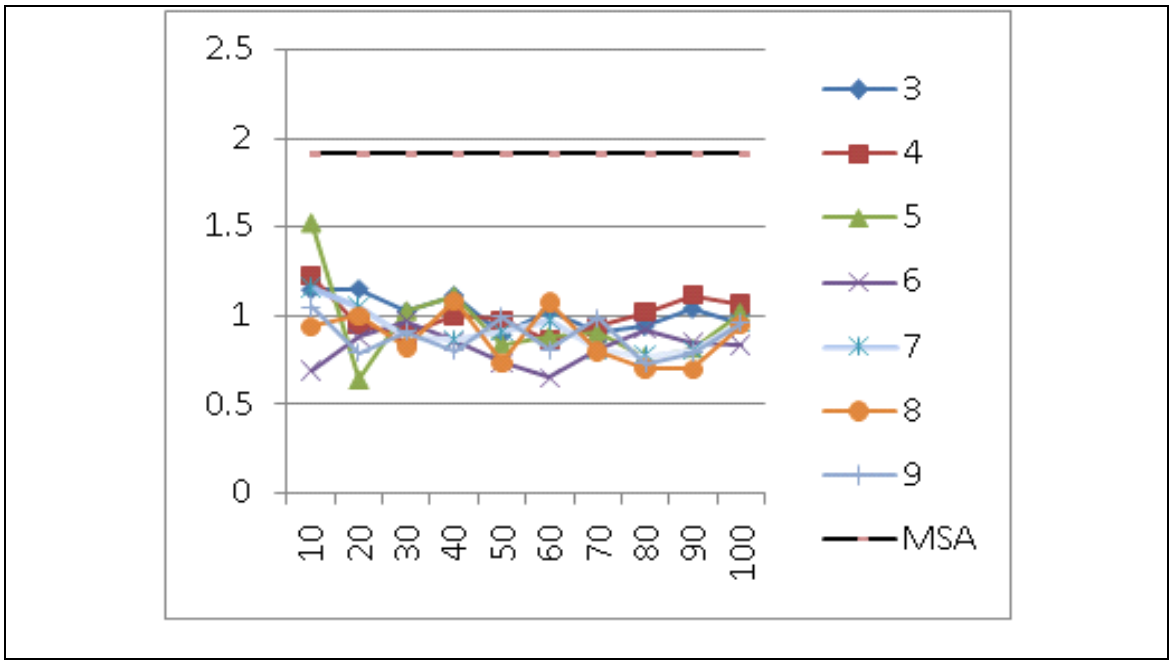

Figure 12 Chart represents the results of the approach using LCS as a comparison method and NJ as a clustering method; compared to the results of Multiple Sequence Alignment, the vertical axis shows the distance value of the algorithmic tree to the gold standard tree, and horizontal axis represents the percentages of the randomly selected k-mers compared to the whole pool. Each line in the chart represents one value for k.

The charts show that with just a random selection of k-mers; the approach would still provide better performance than using alignment-based methods (the black horizontal line in charts represents results of MSA), notice that with any length for k, the method was still successful and provided a better way of comparing the sequences, compared to multiple sequence alignment. Also with a small list for k-mers (up to 10% of all the possible k-mers of specific k), the results would still outperform MSA.

Another point to mention from the charts, that some runs for the experiment showed better results than those obtained in the experiment for all possible k-mers, as in Figure 15 with 60% random selection of k-mers of length 6, the resulting tree has a distance to the gold standard tree of 0.489%, which outperformed any result of all possible motifs as

you can compare it by looking at Figure 11, also another value in Figure 14 was worse than the results of all possible motifs; as the random selection of 10% for k-mers of length 5 has a value of 1.877005%, which is worse than any value in Figure 11 and that shows that some signals would do even better when they are used alone compared to the usage of all the possible k-mers. That would also be computationally less expensive, while other signals would do worse. So in this experiment we were able to answer the second proposed question.

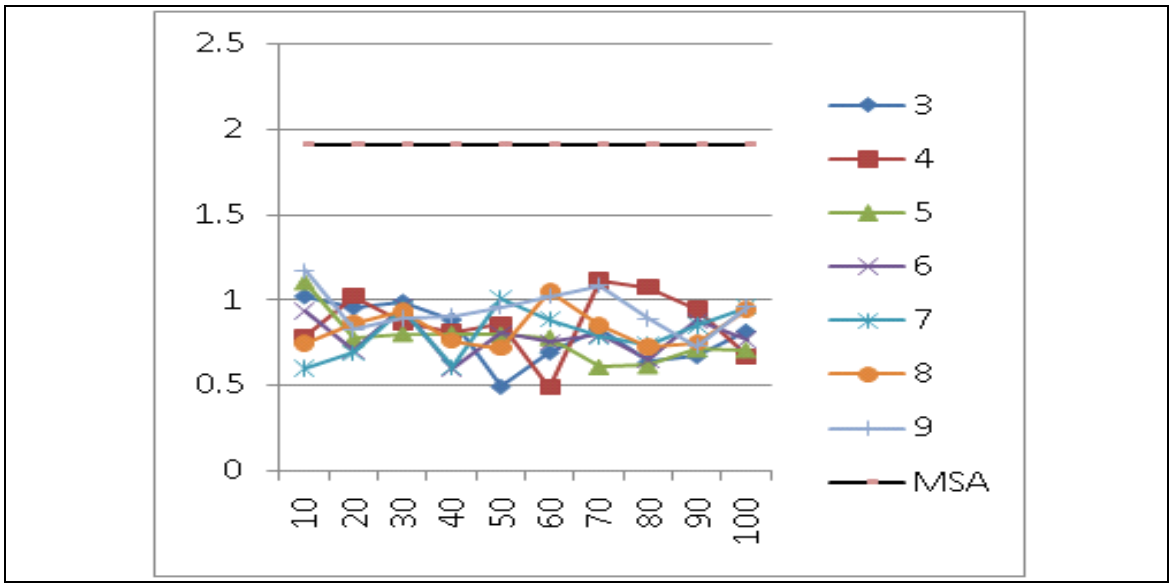

Figure 13 Chart represents the results of our algorithm using LZC as a comparison method and NJ as a clustering method; compared to the results of Multiple Sequence Alignment, the vertical axis shows the distance value of the algorithmic tree to the gold standard tree, and horizontal axis represents the percentages of the randomly selected k-mers compared to the whole pool. Each line in the chart represents one value for k.

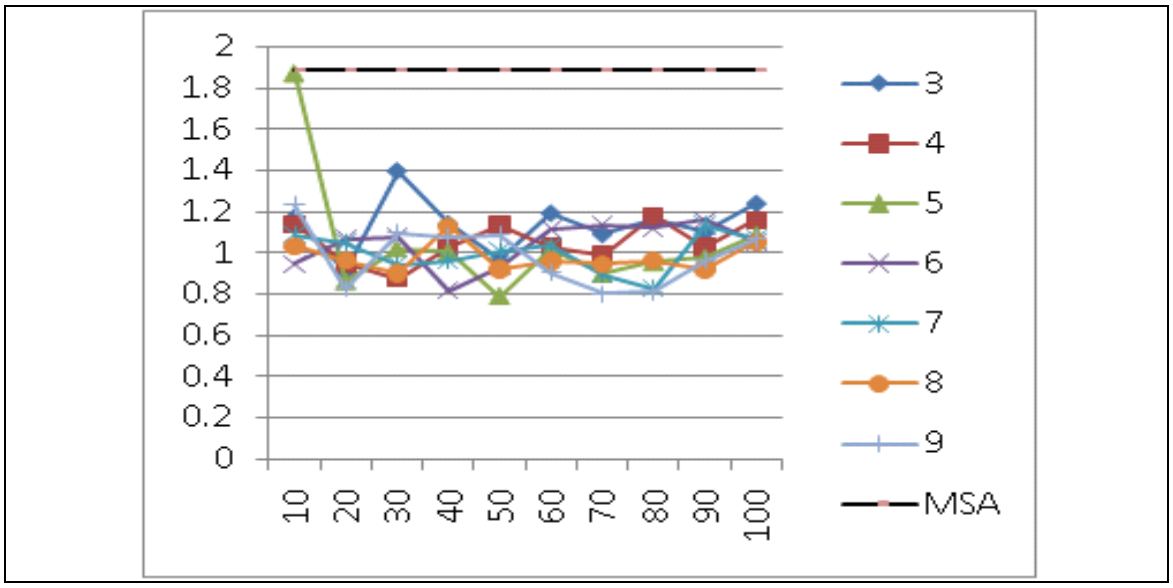

Figure 14 Chart represents the results of our algorithm using LCS as a comparison method and UPGMA as a clustering method; compared to the results of Multiple Sequence Alignment, the vertical axis shows the distance value of the algorithmic tree to the gold standard tree, and horizontal axis represents the percentages of the randomly selected k-mers compared to the whole pool. Each line in the chart represents one value for k.

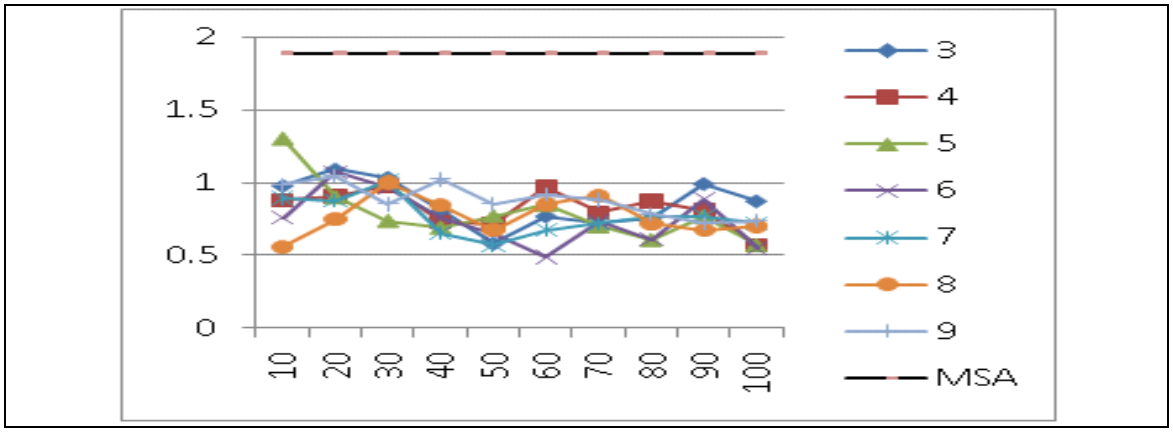

Figure 15 Chart represents the results of our algorithm using LZC as a comparison method and UPGGMA as a clustering method; compared to the results of Multiple Sequence Alignment, the vertical axis shows the distance value of the algorithmic tree to the gold standard tree, and horizontal axis represents the percentages of the randomly selected k-mers compared to the whole pool. Each line in the chart represents one value for k.

5.4.3 The Third Experiment: using restriction enzymes cut positions as the words list

The second experiment showed that results would be impacted with the selection of the words (k-mers) list. And that some signals would have higher impact over others, which motivated us to proceed with the third experiment that deals with words that have biological relevance, and to see how these words would impact the results. The used signals were obtained from a database of restriction enzymes cut positions.

Restriction enzymes are special nucleotide signals that cut the DNA double or single stranded sequence at specific recognition positions. We believe that DNA sequences that share similar restriction enzymes cut positions, would also have similarities in their functions and structures.

We used restriction enzymes cut positions that have lengths 4 to 8 nucleotides. As the number of restriction enzymes for each length was small, we had to use all of them as the words' list, hence we used a modified implementation for the conversion algorithm. This modified algorithm would integrate different lengths of the words, and the following subsection shows how we integrated our conversion approach to take advantage of all restriction enzymes cut positions.

5.4.3.1 More details on using the restriction enzymes:

As restriction enzymes are not many, we had to integrate all of them in the converted sequence. To do so, we looked at restriction enzymes of length 4, and identified their locations in the sequences, we looked then at restriction enzymes of length 5, 6, 7 and 8. This would consider priorities of words with smaller length first, then bigger length, and again these words would have names/codes in their list, so the generated sequences would have a new alphabet that represents words of different lengths and have biological

relevance. The rest of the experiment would be the same as in the previous two experiments. The following example shows the new modification for the conversion approach.

Assume this sequence

ACCGTGC

And the restriction enzymes list we have with their codes is

ACCG = RE1

CGTG = RE2

ACCGT = RE3

Applying the restriction enzymes of length 4 would generate

RE1 (at position 1), RE2 (at position 3).

Applying the restriction enzymes of length 5 would generate

RE3 (at position 1)

And the final sequence of restriction enzymes after integrating both lengths would be:

RE1, RE3, RE2.

Notice we consider the position first, and if we have more than one restriction enzyme (overlapped signals) that occur in that position, we then give the smaller length higher priority in the generated sequence.

Figure 16 shows the results of using a list of restriction enzymes cut positions on the mycobacterium dataset. The results showed better quality with the application of the restriction enzymes' list than those of using MSA.

Figure 17 shows the results of the same experiment, but on the mitochondrial genomes dataset, and again the results outperformed those obtained by MSA.

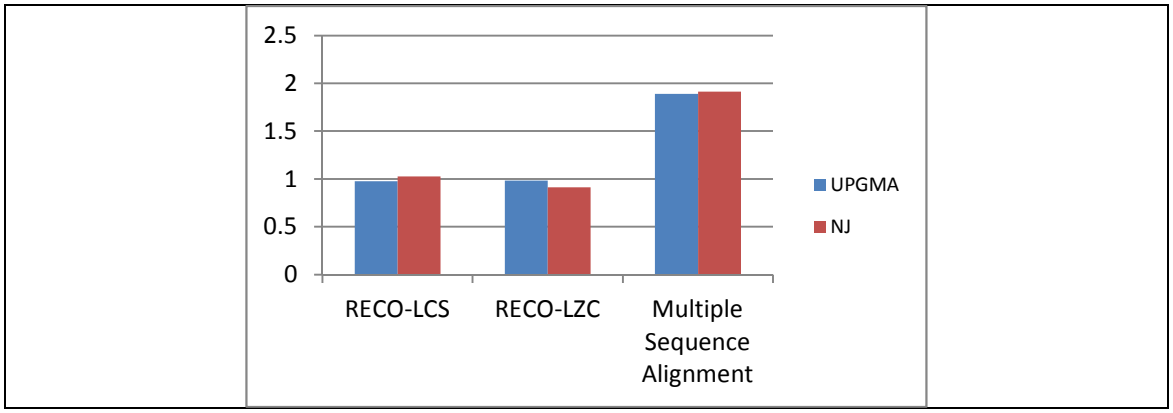

Figure 16 Results of using the approach with a list of restriction enzymes cut positions; Multiple Sequence Alignment results were included as a reference for comparison.

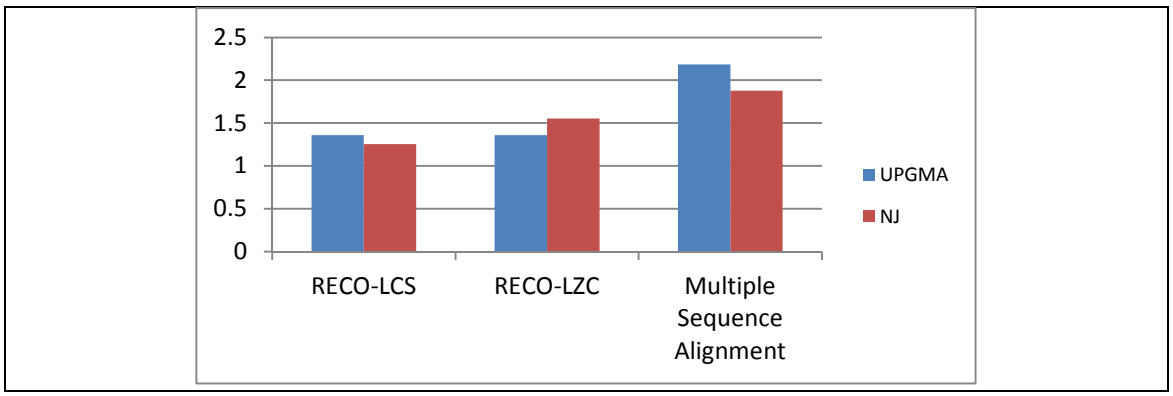

Figure 17 These are the results of using our algorithm with a list of restriction enzymes on the mitochondrial dataset, we also included Multiple Sequence Alignment results as a way of comparison.

The two figures show that results of using restriction enzymes were better than those of MSA. Though in some cases and using the random selection (refer to experiment 2 in section 5.4.2), the results might be even better using the random selection, as shown in Figure 12 and using k = 6 and random selection of 60% we got a 0.489% of tree distance difference to the gold standard tree. Which proves that there are some strong signals that

are not known to the literature, and those signals would improve the results of the comparison method.

5.4.4 The Fourth Experiment::Using k-mers that occur only in CDs regions of the genomes

As our hypothesis of using words with biological relevance had promising results, as with the restriction enzymes, we continued searching for more signals that would also give high quality.

In this experiment, we used signals/words from the CDs regions of the genomes. As these regions are rich of biological information; we thought they would improve the results. In fact CDs are the main DNA source for functional genes, and a lot of species that are closely related, would have similar functions and in turn genes with similar structures. Hence we generated a list of the k-mers that occur within the CDs regions.

Elimination of words lists of lengths 3, 4 and 5 was applied, as those lists were all possible k-mers of these lengths, and would have same exact results as in the first experiment.

The used dataset here were for entire genomes, these mitochondrial genomes are rich with CDs regions, and was a good fit for this experiment, as they also have a gold standard tree.

Figure 18 shows better results for using our approach with signals from the CDs. The results of Figure 18 show that these signals are rich of information that would improve the quality of the method; hence these signals would be a major source as input lists of the approach. And this would come with a positive answer for the fourth question.

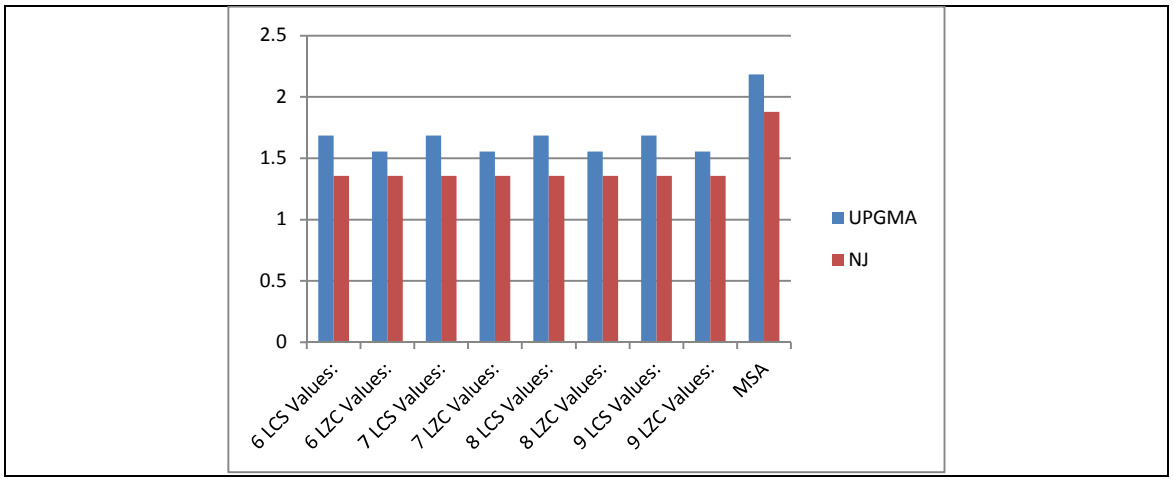

Figure 18 This figure shows the results of using our algorithm with lists that were generated from CDs regions, k ranges from 6 to 9, the used methods of comparison are LCS and LZC, and the clustering methods are UPGMA and NJ.

5.4.5 The fifth experiment::Application of the approach to datasets with different level of gaps errors

We finally applied the approach to special datasets; these datasets were generated and manufactured from the mitochondrial genome dataset, and they are incomplete genomes and/or with errors. The reasons for applying the approach to such datasets, is to measure if it would be able to identify the relatedness among species with errors or not.

These datasets are divided into three categories, the first category is for a dataset where each sequence is a fragment from the original genome, and each fragment's content is a percentage of the original genome's content. The content was chosen randomly from the genome's content. For this category we generated two datasets one with 50% content and the second for 70% content.

The second group has each sequence composed of several fragments from the original genomes, and these fragments are in order. Each sequence would be the merging of several fragments from the original sequence, and these fragments would have a content

represented as a percentage of the original genome, these fragments were chosen randomly from the genome's content and did not overlap. For this category we generated two datasets with percentages 30% and 90%.

The third category is similar to the second one, but the fragments were shuffled randomly, which means that a new sequence has fragments that are not in order, and would still have a content that is represented as a percentage amount of the original genome. These datasets were generated with percentages 40% and 80%.

For more details on these types of datasets, please refer to Figure 3.

We compare the results of using our approach on these datasets to those resulting from MSA on the same datasets. We are evaluating if our approach would identify the relatedness of the species in these datasets, even if they have errors, and if these results would be better than those of MSA.

Figure 19 shows the results of applying the approach to these datasets, each group of columns (colors blue, red and green) represent one dataset and the use of one clustering algorithm (NJ or UPGMA), and each column is the result of using either LCS, LZC or MSA. The results show that in most cases our approach outperform MSA, except in two cases as with the dataset of using several fragments, with 30% contents of the original sequences, and using all possible 4-mers, and LCS comparison method with UPGMA clustering algorithm. The quality of result in this experiment was lower than MSA, same with the dataset of (80% contents, several fragments not in order, 6-mers selected from CDs and using LCS and UPGMA); the result again was lower than MSA.

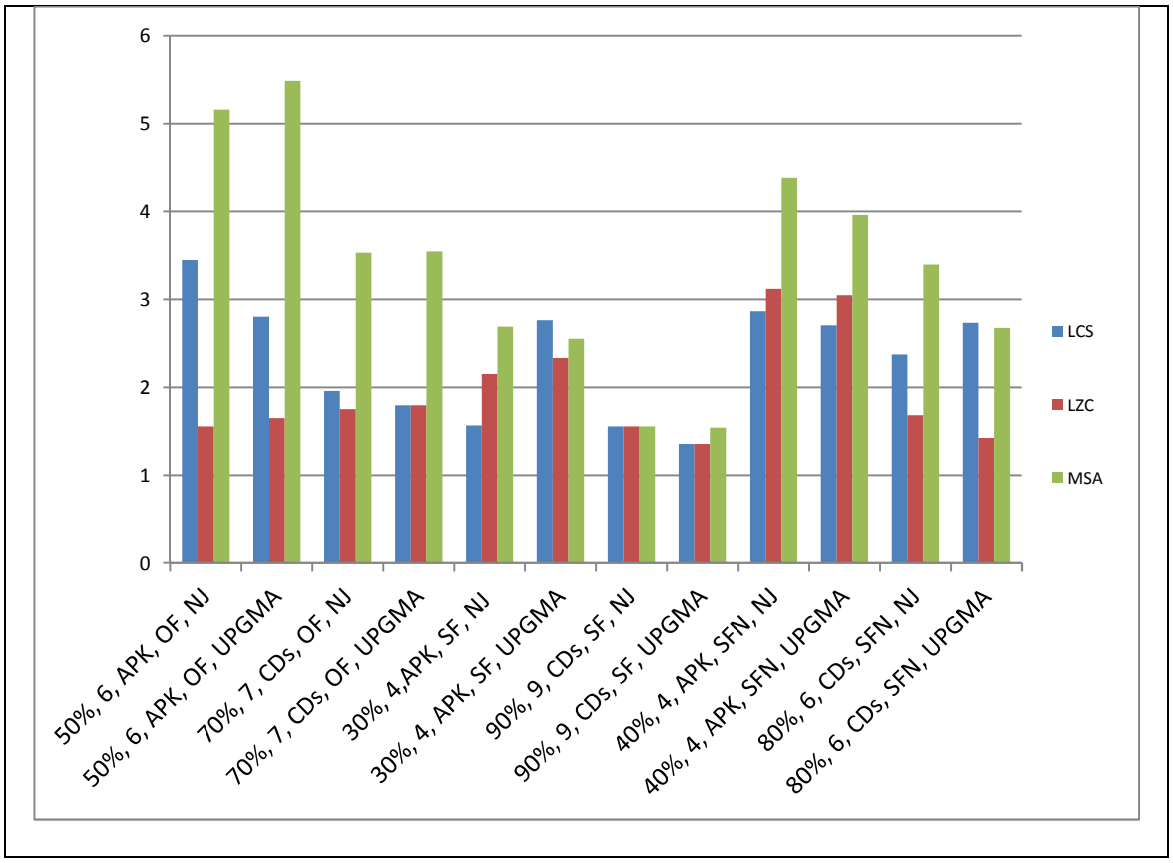

Figure 19 shows the results of applying our approach to datasets with high level of errors.
Abbreviations: APK (All Possible K-mers), CDs (Coding Regions), OF (One Fragment), SF (Several Fragments), SFN (Several Fragments Not in order). Horizontal axis shows the distances for the generated trees to gold standard. Notice MSA was applied to the entire genomes.

The usage of this approach with datasets that have errors would be more convenient than using MSA, as most of the results of our approach outperformed MSA results, 34 results were better than MSA out of 36 runs (94.44%).

5.5 Conclusion

The experimental results we had showed that some signals would improve the quality of comparison of biological sequences. It showed that there are hidden signals that are

possibly overlapped within the sequences, which could identify and improve the relatedness between species. It also showed that with a small number of the signals; we would still be able to get better results than those of MSA, and even in sometimes better results than in all possible signals of specific length. That would mean that we randomly chose strong signals that would identify the relatedness between the species, also this small signals' list would computationally be inexpensive. With more research and extensive experiments we might be able to identify such signals, and even find out if they have any biological information/relevance.

The third experiment dealt with words that are identified as biological signals, restriction enzymes are known for several usages in biological research, and again these biological signals were able to identify better relatedness among species. Same happened with the fourth experiment where we extracted signals that occur only within the CDs regions for genomic sequences, and these signals were able to outperform traditional methods like MSA in identifying the relatedness between genomes. The conclusion of this work; is that specific available signals with biological relevance would improve the results.

We finally evaluated if such approach would have better results over MSA with datasets that have errors, and again with most of the results we had, these signals have better results and identified better relatedness among species compared to those resulting from MSA. So for genomic datasets that have errors, it would be better to use this approach instead of using traditional alignment-based methods, and that the overlapped signals would identity such relationships among species.

Overlapped signals are powerful marks for comparing biological sequences, and would identify more accurate relationship between species compared to alignment-based methods.

CHAPTER 6

GENE PREDICTION USING COMPARATIVE GENOMICS

Gene prediction deals with identifying stretches within the DNA sequences, these stretches might be subsequences with biological functions, and would go through gene expression and start a specific function for the species. As similar species carry similar functions, these functions are results of similar DNA structures, and although these structures are not exactly identical, but they would still carry a lot of similarities, these similarities could be identified using comparison methods, hence the use of comparative genomics would address such stretches.

The work of this chapter focuses on using LZ complexity as a major filter to search for such similarities between small fragments of the DNA sequences, before applying local alignment as a way of confirmation to this method, then report the found segments as the DNA stretches, and finally evaluate the results against the gold standard of these sequences. The work is compared to a strong tool that was published recently in 2010, PRODIGAL has been developed Oak Ridge National Research Lab [33].

6.1 Methodology

We are conducting several experiments to verify our hypothesis of using comparative genomics; the following steps show how to run each experiment.

1. Run the gene prediction approach based on comparative genomics, which would be by applying the following steps:

   a. Split each sequence to fragments of length L, the fragments should overlap with a period O.
   b. Compare the fragments of the first sequence against the fragments from the second sequence, using Lempel-Ziv complexity [4]; with distance measure 2.
   c. Filter the pair-wise comparison using a predefined threshold S.
   d. Compare the filtered pairs using local alignment [2].
   e. Filter the compared pairs based on local alignment using a predefined threshold LS.
   f. Combine the filtered pairs that are consecutives, into subsequences.
   g. Reports the combined subsequences as predicted active regions for genes.
2. Measure the sensitivity and specificity for the predicted regions using the approach.
3. Run PRODIGAL [33] to predict genes in the genomes.
4. Measure the sensitivity and specificity for the predicted regions using PRODIGAL.
5. Evaluate which method has better results.
6. Tune the parameters of the approach to reach better results.

6.1.1 Gene prediction approach

The approach starts first by splitting the two sequences into fragments of length L, with overlapping period O. Figure 20 shows the splitting process.

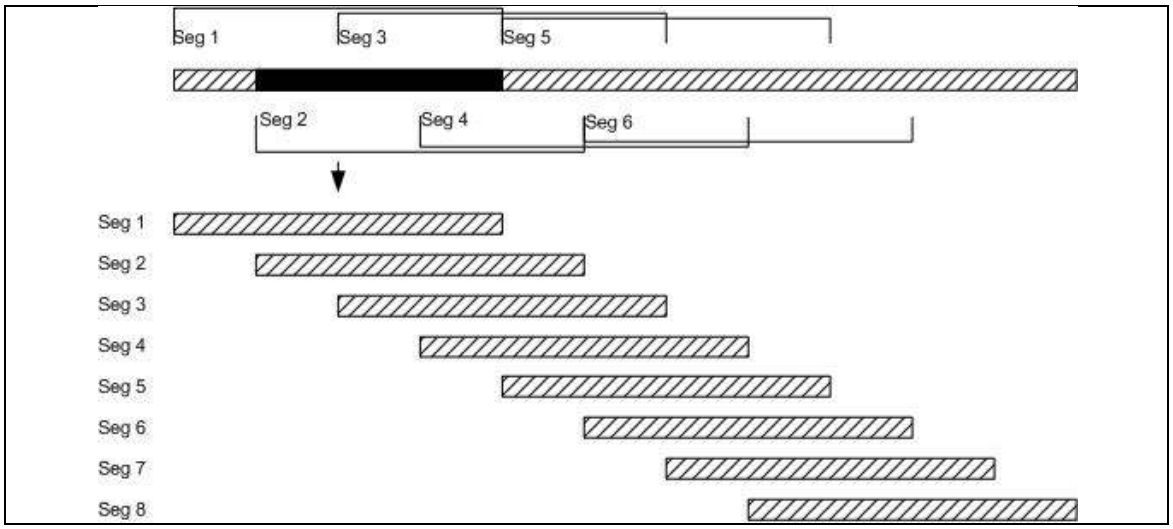

Figure 20 splitting a DNA sequence to fragments of length L, and overlapping period O (which is the shared region between two successive segments), notice the black region is the size of the overlapping.

The next step after splitting the sequences is to measure their LZ complexity. LZ complexity showed a lot of efficiency and high speed in a previous research. It is used as a way of preprocessing, to filter the fragments to the closely related ones. It also has another big advantage over traditional comparison methods (like sequence alignment), which is its heuristic speed. For more details about LZ complexity please refer to the work of Otu et al [4] or section 4.1.2.

Comes next is another filtration for these results. This filtration is based on the scores of the local alignment, which is an application of a threshold applied to the selected pairs. This threshold is the percentage of the scores of the Local alignment to the length of the fragments, and in most cases we used 80%, which would provide a score of 40 for fragments of length 50.

When the strongly related fragments are finally selected using the Local Alignment threshold, we merge those that are consecutives, and that would result in bigger

fragments. Those bigger fragments are our prediction for the Coding Regions in the sequences.

For assessment, a measure of sensitivity and specificity is applied on the predicted genes. Sensitivity means how many accurate genetic nucleotides were predicted, and specificity reveals how much wrong prediction was made.

Figure 21 provides means of definitions for needed terminology of the evaluation, like true positive (TP), false positive (FP), true negative (TN) and false negative (FN).

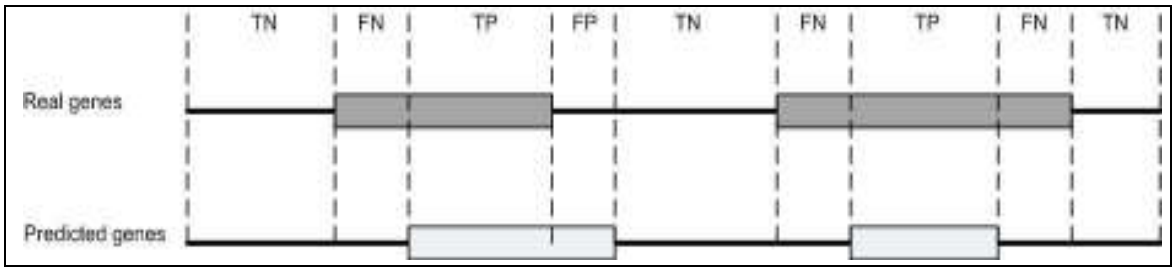

Figure 21 TP is the predicted nucleotides and they are truly within the coding regions, FP is the predicted nucleotides but they are not truly exist in the coding regions, TN unpredicted nucleotides that exist outside the coding regions and FN are the unpredicted nucleotides that exist in coding regions.

6.1.1.1 Accuracy

Accuracy is measured using sensitivity and specificity. Sensitivity measures how much accurate results were measured, while specificity specify how much unneeded data were measured and they are defined as follows:

(Sensitivity) Sp = TP/(TP + FN)

(Specificity) Sn = TP/(TP + FP)

6.1.1.2 PRODIGAL

Prodigal (**Pro**karyotic **Dy**namic Programming **G**enefinding **Al**gorithm) is a microbial (bacterial and archaeal) gene finding program developed at Oak Ridge National Laboratory and the University of Tennessee [33].

In general gene prediction software is case specific; some of them works for microbial genomes, some works for eukaryotes or any specific type of species. In our work, gene prediction is not a case specific, but it focuses on the usage of comparative genomics. We are comparing our approach's results to those of other software like PRODIGAL, so we can evaluate the performance of our approach.

6.1.1.3 Local Alignment [2]

Local alignment was used with the default scoring criteria.

The evaluation process for the results of PRODIGAL is the same as with our approach, that would provide us with a good way to assess the results, and would help us to evaluate the approach, and modify the parameters to provide better results.

The approach parameters are the fragment's length, the overlapping period, the LZ complexity distance's value and the local alignment score. The following bulletin shows the parameters, with a brief discussion on their general effect.

1. Fragment length, this is the length of the used fragments, it should range from 1 (represents one nucleotide) to the full length of the shortest sequence. The best fragment length is subjective and cannot be predicted without experiments.
2. Overlapping period (this is the shared sequence between two successive segments), this is the length of the shared region between two successive fragments, it could be

zero, which means no overlap at all, and up to fragment's length – 1 (L-1). The reason of having the overlapping period; is to take advantage of shared information between successive fragments. The good value for the period is subjective as well, and was left for the experimental work.

3. LZ complexity distance, this value would provide an evidence on the relatedness between fragments. It is a normalized value and scaled from 0 to 1. 0 or close to 0 means closely related fragments, and 1 means distantly related fragments. The smaller the LZ complexity distance's value, the similar the fragments are. Also LZ complexity is a very fast tool of filtering the bad fragments, as it has a linear time complexity.
4. Local alignment, is another verification method for the similarity between fragments, and would help in approving that these fragments are similar and their structures are in order. The measurement of the good local alignment happens by dividing the local alignment score by the length of the fragment. For example if the fragment length is 100 and the local alignment score is 85, then the output would be 0.85, and the bigger the output the closer the fragments. Local alignment was used instead of global alignment, as we were looking at a detailed alignment instead of a generalized one.

The first groups of experiments were conducted to measure the output delivered after changing the parameters. Then second group of experiments was applied with a focus on having best results, with a mind set on how to use the parameters.

6.2 Experimental design

Conducting such research requires running a group of experiments to verify the hypothesis. Each experiment has several steps, and each step would have a different impact on the results. These steps include splitting the sequences to fragments that overlap, filter these fragments by comparing them using LZ complexity, and finally compare the filtered fragments using Local Alignment. These steps lead to the use of 4 parameters, the first one is the fragments' length, and the second one is the overlapping period, and these two parameters can be combined together. The third parameter is the value of the LZ complexity distance, and the fourth parameter is the Local Alignment value.

The rest of steps of any experiment would be the merging of the finely selected fragments that are in order, and those would be the predicted genes. We finally apply assessment procedure based on sensitivity and specificity.

The previous discussion would provide an understanding on what are the needed experiments to conduct such a research. The first issue we needed to address and conduct; is building a sense of the parameters and which values would enhance the results. For this we designed a group of experiments to establish such a sense. Each group of these experiments deals with one parameter, and the basic idea is to fix the values of the other parameters and then change the value of the parameter of choice, then measure the quality of the results and evaluate the performance accordingly. This would provide us with good understanding on how to set the values for the parameters.

Once we establish this sense of the good values for the parameters, we move to the next and main experiment, which is how to use these parameters to predict as much regions as possible of genetic information.

This experiment is designed with a consideration of the good values for the parameters in mind, and with some expectation for the performance of the results accordingly.

6.3 Results and analysis

6.3.1 Analysis of the experimental parameters

6.3.1.1 First group of parameters (fragment length and overlapping period)

The first group of experiments focused on understanding how to use the parameters. This was conducted by fixing all the parameters except one, which would be the one on focus for a specific group of experiments, then analyze the behavior of this parameter was conducted.

To conduct such experiments, we fixed the length of the fragments, and then slowly changed the overlapping period, so an understanding of the effect of the size of the overlapping period would be reached. We used fragments of length 150 nts, and overlapping period of length 75, 120 and 135 nts. The results of sensitivity and specificity showed improvement with the increase of the overlapping period; as shown in Table 6. For verification, same experiment was run several times on fragment's length 100 nts, and with overlapping periods (50, 70 and 90 nts), and also on fragment length 50 nts with overlapping periods (25, 30 and 45 nts). The results also showed that the smaller the fragment's length, the better the results.

The first group of experiments focused on the first two parameters. Smaller fragments length, with bigger overlapping periods would provide better sensitivity and specificity.

Table 6 results of using different fragments lengths, these fragments are the main unit of comparison, and also different overlapping periods, which is the shared sequence between two successive segments, the results show that smaller fragment length and bigger overlapping period would provide better results

Parameters	Seq 1		Seq2		Both Seqs (Total Sn and Sp)	
	Sn	Sp	Sn	Sp	Sn	Sp
150, 75, 0.2, 80%	0.08926	0.24564	0.09547	0.26431	0.09237	0.25497
150, 120, 0.2, 80%	0.15693	0.31158	0.16248	0.3243	0.15971	0.31794
150, 135, 0.2, 80%	0.18847	0.34612	0.19383	0.3578	0.19116	0.35199
100, 50, 0.2, 80%	0.14988	0.32412	0.15547	0.3381	0.15268	0.33111
100, 70, 0.2, 80%	0.20812	0.38282	0.2131	0.39433	0.21062	0.38857
100, 90, 0.2, 80%	0.30884	0.46076	0.31348	0.47036	0.31117	0.46556
50, 25, 0.2, 80%	0.19808	0.3998	0.20338	0.41287	0.20074	0.40637
50, 30, 0.2, 80%	0.2321	0.42486	0.23649	0.43534	0.23434	0.4301
50, 45, 0.2, 80%	0.4262	0.53406	0.43093	0.54275	0.42858	0.5384
Prodigal	0.5991	0.90034	0.48639	0.85621	0.54287	0.88008

6.3.1.2 Second group of parameters (LZ complexity distance)

The second group of experiments dealt with the analysis of the effect of LZ complexity distance. To investigate the effect of these parameters, we fixed the fragment length, overlapping period and local alignment threshold score, and had the LZ complexity changed.

We picked the best parameters we had from the first group of experiments to use in this group of experiments, fragments' length of 50 nts, and overlapping period of 45 nts.

A tuning of the LZ complexity parameter was applied from value of 0.2, 0.25 and 0.3, and evaluation of sensitivity and specificity was analyzed. Results in Table 7 show that with the increase of LZ complexity distance, outputs would be enhanced.

Table 7 results of using different LZ complexity values, the results show that higher values would provide better results

Parameters	Seq 1		Seq2		Both Seqs (Total Sn and Sp)	
	Sn	Sp	Sn	Sp	Sn	Sp
50, 45, 0.2, 80%	0.4262	0.53406	0.43093	0.54275	0.42858	0.5384
50, 45, 0.25, 80%	0.70887	0.62955	0.71236	0.63549	0.71062	0.63252
50, 45, 0.3, 80%	0.84889	0.66354	0.84996	0.66855	0.84942	0.66605
Prodigal	0.5991	0.90034	0.48639	0.85621	0.54287	0.88008

6.3.1.3 Third group of parameters (local alignment score)

The third group of experiments dealt with the local alignment score. Fixation of fragment length, overlapping period and LZ complexity distance was applied, and changing of the local alignment score from 50%, 70%, 80% and to 90% was provided.

Table 8 shows that performance has not changed at all, or was very slight with the change in local alignment score. The reason for this; is that LZ complexity is a strong filter, so with a small value like 0.1 (in our dataset case), the filtered fragments are very similar. These fragments with high similarity would have a high score for local alignment, and with the application of scores less than 90%, no change in the score was reached. While when the score reached 90%, a slight improvement in the specificity was achieved.

Table 8 results of using different percentage of local alignment similarity, this parameter reflect the fine results of a detailed comparison between the different segments, the results show that higher values would provide slightly better results

Parameters	Seq 1		Seq2		Both Seqs (Total Sn and Sp)	
	Sn	Sp	Sn	Sp	Sn	Sp
50, 45, 0.1, 50%	0.05507	0.26732	0.05457	0.26545	0.05482	0.26638
50, 45, 0.1, 70%	0.05507	0.26732	0.05457	0.26545	0.05482	0.26638
50, 45, 0.1, 80%	0.05507	0.26732	0.05457	0.26545	0.05482	0.26638
50, 45, 0.1, 90%	0.05507	0.27316	0.05457	0.27122	0.05482	0.27219
Prodigal	0.5991	0.90034	0.48639	0.85621	0.54287	0.88008

6.3.2 Analysis of the performance of good parameters on the results

After conducting the previous experiments to build a sense on the performance of the parameters, we conducted another experiment to measure the enhancement of these parameters on the performance on the results.

We picked fragments of length 50, and overlapping period of length 45, and fixed the local alignment score to 80%, and gradually increased the LZ complexity from 0.2 to 0.5, so we can measure the gradual performance.

Table 9 shows the improvement of the sensitivity and specificity with the increase of the LZ complexity distance.

This experiment proves that our conclusion of the first group of experiments was correct, and that the right use of these parameters would provide better results for the gene prediction process.

Table 9 results of using different LZ complexity values, the results show the smooth improvement of the results with the increase of LZ complexity

Parameters	Seq 1		Seq2		Both Seqs (Total Sn and Sp)	
	Sn	Sp	Sn	Sp	Sn	Sp
50, 45, 0.2, 80%	0.4262	0.53406	0.43093	0.54275	0.42858	0.5384
50, 45, 0.22, 80%	0.5212	0.57494	0.52413	0.58165	0.52267	0.5783
50, 45, 0.23, 80%	0.62693	0.60786	0.62845	0.61341	0.62769	0.61063
50, 45, 0.24, 80%	0.6546	0.61321	65753	0.61939	0.65607	0.6163
50, 45, 0.25, 80%	0.70887	0.62955	0.71236	0.63549	0.71062	0.63252

50, 45, 0.27, 80%	0.78395	0.64975	0.7848	0.65478	0.78437	0.65226
50, 45, 0.3, 80%	0.84889	0.66354	0.84996	0.66855	0.84942	0.66605
50, 45, 0.5, 80%	0.89629	0.67534	0.8969	0.68013	0.8966	0.67773
Prodigal	0.5991	0.90034	0.48639	0.85621	0.54287	0.88008

6.4 Overall analysis

We conducted several groups of experiments to verify our hypothesis of using comparative genomics in predicting genes. As these experiments have several parameters, it was necessary to understand the nature of these parameters, and then we can use them to maximize the results of predicting genes. These parameters are fragments' length, overlapping period length, LZ complexity distance and local alignment score. Our experiments showed that smaller fragments have better results compared to bigger fragments. But there are computational limitations of using smaller fragments, especially with using big overlapping period parameters, as this would increase the number of generated fragments, and that might take the entire memory of the used computer. This limitation was met when we tried to break the sequences to fragments of length 40 and 30. Although a customization could be applied to the code to overcome such limitation, this customization would slow down the process and would take several months to achieve the output. Also the first groups of experiments showed that bigger overlap would improve the results. The local alignment parameter is the last parameter, and this is the main parameter, as the quality of the local alignment would reflect the similarity between the fragments; with consideration of the order of the

substrings and nucleotides in these fragments. 80% of local alignment was the least value for this parameter, as lower than this value would provide distantly related fragments.

After fixing these parameters for the quality purposes or computational limitations, the only parameters that would enhance the results would be the LZ complexity distance, and the last experiment showed that increasing this parameter would improve the results.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Sequence alignment has gained a lot of trust among researchers, but with the new achievements in the domain, it also showed limitations regarding performance and speed. Compression was suggested as an alternative, and based on its algorithmic structure; compression would address any similarities between any two strings, even if these similarities are not in order.

Compression showed high performance in catching such similarities and identifying the relationships among a group of species. Compression also was able to address the relatedness with datasets that have high level of errors, these errors ranged from random mutations, to incomplete genomes that come as fragments, and these fragments could be one fragment, several fragments, in or out of order.

Another alignment-free technique was based on k-mers, and the major motivation is to address the strength of specific signals, and see how much impact they would have on identifying the relationships between species. These signals could be hidden signals within the sequences, or could be specific motifs that have a biological nature and or significance. The technique showed better results over sequence alignment, starting from using all possible motifs of specific length k, random groups of these motifs, specific motifs that carry known biological information like restriction enzymes, and motifs that occur only in coding regions that have biological function.

Results showed better performance than in sequence alignment, and it showed that specific signals are better than others in finding the relationships between species.

Compression based techniques have the advantage of catching similarities even if it is hard to catch them, they are also fast, and mostly have a linear time complexity. While the motif-based technique has the advantage of using and addressing specific signals in the sequence, these signals would identify the relationships between the sequences, hence the technique would be useful with the advances in biology that would result in knowing more signals, and would be biologically more accurate to find the relationships between species that carry them.

Compression based was also a good source of providing filters for the gene prediction using comparative genomics. Comparative genomics is the base for one of the suggested techniques for gene prediction, which we used to identify specific regions of stretches within the DNA sequences. Compression was accurate to identify closely related segments, and very fast to speed up the process.

Compression is a good tool for comparing biological sequences, and it could be applied for sequences with different level of errors. Hence more work in this domain could be achieved by applying it to different erratic datasets. Same with k-mers approach, as it showed from the results of last experiments, that they would provide better results to erratic datasets than MSA results. Hence a good direction for future research would be by applying the approach to other erratic datasets.

Although the results of gene prediction approach were impressive, we found that the parameters that would provide such good results are not consistent, but they change according to the sequences in use. A good work could be done to find the mathematical

relationship between the good gene prediction parameters and some other parameters, like LZ complexity for the entire genomes, and test if this would provide good estimate for the used parameters, and have them dataset dependent.

REFERENCES

[1] Needleman, Saul B.; and Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". Journal of Molecular Biology 48 (3): 443–53.

[2] Smith, Temple F.; and Waterman, Michael S. (1981). "Identification of Common Molecular Subsequences". Journal of Molecular Biology 147: 195–197.

[3] Vinga S, Almeida J.: Alignment-free sequence comparison-a review., Bioinformatics.;19(4):513-23; 2003

[4] Otu, H.H., Sayood, K.: A new sequence measure for phylogenetic tree construction., *Bioinformatics* Vol. 19 no. 16 2003, Pages 2122-2130

[5] Ferragina, P., Giancarlo, R., Greco, V., Manzini G., Valiente G.: Compression-based classification of biological sequences and structures via the Universal Similarity Metric: experimental assessment. *BMC Bioinformatics* 2007.

[6] Ming Li, Jonathan H. Badger, Chen Xin, Sam Kwong, Paul Kearney, Haoyong Zhang.: An Information Based Sequence Distance and Its Application to Whole Mitochondrial Genome Phylogeny. *Bioinformatics*, 17(2):149-154. 2001.

[7] Burstein D., Ulitsky I., Tuller T. and Chor B.: Information theoretic approaches to whole genome phylogenomics. Proceedings of the ninth annual international conference on research in computational molecular biology (RECOMB 2005). pp.283-295.

[8] Chen X, Kwong S, Li M.: A Compression Algorithm for DNA Sequences and Its Applications in Genome Comparison. In The Tenth Workshop on Genome Informatics, 14-15 December 1999

[9] Behzadi, B. and Le Fessant, F.: DNA Compression Challenge Revisited. In Combinatorial Pattern Matching, 16th Annual Symposium (CPM'05), Jeju Island, Korea, June 19-22, 2005, Proceedings, 2005

[10] E. Rivals, O. Delgrange, M. Dauchet, and J-P. Delahaye.: Compression and sequence comparison. In Proceedings of DIMACS Workshop on Sequence Comparison, 1994.

[11] Mina, R., Ali, H.: A Compression-Based Technique for Comparing Biological Sequences. In Proceedings of the 5th International Biomedical Engineering Conference, Cairo, Egypt (CIBEC 2010).

[12] Guoqing Lu, Shunpu Zhang and Xiang Fang: An improved string composition method for sequence comparison, *from Symposium of Computations in Bioinformatics and Bioscience (SCBB07) Iowa City, Iowa, USA*. 13–15 August 2007,Published: 28 May 2008.

[13] Bernhard Haubold, Nora Pierstorff, Friedrich Möller and Thomas Wiehe: Genome comparison without alignment using shortest unique substrings, *BMC Bioinformatics* 2005, 6:123, Published 23 May 2005

[14] Didier, G., Debomy, L., Pupin, M., Zhang, M., Grossmann, A., Devauchelle, C., and Laprevotte, I..: Comparing sequences without using alignments: application to HIV/SIV subtyping, BMC Bioinformatics Vol. 8 pp. 1

[15] A. Lempel and J. Ziv, "On the complexity of finite sequences," IEEE Trans. Inf. Theory, vol. IT-22, no. 1, pp. 75–81, 1976.

[16] Jacob Ziv and Abraham Lempel; A Universal Algorithm for Sequential Data Compression, IEEE Transactions on Information Theory, 23(3), pp. 337–343, May 1977.

[17] Kolmogoro, A.N. (1965). "Three Approaches to the Quantitative Definition of Information". Problems Inform. Transmission 1 (1): 1–7.

[18] Qi Dai and Tianming Wang, Comparison study on k-word statistical measures for protein: From sequence to 'sequence space, BMC Bioinforamtics, methodology article, 23 September 2008

[19] Mahmood Akhtar, Julien Epps, and Eliathamby Ambikairajah" Signal Processing in Sequence Analysis: Advances in Eukaryotic Gene Prediction", IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 2, NO. 3, JUNE 2008

[20] P.P. Vaidyanathan and Byung-Jun Yoon, Digital filters for gene prediction applications. Proc. IEEE Asilomar Conf. Signals, Systems and Computers, Nov. 2002, Monterey, CA

[21] T. Eftestol, T. Ryen, et al, Eukaryotic gene prediction by spectral analysis and pattern recognition techniques. Signal processing symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic, 2007,146-149

[22] Yang Weng and Yunmin Zhu, Combining Gene-Finding Programs by Using Dempster-Shafer Theory of Evidence for Gene Prediction, IEEE 2006

[23] C. Q. Chang, Peter C. W. Fung, and Y. S. Hung, Improved Gene Prediction by Resampling-based Spectral Analysis of DNA Sequence.Proceedings of the 5th International Conference on Information Technology and Application in Biomedicine, in conjunction with The 2nd International Symposium & Summer School on Biomedical and Health Engineering Shenzhen, China, May 30-31, 2008

[24] Qing Tong, Haoran Zheng, Xufa Wang, A Gene-Prediction Algorithm Based on the Statistical Combination and the Classification in Terms of Gene Characteristics.IEEE 2005

[25] Rong Chen and Hesham Ali, On Gene Prediction by Cross-Species Comparative Sequence Analysis.Proceedings of the 2003 IEEE Bioinformatics Conference (CSB'03)

[26] Sayood, K.: Introduction to Data Compression. 3rd edition, Morgan Kaufmann 2005.

[27] Borowska, M., Oczeretko, E., Mazurek, A., Kitlas. A., Kuć, P.: Application of the Lempel-Ziv complexity measure to the analysis of biosignals and medical images. *Annual proceedings of Medical Science*, Suplement 2, Vol. 50, 2005.

[28] Fisher, R.A.; Yates, F. (1948) [1938]. *Statistical tables for biological, agricultural and medical research* (6th ed.). London: Oliver & Boyd. pp. 37–38

[29] Edgar, Robert C. (2004), MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucleic Acids Research* 32(5), 1792-9

[30] Felsenstein, J.: *Inferring Phylogenies*. Sinauer Associates 2004.

[31] Rajib Sengupta, Dhundy Bastola, Hesham H. Ali: Classification and Identification of fungal Sequences Using Characteristic restriction endonuclease Cut Order., J. Bioinformatics and Computational Biology, 2010: 181-198

[32] Srinivas Aluru: Handbook of Computational Molecular Biology, page 15-10, published 2005

[33] Hyatt D, Chen GL, Locascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010 Mar 8;11(1):119.

BIOGRAPHICAL SKETCH

Name of Author: Ramez Mina

Place of Birth: Kafr El Sheikh, Egypt

Date of Birth: March 27, 1976

Graduate and Undergraduate Schools Attended:
Cairo University, Gizah, Egypt
University of Nebraska at Omaha, Omaha, NE, USA
Degrees Awarded:
Master of Science in Computer Science, 2011, Omaha
Bachelor of Science in Engineering, 2000, Gizah, Egypt

Awards and Honors:

1. Graduate Assistant, 2007 till 2011
2. "First place graduate poster presentation", the first research fair by University of Nebraska at Omaha, March 2009
3. ISMB 2008 Fellowship, funded by NSF.

Publications:

1. "Compression-Based Technique for Comparing Biological Sequences", conference paper in "The Fifth Cairo International Biomedical Engineering Conference 2010", located in Cairo/Egypt.
2. "A New Approach to Compare Biological Sequences based on Motif Alphabets", a poster in the conference "BIOT-2009", Lincoln/Nebraska USA, October 2009.
3. "On Evaluating the Performance of Compression Based Techniques for Sequence Comparison", a poster in the conference "ISMB 2008" located in Toronto/Canada.
4. "Brain-Computer Interface Based on Classification of Statistical and Power Spectral Density Features", a paper in "the Third Cairo International Biomedical Engineering Conference 2006", located in Cairo/Egypt.