Palestine Polytechnic University

College of Administrative Sciences & Informatics

Department of Information Technology

Developing Bioinformatics Approaches to Analyze and Cluster Pathogenic Bacteria Based on Segmental Genomic Duplication

Submitted by:

Amjad Abed Al-Hafith Al-Khateeb

Khaldoun Mohammad Khaled Al-Halawani

Supervisors:

Dr. Yaqoub Ashhab

Dr. Hashem Tamimi

This project is presented as partial fulfillment for requirements of BSc degree in information technology.

2009

Abstract

This project aims to implement the computer science concepts in the biotechnology field. The idea is to apply machine learning algorithms such as Fuzzy C-Means, Subtractive, and genetic algorithms.

A set of pathogenic and non-pathogenic bacterium is selected to be clustered based on its genomic duplication features. The clustering is done by extracting a set of features from the genomic duplication in the DNA sequence of each bacterium. And then the correlation between the clusters and a group of biological features is calculated. To select the best combination of duplication features a genetic algorithm is used, each clustering process is evaluated and fitness is calculated, and the genetic algorithm select the best fitness.

A hierarchical clustering is implemented on each of the duplication features, so we can analyze the feature from one dimension. The output of the hierarchical clustering is analyzed manually.

Acknowledgement

After the finishing of the project, we want to acknowledge many persons, starting by the king Dr.Yaqoub Ashhab for his hard working and brilliant ideas. We want to thank Dr.Hashem Tamimi for his intelligence and for his support.

We will not forget to express our thankfull to Mrs. Asmaa Tamimi for her helping in the biological field.

Thanks to Dr. Mahmoud Al-Saheb and Dr. Mohammad Aldasht for revising and discussing this project.

We will never forget our university, information technology department for their support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

When the science reached a point that it can deal with and manipulate the micros, the biological science does not satisfy the human needs in studying the world of microbiology. At that moment the needs have been increased for a new science that can achieve this goal. Biotechnology is defined as "The scientific manipulation of living organisms, especially at the molecular genetics level, to produce useful products" [1]. The biotechnology has a wide range of fields. One of these fields is Bioinformatics, which is the science that involves the use of techniques and methods including applied mathematics, informatics, statistics, computer science, chemistry and biochemistry to solve biological problems [2].

The major applications of Bioinformatics include genome assembly, looking for similarity between DNA sequences, genes finding using pattern recognition approaches and protein structure alignment.

1.1. The creature's database

The easiest way to think about the genetics is by describing it as a huge amount of information stored in a database that contains all characteristics of the organism. This information comes up from the parents of that organism, and will pass to the offspring. This information is stored in each cell of that organism as a chemical material known as DNA (*deoxyribonucleic acid*), which is the molecule that stores the genetic information. [3]

The DNA is stored in cellular structure called chromosomes; the organisms may have its entire DNA in one chromosome or in more than one. When the organisms multiply to produce new offspring; one pair

of chromosomes is come from each parent. In microorganism when it wants to multiplied a copy of the chromosome is transferred to child/Children.

1.1.1. DNA Structure

As we describe the DNA as huge data storage but it is composed of just four components called nucleotides, these components are: [3] [4]

- Adenine, abbreviated as (A)
- Guanine, abbreviated as (G)
- Cytosine, abbreviated as (C)
- Thymine, abbreviated as (T)

These components are arranged in double helix structure, where each of the components are paired with another one, (A with T) (C with G) and (T with A) (G with C).

The Figure 1.1 illustrates the components of the DNA sequence:

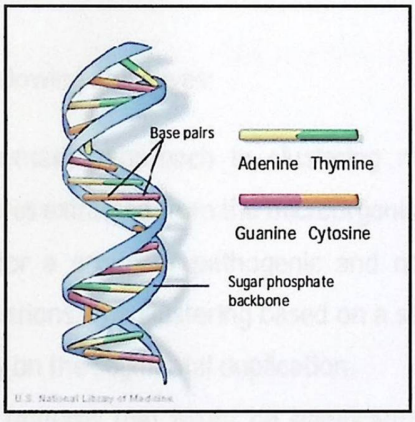

Figure 1.1: The DNA Structure [4]

1.1.2. DNA Replication

The DNA Replication process is done when the microorganisms multiply, where each copy of the DNA is going to one of the daughter cells. In more details, the double helix is broken down into two stands, and a special protein creates the complement part of each stand. During this process errors may happen. These errors are called mutation, which may cause the copying one part of the DNA two times or more.

This mutations specifically duplication play an essential role in the evolution process, If the second copy of the DNA part contain a gene, and the modification of this gene cases it to perform additional or modified function.

1.2. Bacteria and Diseases

Recent reports have shown that several prokaryote (without nucleolus) genomes contain short sequence repeats as well as large scale segmental duplications. [5]

Despite the completion of several pathogenic microorganisms, very little is known about the impact of segmental duplication on the molecular evolution and the adaptation capabilities of these microorganisms. It is conceivable that gene duplication would play a crucial role in the adaptation of pathogenic microbes to the highly changeable environmental conditions such as nutritional status, host immune response, and anti-microbial chemotherapies. A set of characteristics are chosen to be used in this study, these characteristics are biologically relevant to the nature if the bacterium and its ability to case diseases. [6]

1.3. Project Objectives

This project aims to achieve the following objectives:

- Introduce a new Bioinformatics approach in clustering microorganisms: The clustering is performed based on features extracted from the microorganisms' DNA sequences.
- To carry out clustering for a group of pathogenic and non-pathogenic bacteria based on segmental genomic duplications. The clustering based on a set of features extracted from their DNA sequence, specifically on the segmental duplication.
- Find a set of duplication features that might be significantly correlated to a given biological feature among the tested bacteria.

The goal of this project is to study the association between segmental duplications and various biological features of a group of pathogenic (the bacteria that causes diseases) and non-pathogenic bacteria by using data clustering methods like Fuzzy C-Means [6], Hierarchal Clustering [7] [8] and Subtractive Clustering [9] [10].

1.4. Project Hypothesis

The objectives of the project will achieve if these hypothesis are met.

This mutations specifically duplication play an essential role in the evolution process, If the second copy of the DNA part contain a gene, and the modification of this gene cases it to perform additional or modified function.

1.2. Bacteria and Diseases

Recent reports have shown that several prokaryote (without nucleolus) genomes contain short sequence repeats as well as large scale segmental duplications. [5]

Despite the completion of several pathogenic microorganisms, very little is known about the impact of segmental duplication on the molecular evolution and the adaptation capabilities of these microorganisms. It is conceivable that gene duplication would play a crucial role in the adaptation of pathogenic microbes to the highly changeable environmental conditions such as nutritional status, host immune response, and anti-microbial chemotherapies. A set of characteristics are chosen to be used in this study, these characteristics are biologically relevant to the nature if the bacterium and its ability to case diseases. [6]

1.3. Project Objectives

This project aims to achieve the following objectives:

- Introduce a new Bioinformatics approach in clustering microorganisms: The clustering is performed based on features extracted from the microorganisms' DNA sequences.
- To carry out clustering for a group of pathogenic and non-pathogenic bacteria based on segmental genomic duplications. The clustering based on a set of features extracted from their DNA sequence, specifically on the segmental duplication.
- Find a set of duplication features that might be significantly correlated to a given biological feature among the tested bacteria.

The goal of this project is to study the association between segmental duplications and various biological features of a group of pathogenic (the bacteria that causes diseases) and non-pathogenic bacteria by using data clustering methods like Fuzzy C-Means [6], Hierarchal Clustering [7] [8] and Subtractive Clustering [9] [10].

1.4. Project Hypothesis

The objectives of the project will achieve if these hypothesis are met.

1- It is possible to cluster bacteria based on its genomic duplication features.
2- The selected bacteria samples are representative, so we can generalize the results.
3- There is a relationship between the genomic duplication features and the selected biological features.

1.5. Organization of the Document

This document is divided into the following chapters:

- Chapter 1- Introduction: this chapter introduces the purpose of the projects, a brief background about some biological concepts. The idea behind the project, the assumptions and hypotheses of the projects.
- Chapter 2-Background: this chapter aims to provide a clear view and information, about the previous researches and projects in the field. After that all the concepts that the project depends on are discussed and explained. These concepts are MegaBLAST, Data Clustering (K-Means, Fuzzy c-Means, Hierarchical Clustering), Genetic algorithms and Subtractive Clustering.
- Chapter 3-Methodology: in this chapter the details of the algorithms and tools are explained, how these algorithms are used. What is the collected data, and how it is collected. The selected duplication features and biological features.
- Chapter 4-Experements and Results: this chapter includes how the experiments are performed; the Hierarchical Clustering and Fuzzy C-means with the genetic algorithm, all of the details and conditions are demonstrated. And at the end what is the resulting output from these experiments.
- Chapter 5- Conclusions and Future Work: the project goals are reviewed, and what is achieved from these goals, and what is not. Then the future work in the project, and what could be done to gain better results are listed.

Chapter 2

Background

This chapter will be divided into two main parts; in the first part will take a look on the related work and researches. Two subjects will be taken in consideration, the researches that deal with the data clustering and classification field, and the researches that deal with the problem of genomic duplications. The second part of this chapter will cover the theoretical background about the terms, methodologies, algorithms, principles and tools which are used in this project; such as the K-Means, FC-Means, Subtractive clustering, genetic algorithms, MegaBlast and the Hierarchical Clustering concepts.

2.1. Previous Works

As it is clarified previously the idea of applying the clustering on microorganisms based on its DNA sequences, is a new idea in the Bioinformatics field, so the previous researches in this subject are rare, thus, a number of previous projects will be reviewed and they are divided into two subjects. The first is the data clustering in general, and data clustering in some specific field. The second subject is the genomic duplications and feature extraction from the DNA sequences.

2.1.1. Data Clustering Projects

"A Comparative Study of Data Clustering Techniques" [10]

Four clustering techniques have been reviewed which are: K-means clustering, Fuzzy C-means clustering, Mountain clustering, and Subtractive clustering. These approaches solve the problem of categorizing data by partitioning a data set into a number of clusters based on some similarity measure so that the similarity in each cluster is larger than between clusters. [10]

"Optimizing of Fuzzy C-Means Clustering Algorithm Using GA" [9]

The subtractive clustering parameters, which are the radius, squash factor, accept ratio, and the reject ratio are optimized using the GA.

The time needed to reach an optimum through GA is less than the time needed by the iterative approach. Also GA provides higher resolution capability compared to the iterative search.

They also conclude that the time needed for the GA to optimize an objective function depends on the number and the length of the individual in the population and the number of parameter to be optimized.

2.1.2. Genomic Duplication Projects

"Analysis of Distribution Indicates Diverse Functions of Simple Sequence Repeats in Mycoplasma Genomes" [11]

The Mycoplasma family (does not have a cell wall) includes many pathogens of humans and other organisms, is interesting from an evolutionary viewpoint with respect to adaptation to its lifestyle and different modes of interaction with host cells. Mycoplasmas are characterized by vastly reduced genomes, among the smallest of the free-living organisms. Furthermore, Mycoplasmas are diverse in terms of host environment, phenotypic features, as well as genomic characteristics. They feature a reduced number of DNA repair proteins and exhibit high mutation rates, which contributes to the accelerated evolution within the genus. Owing to wide interest from both medical and evolutionary perspectives.

Simple Sequence Repeats (SSRs) composed of extensive tandem iterations of a single nucleotide or a short oligonucleotide are common among Mycoplasma. By contraction or expansion during replication, these SSRs increase genetic variance of the population and facilitate avoidance of the immune response of the host.

SSRs can play various roles depending on their sequence, length, and location. SSRs located in coding regions can function as contingency loci by promoting mutations that influence the rate of transcription initiation. Some SSRs located in protein-coding regions, particularly near the translation start site, can also function as contingency loci by causing frame shift mutations. Other SSRs, especially those involving trinucleotide tandem repeats translated as an amino acid run, can affect structure and function of the encoded protein, particularly with respect to protein-protein interactions. Yet another class of SSRs may affect structural of physical properties of DNA and facilitate proper folding and organization of the

chromosomal DNA in the cell. Perhaps, the most surprising aspect of the SSR comparisons among different Mycoplasma genomes is the apparent functional diversity of the SSRs, which include all the classes listed above. This functional diversity is further underscored by absence of long SSRs in several genomes, specifically M. penetans, M. mobile, and M. synoviae. The apparent functional diversity of SSRs among different Mycoplasma species may have arisen in conjunction with the host cells. [20]

"Short-Sequence DNA Repeats in Prokaryotic Genomes" [12]

Monitoring short sequence repeats (SSRs) enables the study of molecular processes involved in microbial pathogenicity. Regulation of virulence-associated genes, a critical factor in infectious disease progression, can be determined in detail with the help of animal model studies involving site-directed mutants of the pathogen. The really short SSRs seem to be involved mainly in regulation of the expression of genes, whereas the somewhat longer repeat moieties seem to have other functions. The latter structures appear to be mainly involved in implementing size variation in cell wall- or membrane-associated proteins, which may cause enhanced or diminished exposure of active protein domains on bacteria surfaces. In addition, SSR evolution may be a useful feature for monitoring short-term variability in the genome of a large number of medically important microorganisms. The analysis of SSR composition in clinical isolates may in the end be a useful prognosticator of a patient's risk of developing severe infections. [21]

2.2. Genomic Duplication

DNA sequence is read using a method called shotguns method. The method is briefly described as follows:

1. Breaking the chromosomes into short segments using restriction enzymes.
2. Reading the short segments using DNA sequence machine (sequencing).
3. Reading these segments into a nucleated acid strings and assembling them into one sequence.

We can deal with the DNA sequence as a string, and we can detect the genomic duplication in this sequence. Figure 2.1 shows an example of the DNA.

```
TTGACCGATG ACCCCGGTTC AGGCTTCACC ACAGTGTGGA ACGCGGTCGT CTCCGAACTT
AACGGCGACC CTAAGGTTGA CGACGGACCC AGCAGTGATG CTAATCCAGC CGCTCCGCTG
ACCCCTCAGC AAAGGGCTTG GCTCAATCTC GTCCAGCCAT TGACCATCGT CGAGGGGTTT
```

Figure 2.1: DNA Sequence Example

In the assembled genome, the genomic duplications exist when there are two or more similar segments in the genome. The identity (similarity) of the duplicated segments has specified proportion as shown in Figure 2.2.

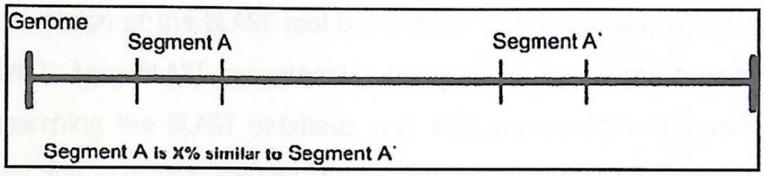

Figure 2.2: Genomic Duplication

There are two main types of the genomic duplication from the view point of the functionality: gene duplications, which are occurred a position where there is a gene, this duplication is also called functional duplication. The second type is the non-functional duplication or the segmental duplication; which are segments of DNA with near-identical sequences in the genome. This kind of duplication is considered as one of the mechanisms for the evolution of new genes through gene duplication, which is occurs in any place in the DNA regardless if there is a gene in that position or not. In this project we deal with the second type of the duplications.

The detection of the genomic duplications is mainly done using sequence alignment algorithms [13] [14]. In this project we are using the MegaBlast sequencing tool [15] to detect the duplications in the DNA sequences.

2.3. MegaBlast

MegaBlast is an enhanced version of another tool called BLAST: "Basic Local Alignment Search Tool", which is a bioinformatics tool that is used to compare biological sequences like amino-acid, protein and DNA sequences. The BLAST tool is developed by Eugene Myers, Stephen Altschul, Warren Gish, David J.Lipman and Webb Miller at the NIH and was published in J. Mol. Biol. in 1990.

Using the BLAST tool we can compare one sequence (Query Sequence) with another sequence or with a set of sequences (Called library or Database) and find the similarity between these sequences.

An example of using the BLAST tool: suppose that a biotechnological scientist has discovered a new gene in some animal or microorganism and he is interested to see if this gene is exist in another animal, the solution is to use the BLAST tool so he can find a similar sequences in a database of sequences. It is possible for blast to carry out sequence alignment either by reading sequences stored in a database called a GeneBank, or a file with FASTA[‡] Format. To produce the FASTA format files we used the 'format' tool which is provided by the NCBI (U.S National Center for Biotechnology information)[§].

MegaBLAST tool is a version of the BLAST tool but it deals with larger sequences and processes them faster than the BLAST. MegaBLAST concatenates many input sequences together to form a large sequence before searching the BLAST database, and then post-analyze the search results to glean individual alignments and statistical values. The MegaBLAST tool can be downloaded from the NCBI website. [15]

2.3.1. Sequence Alignment

The implementation of BLAST and MegaBLAST tool depends on the sequence alignment algorithm. In this section we will make an overview on the sequence alignment.

"Sequence alignment is a way of arranging the sequences such as DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences." [13]

Word method

This is a heuristic method that does not guarantee finding an optimal alignment solution, but is significantly more efficient than dynamic programming [14]. This method is especially useful in large-scale database searches. Word methods identify a series of short, non-overlapping subsequences (words) in the query sequence that is matched to database sequences. The relative positions of the word in the

[‡] FASTA Format: is file format that starts with single line description that start with the ">" mark, this line is followed by the DNA sequence data. It is recommended that each line contain 80 characters at most. The file is stored in ASCII encoding so it could be opened using any text editor [22].

[§] Established in 1988 as a national resource for molecular biology information, NCBI creates public databases, conducts research in computational biology, develops software tools for analyzing genome data, and disseminates biomedical information - all for the better understanding of molecular processes affecting human health and disease.

Using the BLAST tool we can compare one sequence (Query Sequence) with another sequence or with a set of sequences (Called library or Database) and find the similarity between these sequences.

An example of using the BLAST tool: suppose that a biotechnological scientist has discovered a new gene in some animal or microorganism and he is interested to see if this gene is exist in another animal, the solution is to use the BLAST tool so he can find a similar sequences in a database of sequences. It is possible for blast to carry out sequence alignment either by reading sequences stored in a database called a GeneBank, or a file with FASTA[‡] Format. To produce the FASTA format files we used the 'format' tool which is provided by the NCBI (U.S National Center for Biotechnology information)[§].

MegaBLAST tool is a version of the BLAST tool but it deals with larger sequences and processes them faster than the BLAST. MegaBLAST concatenates many input sequences together to form a large sequence before searching the BLAST database, and then post-analyze the search results to glean individual alignments and statistical values. The MegaBLAST tool can be downloaded from the NCBI website. [15]

2.3.1. Sequence Alignment

The implementation of BLAST and MegaBLAST tool depends on the sequence alignment algorithm. In this section we will make an overview on the sequence alignment.

"Sequence alignment is a way of arranging the sequences such as DNA, RNA, or protein to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences." [13]

Word method

This is a heuristic method that does not guarantee finding an optimal alignment solution, but is significantly more efficient than dynamic programming [14]. This method is especially useful in large-scale database searches. Word methods identify a series of short, non-overlapping subsequences (words) in the query sequence that is matched to database sequences. The relative positions of the word in the

[‡] FASTA Format: is file format that starts with single line description that start with the ">" mark, this line is followed by the DNA sequence data. It is recommended that each line contain 80 characters at most. The file is stored in ASCII encoding so it could be opened using any text editor [22].

[§] Established in 1988 as a national resource for molecular biology information, NCBI creates public databases, conducts research in computational biology, develops software tools for analyzing genome data, and disseminates biomedical information - all for the better understanding of molecular processes affecting human health and disease.

two sequences being compared are subtracted to obtain an offset; this will indicate a region of alignment if multiple distinct words produce the same offset.

In FASTA format, the user defines a value k to use as the word length with which to search the database. BLAST uses a word search of length k and evaluates only the most significant word matches. Most BLAST implementations use a fixed default word length that is optimized for the query and database type, and that is changed only under special circumstances, such as when searching with repetitive or very short query sequences. [13]

2.4. Data Clustering

In machine learning, clustering is the process of distributing a set of samples (observations) into a set of clusters (groups) according to shared features between the samples. [6]

The features of the samples are a set of characteristics that this sample is carry, and these features are represented numerically. Each sample in a population have one or more features, when arrange the features of each sample in a row we get a feature vector for this sample. One feature vector may contain the data of more than one sample (each sample in a row) so we can use it in the clustering process.

There are many clustering algorithms, such as: Support Vector Machine (SVM) [16], Self Organizing Map (SOM) [17], K-Means [18], Fuzzy C-Means [6] and Subtractive Clustering [9] [10].

The clustering process is carried out through steps illustrated in Figure 2.3: [19]

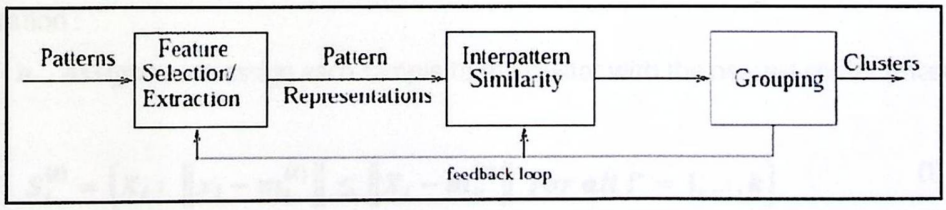

Figure 2.3: Clustering Process

(1) Feature Selection/Extraction: either the selection of the best features in the feature vector is selected, or the extraction of new features based on the entered features is done.

(2) Pattern representation: determining the number of clusters, the patterns used in the clustering process.

(3) Calculate the similarity between samples using one of the distance measures like (Euclidean Distance).

(4) Making the clustering or grouping process.

2.4.1. K-Means Clustering: (18)

In this clustering methodology, mapping an unknown feature to certain cluster is done by measuring the distance between the cluster center and the sample, and then each sample belongs to the cluster with the nearest mean.

The following steps are applied:

1. First random centers of the clusters are generated.
2. Then the distance from the clusters centers to the position of each sample is calculated.
3. The samples are grouped according to the shortest distance from the clusters center.
4. The centers are recalculated for each cluster according to the samples distribution in the cluster.
5. The steps 2 to 4 are repeated until no changes on the mapping f samples

In more details: given a set of samples (X_1, X_2...X_n) each sample has a d-dimension feature vector; the purpose of the K-means clustering is to distribute these samples into a K Clusters S= {S_1, S_2...S_K}. Where the number of clusters is less than the number of samples: (K<n).

Basically the K-Means algorithm pass throw the following steps iteratively:

- Initialization: generate random K center (m_1, m_2...m_k) for the clusters.
- Iteration :
  - ***Assignment***: assign each sample to the cluster with the nearest center (mean)

$$S_i^{(t)} = \left\{ X_j : \left\| x_j - m_i^{(t)} \right\| \leq \left\| X_j - m_{i^*}^{(t)} \right\| \ \textit{for all } i^* = 1, \ldots, k \right\} \quad (1)$$

  - ***Update Step***: Calculate the new mean of each cluster, according to the samples in these cluster

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_i \quad (2)$$

- The algorithm stop when it reaches to the convergence point, which mean there is no change between iteration updates.

Figure 2.4 demonstrates the evolution of K-Means clustering:

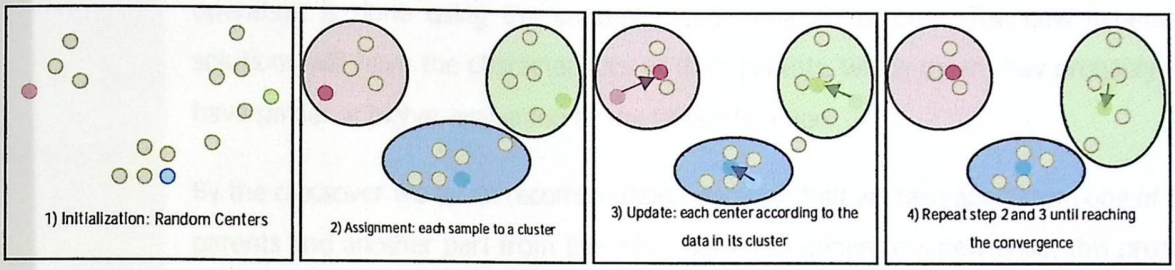

Figure 2.4: Clustering Example

2.4.2. Fuzzy C-Means Clustering

The Fuzzy C-Means algorithm is very similar to the K-Means algorithm but in the fuzzy clustering each point (sample) has a belonging degree to a cluster rather than belonging completely to a cluster, which means that a point on the edge of a cluster it is less belonging to that cluster than the points on its center. [6]

2.5. Genetic Algorithms

In general the genetic algorithm is a search technique used to find an optimal solution in space of solutions. The genetic algorithm simulates the evolution of the creatures in the suggestion of its solutions (Genetic crossover process), and it uses a heuristic search approach to find the solution, this heuristic need a fitness function to evaluate each case (solution). [20]

2.5.1. The genetic algorithm steps:

- ***Initialization***

  In the first step of the genetic algorithm; a set of solutions generated randomly or by user, these solutions are considered the first or initial generation.

- ***Selection***

  After the generation is created a set of solutions is selected from that generation, the selection is done according a fitness function that evaluate each solution. In general the genetic algorithms try to minimize or maximize a fitness value.

- ***Re-generation***

  The selected solutions in the previous step are used to generate new solutions. This operation is done using the crossover and mutation process. The new generated solutions will have the characteristics of their parents, which mean they probably will have similar or higher evaluation by the fitness function.

  By the crossover we mean recombination operation that we take apart from one of the parents and another part from the other parent to generate a new child, this process happens in natural chromosomes in the organisms when it is multiplied. A simple (one point) crossover is illustrated in the Figure 2.5: [21]

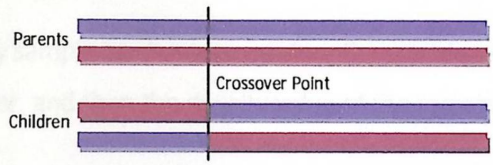

Figure 2.5: Single Point Crossover

Other types of the crossover are two point crossover, cut and split crossover and others.

The Mutation process involves changing the properties of the chromosomes randomly in order to generate solutions widely different from the parents; this will reduce the possibility of falling into local minima problem. [21]

- ***Termination***

  The previous process is executed until a termination condition is occurred, this condition may be either solution that satisfy some condition found, or a given number of generations is reached or other termination conditions.

2.6. Subtractive Clustering

The subtractive clustering method assumes that each sample is a potential cluster center. A sample with more neighboring data will have a higher opportunity to become a cluster center than samples with fewer neighboring data. The algorithm is: [9]

- Select the samples with the highest density to be the first cluster center.
- Remove all samples that belong to the first cluster in order to determine the next data cluster and its center location.
- Iterates on this process until all of the data is within the radius of a cluster center.

The density value for each sample is calculated as follows:

$$D_i = \sum_{j=1}^{n} exp\left(-\frac{\|x_i - x_j\|^2}{(r_a/2)^2}\right) \quad (3)$$

Where x_I and x_J are samples and r_a is the neighborhood radius.

After the density value of every sample has been computed, the sample with the highest density value is chosen as the first cluster center, and then the density value of the remaining samples is revised by:

$$D_i = D_i - D_{c1}\, exp\left(-\frac{\|x_i - x_{c1}\|^2}{(r_b/2)^2}\right) \quad (4)$$

Where x_{c1} is the first cluster center, D_1 its density value and r_b is a positive constant which defines a neighborhood that has measurable reductions in density measure. Therefore, the samples near the first cluster center x_{c1} will have significantly reduced density measure, and unlikely to be selected as next cluster center.

After revising the density, the sample with the highest density value is selected as next cluster center. [10]

2.7. Hierarchical Clustering [7] [8]

Hierarchical clustering groups a data over a variety of scales by creating a cluster tree or dendrogram. The tree is not a single set of clusters, but it is a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. So the scale or level of clustering can be varied to get the most appropriate level.

Hierarchical clustering is performed on a data set through these steps:

1- Find the similarity or dissimilarity between every pair of objects in the data set. This process is also called linkage, which is done according to various methods such as:
The cluster r is formed from other two clusters p and q; n_r is the number of objects in the cluster r x_{ri} is the i^{th} object in cluster r. the linkage could be calculated as one of the following.
-Single linkage (nearest neighbor) which uses the smallest distance between the objects

$$d(r,s) = min\left(dist(x_{ri}, x_{xj})\right), i\epsilon(i,\dots,n_r), j\epsilon(1,\dots,n_s) \quad (5)$$

-Complete linkage or furthest neighbor which is gets the largest distance

$$d(r,s) = max\left(dist(x_{ri}, x_{xj})\right), i\epsilon(i,\dots,n_r), j\epsilon(1,\dots,n_s) \quad (6)$$

- Average linkage: which is uses the average distance between the objects:

$$d(r,s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_r} dist(x_{ri}, x_{sj}) \quad (7)$$

-Centroid linkage that uses the Euclidean distance between the centers of the clusters:

$$d(r,s) = \|\tilde{x}_r - \tilde{x}_s\|_2 \quad (8)$$

- And other methods like: Median linkage, ward linkage.

2- Group the objects into a binary, hierarchical cluster tree. In this step, pairs of objects that are in close are linked depending on distances calculated in step 1 to determine the proximity of objects to each other. As objects are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed.

3- Determine where to cut the hierarchical tree into clusters. In this step, the objects in the tree are clustered by assigning all the objects below each cut to a single cluster. The clusters could be created by detecting natural groupings in the hierarchical tree or by cutting off the hierarchical tree at an arbitrary point.

2.7.1. Dendrogram

A dendrogram consists of many U-shaped lines connecting objects in a hierarchical tree. The height of each U represents the distance between the two objects being connected. Some leaves in the plot may correspond to more than one data point if the data set is large.

Ex: Here is an example that shows how the Hierarchical clustering works, on a data set of five samples each has two features. Let's assign each sample to an object as shown in Table 2.1:

Table 2.1: Hierarchical Clustering Example

Object 1	1,2
Object 2	2.5, 4.5
Object 3	2, 2
Object 4	4, 1.5
Object 5	4, 2.5

We can represent the distance between the objects by the Figure 2.6 where the Euclidean distance is used.

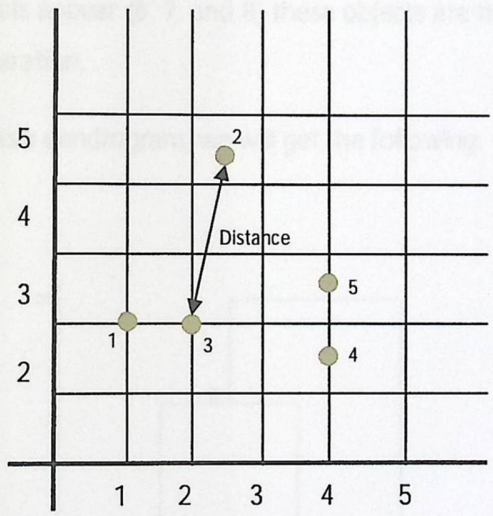

Figure 2.6: Euclidean Distance between samples

After calculating the distances we are able to link the objects together in a hierarchical linking. As illustrated in Figure 2.7:

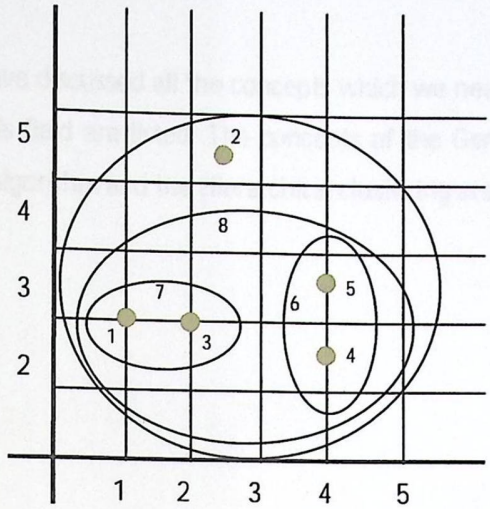

Figure 2.7: Hierarchical Objects

You can notice that new objects appear (6, 7, and 8) these objects are new formed clusters that will be entering in a new clustering iteration.

Finally if we draw the output as a dendrogram, we will get the following:

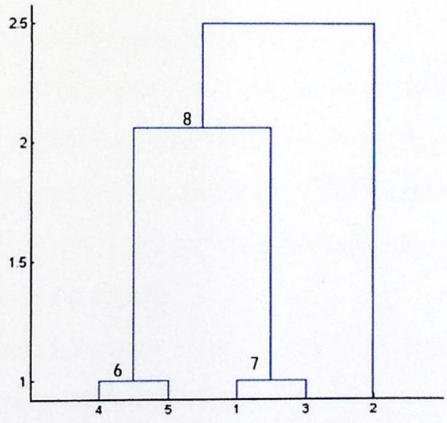

Figure 2.8: Dendrogram

2.8. Summary

In this chapter, we have discussed all the concepts which we need in this project. A summary of the previous works in this field are listed. The concepts of the Genomic Duplication, MegaBLAST, Data Clustering, Genetic algorithm and the Hierarchical clustering are explained.

Chapter 3

Methodology

In this chapter we talk about the methodologies used in this project. The chapter covers the collected data, how it is collected, and the selected features, why these features are selected, and how it is extracted from the data. The chapter covers the tools that are used in the project, why these tools are used and the suggested evaluation of the results.

The development of the project passes through these steps:

- Define the problem and the needed data to solve it: the problem of bacteria clustering based its DNA, the DNA sequences and biological features.
- Data collection and preparation: where the DNA sequences will be downloaded,
- Features Extraction: Duplication Features Extraction and Biological Features Collection.
- Applying the Clustering: Fuzzy C-Means and Subtractive clustering.
- Features Selection and Evaluation: using the genetic algorithm the features that have the largest impact on the clustering will be chosen.
- Applying the Hierarchical Clustering on the genome duplication features and analyzing the output manually by a biology specialist.

The application of the Hierarchical Clustering is done on each dimension (feature) alone. The Fuzzy C-Means is used to cluster the multi-dimensions feature vector because we cannot recognize the distribution of the data in high dimensionality space.

3.1. Problem Definition

As described in Chapter 1, the problem is to study the association between segmental duplications and various biological features of a group of pathogenic and non-pathogenic bacteria by using deferent data clustering methods.

One can understand from this description that first we need to prepare a training set of a pathogenic and non-pathogenic bacterium, and then analyze the DNA sequence of each bacterium, we mean by the analysis that the segmental duplication in each DNA sequence must be found, and extract a set of features from these duplications. Then, we need to apply a clustering algorithm on a set of features that achieves high correlation with the biological features.

3.2. Data Collection

It is clear now that the study is about the pathogenic and non-pathogenic bacterium, which means that the genome (DNA sequence) for a large population of that bacterium is needed. When searching in the NCBI which is the largest and widest resource about the biotechnological information, we found that the complete genome sequencing projects for that bacterium is about 75 bacteria sequence.

The sequences of this bacterium are downloaded in FASTA format files that contain a string of characters which represents the nuclides of the bacterium DNA. The files are downloaded using the **`getgenbank`** command (Bioinformatics Toolbox) in the MATLAB[**]. With each genome we get the length of that genome (the number of bases (nucleotides) in the DNA sequences) using the same command.

3.2.1. Duplication Detection

The next step is to find the duplicated segments in the genome of each bacteria. To do this process we used the MegaBLAST tool, In general the MegaBLAST tool is used to detect the similarity regions between one sequence and a large database of sequences. But in our case we use this tool to detect the similarity regions in the same sequence, so we provide the MegaBLAST tool with the DNA sequence as a query sequence and as a database. Note that each sequence or file has identification ID. The MegaBLAST requires a set of parameters in order to work properly, the main parameter is the word size parameter which is 28 by default.

[**] MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and FORTRAN. It is also provide a wide range of Toolboxes and functions in the most science fields.

After applying MegaBLAST on all the sequences, we get the results for each genome as a single file which contains all duplicated segments represented using the fields listed in Table 3.1:

Table 3.1: MegaBLAST Output Fields

Filed	Description
`Query id`	Represent the ID of the query sequence or file
`Subject id`	Represent the ID of the sequence in the Database
`% identity`	The ratio of similarity between the two segments
`alignment length`	The length of segment (duplicated segment)
`mismatches`	Number of characters where the character in the first segment is not matched with the character in the same position in the second segment
`gap openings`	The number of gaps
`q. start`	The start position of the first segment in the query sequence
`q. end`	The end position of the first segment in the query sequence
`s. start`	The start position of the second segment in the subject sequence
`s. end`	The end position of the first segment in the subject sequence

3.2.2. Duplication Filtering

Although the previous results contains all the duplicated segments in the sequence, but still there is a problem of a very huge rubbish results such as the similarity of the segment with itself, and the transitive duplication: which can be considered as an ambiguous results.

To solve these problems; a data filtering is needed. This data filtering is applied a three levels filtering.

-***Level 1***: Filters the results that have an identity (descried in Table 3.1) less than a given thresholdτ. This enables us to maintain a high similarity between the segments and to neglect the segments that have low similarity.

-***Level 2***: Filters the results in the MegaBLAST output file, where segments in the first sequence are similar to the same segment in the second sequence.

Ex: If the MegaBLAST output file contains such result (100-200 == 100-200), this means that in the first sequence the segment 100-200 is similar to the segment 100-200 in the second sequence, which means that the segment is similar to itself, as illustrated below.

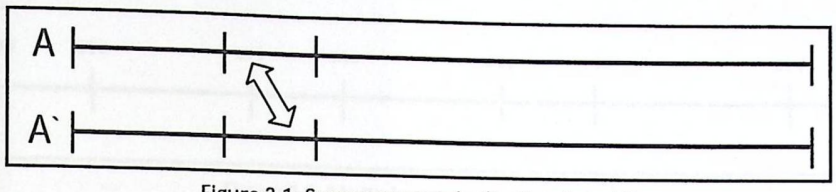

Figure 3.1: Same segment duplication (Level 1)

-Level 3: Filters one of two results in the MegaBLAST output file when there is two results represent the same segment (commutative relation).

Ex: In a sequence if the Segment 100-200 is similar to the segment 500-600, there will be two results in the MegaBLAST output file, (100-200 == 500-600 AND 500-600 == 100-200) -as illustrated in Figure 3.2-. This means we need to delete one of them.

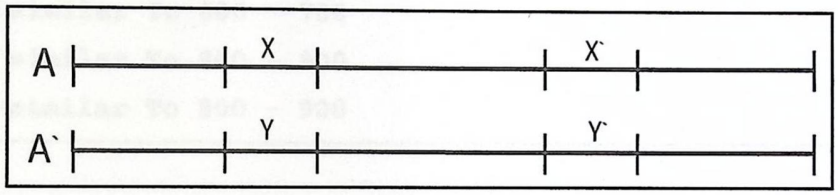

Figure 3.2: Commutative relation duplication

In this Figure, we need to delete the record (Y == X`). Note that the records (X==Y) and (X`=Y`) was deleted in the previous step.

3.3. Features Extraction

Now we have a set of records that contain the information about the genomic duplications, which means that we can start extracting the features from these records. The first thing we need in the extraction of the features is to count the genomic duplications.

3.3.1. Segments Counting

We mean by counting the segments that the number of the similar segments with about same length and content (similarity larger than 95%).

Ex: In the Figure 3.3 if the segments 100-200[††], 400-500, 600-700 and 800 -900 are similar, then the number of segments is to be 3. The filtered MegaBLAST output file will contain at most six recorders for these segments.

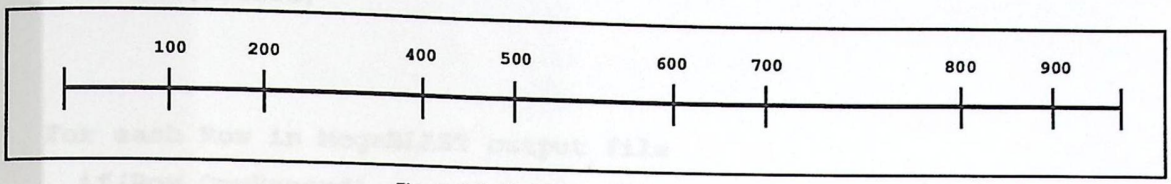

Figure 3.3: Multiple segments duplication

The result is a records listed in Table 3.2:

Table 3.2: Multiple segments duplication output

```
100 - 200  similar To 400 - 500
100 - 200  similar To 600 - 700
100 - 200  similar To 800 - 900
400 - 500  similar To 600 - 700
400 - 500  similar To 800 - 900
600 - 700  similar To 800 - 900
```

Now to do the counting process the following algorithm is used:

```
SegmentsCounting()
{
   For each Row in MegaBLAST output file
   Delete(Row)
   X=insertInID(Row.Q,-1)
   insertInID(Row.S,x)
   End For
}

insertInID(Record, id)
{
   If(id=-1)
```

[††] We mean by 100 – 200 the Start and End positions of the segments in the sequence.

```
    createId(NewId)

  if(Record Not Exist In The id)
    id.Add(Record)


  For each Row in MegaBLAST output file
    if(Row.Q==Record)
    {
       Delete(Row)
       insertInID(Row.S,id)
    }
    Else if(Row.S==Record)
    {
       Delete(Row)
       insertInID(Row.Q,id)
    }

  End For
  Return id
}
```

Algorithm 3.1: Segments Count Algorithm

Where Row.Q represent [Row.QueryStart, Row.QueryEnd] and Row.S represnet [Row.SubjectStart, Row.SubjectEnd], and the Record is a struct that contain [QueryStart, QueryEnd] or [SubjectStart, SubjectEnd] according to Table 3.1.

The output of this process is a table that contains a group of IDs each id represents a unique segment in the sequence, and the number of copies for of each segment.

3.3.2. Duplications Features

At this moment, we are able to start extracting the features from the previous information. We suggest a set of duplication features that cover a wide set of interest points in the duplications. These features are:

1) ***Number of Duplications***

Determine the amount of duplications in the DNA sequence (normalized number of duplications), relative to the length of the sequence. This feature is computed as:

$$Amount = \frac{Number\ of\ Duplications}{length\ of\ sequence} \quad (9)$$

2) ***Average length of duplications (Mean)***

Determine the average length of all duplications in the DNA sequence.

$$Mean = \frac{\sum_{i=1}^{N} Length\ of\ Segment_i}{N} \quad (10)$$

Where N is the Number of segments (IDs).

3) ***Length of genome***

This feature represents the actual length of the DNA sequence, where the length is number of nucleotides (bases) in the sequence.

4) ***Duplications Distribution Histogram***

This feature gives us an indication about the distribution of duplications over each sequence. Which mean where the duplications are concentrated and where are not.

To implement this feature, we divide the genome into fixed number of regions (Bins)

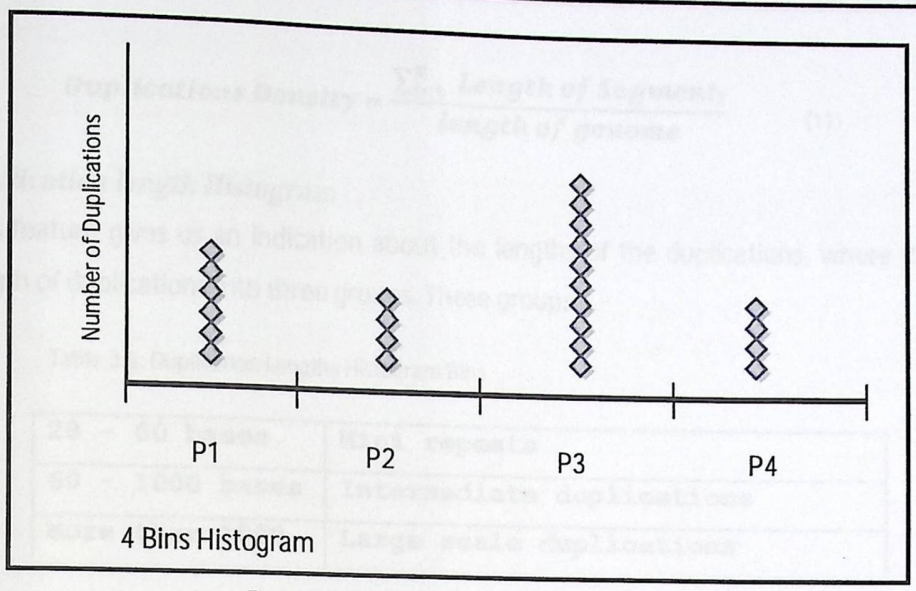

Figure 3.4: Duplication distribution histogram

After that we find the number of duplication in each bin as Illustrated in Figure 3.4: Duplication distribution histogram, and then the number of duplications in each bin is converted to a ratio relative to the total number of duplications. In order to generate the histogram, we use the following algorithm:

$BinLen$ = Genome Length / Number of Bins

For each segment of in the duplication results

$$center = \frac{Segment\ End - Segment\ Start}{2} + Segment\ Start$$

$$Bins\left[\frac{center}{BinLen}\right] + +$$

End for

Algorithm 3.2: Calculating duplication distribution histogram

5) *Duplication Density*

This feature gives us a percent of how much of the genome content is duplicated, which means the ratio between the duplicated segments length and the sequence length itself. This feature is calculated as following:

$$Duplications\ Density = \frac{\sum_{i=1}^{N} Length\ of\ Segment_i}{length\ of\ genome} \quad (11)$$

6) Duplication length Histogram

This feature gives us an indication about the lengths of the duplications, where it divides the length of duplications into three groups. These groups:

Table 3.3: Duplication Lengths Histogram Bins

28 – 60 bases	Mini repeats
60 – 1000 bases	Intermediate duplications
More than 1000	Large scale duplications

These lengths are selected based on the nature of the duplications, where the mini repeats (28-60 bases) are popular in the bacterium that is changing frequently, and the duplication lengths larger than 1000 usually represents gene duplication. The duplication length histogram feature is extracted in the same way as the Duplications Distribution Histogram (feature 4).

Finally all the features are arranged in one feature vector shown in Table 3.4:

Table 3.4: Duplication Feature Vector

Feature No	Description
DF1	Number of Duplications
DF2	Average length of duplications (Mean)
DF3	Length of genome
DF4	Bin1 of Duplications Distribution Histogram
DF5	Bin2 of Duplications Distribution Histogram
DF6	Bin3 of Duplications Distribution Histogram
DF7	Bin4 of Duplications Distribution Histogram
DF8	Duplication Density
DF9	Bin 1 of Duplication length Histogram
DF10	Bin 2 of Duplication length Histogram
DF11	Bin 3 of Duplication length Histogram

Table 3.5 shows 10 examples of feature vectors for 10 different bacteria:

Table 3.5: Feature Vector Sample for 10 bacteria

	Acc No	DF1	DF2	DF3	DF4	DF5	DF6	DF7	DF8	DF9	DF10	DF11
1	NC_000908	10.00	94.64	580076	0.052	0.414	0.379	0.155	0.009	0.517	0.483	0.000
2	NC_000912	60.39	176.72	816394	0.396	0.158	0.371	0.075	0.107	0.337	0.651	0.012
3	NC_001318	3.07	725.79	910724	0.571	0.214	0.214	0.000	0.022	0.143	0.714	0.143
4	NC_002162	17.29	387.10	751719	0.115	0.046	0.831	0.008	0.067	0.154	0.831	0.015
5	NC_002771	42.12	594.97	963879	0.025	0.103	0.791	0.081	0.251	0.148	0.670	0.182
6	NC_002950	47.54	418.22	2343476	0.321	0.271	0.256	0.152	0.199	0.571	0.325	0.104
7	NC_002951	33.85	289.36	2809422	0.187	0.215	0.343	0.256	0.098	0.517	0.441	0.042
8	NC_002967	49.35	98.88	2843201	0.388	0.096	0.416	0.100	0.049	0.644	0.344	0.012
9	NC_004311	20.21	85.15	1207381	0.414	0.139	0.324	0.123	0.017	0.533	0.463	0.004
10	NC_004342	287.33	163.06	4332241	0.237	0.189	0.263	0.311	0.469	0.643	0.319	0.038

3.3.3. Biological Features

The second type of features that we need to collect is the biological features, which describe the general behavior and characteristics of pathogenic and non-pathogenic bacteria. These features are:

Table 3.6: Set of Biological Features

Biological Feature	Description
Gram stain	The nature of the cell wall
Respiration	The rate of dependence on oxygen
Extracellular/Intracellular	Living inside or outside a cell
Generation time	period of time required to double in size
GC Content	percentage of guanine and cytosine
% Coding	Percentage of DNA that translates to protein
Topology	DNA structure
Genes	Number of genes in the DNA sequence
Protein coding	Number of genes translated to proteins
Pseudo genes	The genes that does not expressed in the cell

These features (in Table 3.6) are collected through reviewing a number of researches which are done by a biological researcher.

3.4. Applying the Clustering

Two types of clustering are applied; the Hierarchical Clustering is done on the each feature alone. And the Fuzzy C-Means Clustering is applied on the feature vectors for all of the samples.

3.4.1. Hierarchical Clustering

The Hierarchical Clustering is applied on the genomic duplication features, the output of the Hierarchical Clustering will be analyzed manually using the dendrogram of each feature. So it may be used to get a biological meaning from the process of grouping the samples in a tree structure.

On the other hand, the application of the Hierarchical Clustering is done to explore the nature of the data and its ability to be clustered and gathered in groups.

The output of the Hierarchical Clustering for each of the duplication features is illustrated in chapter 4.

3.4.2. Fuzzy C-Means

In order to achieve the objectives of bacteria clustering based on its genomic duplications features, the Fuzzy C-Means clustering approach is chosen to be applied on the selected feature vector.

As mentioned before, the inputs of the Fuzzy C-Means clustering are the feature vector and the number of clusters. The output of this process is a vector that contains the distance for each sample from the center of each cluster. Each sample is belonging to cluster with shortest distance from that cluster.

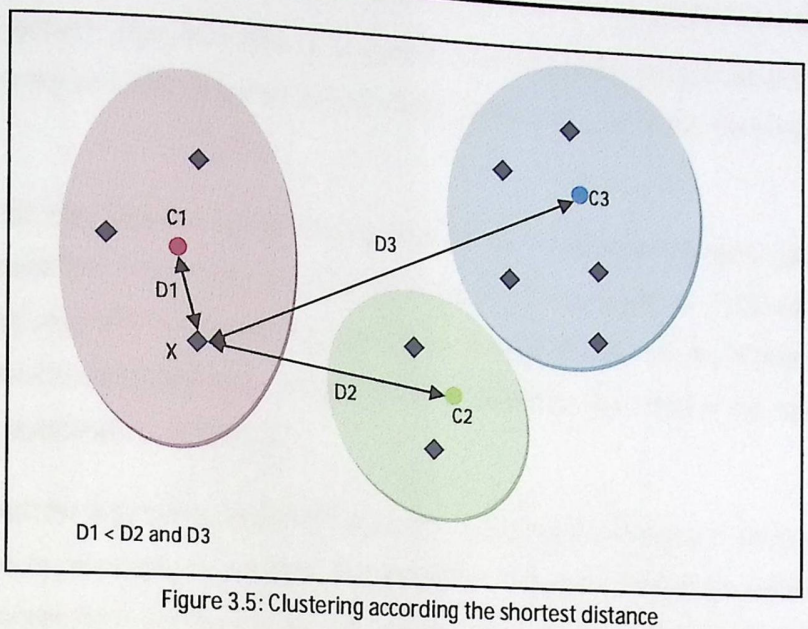

Figure 3.5: Clustering according the shortest distance

In the Figure 3.5 the Distance between sample X and the cluster center C1 is less than the other distances.

The Fuzzy C-Means need the number of clusters as input, but here we did not know the number of clusters, so we need a mechanism that determines it depending on the data itself. We choose the subtractive clustering to decide the best number of clusters. The subtractive clustering inputs are the feature vector in addition to the radius of the clusters. The output is the number of clusters in these data.[9]

At this moment, we have two choices to apply the subtractive clustering, the first one is by training all the possible radiuses which are enclosed between 0 and 1. The second alternative is to test a determinant set of radiuses that gives us a different number of clusters. In this project we choose the second alternative, where a set of 4 radiuses are tested and the results will be evaluated later!

3.5. Features Selection and Evaluation

To get the best clustering results, the features which should enter in the clustering process must have the most effect in it. It is not possible to try all the combinations between all the features manually, to solve this problem we use genetic algorithm, for features selection.

The eleven features in Table 3.4 are represented as a string of 0s and 1s, where the 0 value means that this feature will not be used in the clustering process, and the features represented by 1 will be used.

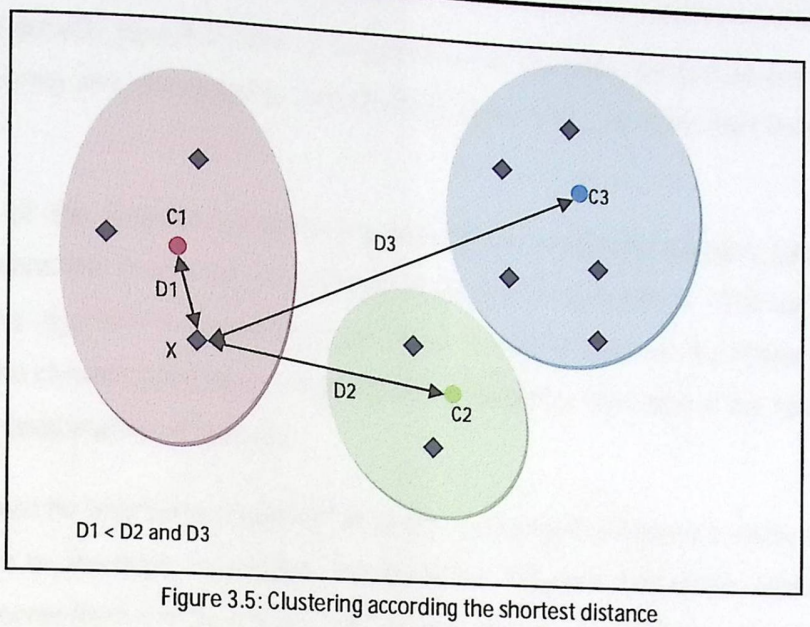

Figure 3.5: Clustering according the shortest distance

In the Figure 3.5 the Distance between sample X and the cluster center C1 is less than the other distances.

The Fuzzy C-Means need the number of clusters as input, but here we did not know the number of clusters, so we need a mechanism that determines it depending on the data itself. We choose the subtractive clustering to decide the best number of clusters. The subtractive clustering inputs are the feature vector in addition to the radius of the clusters. The output is the number of clusters in these data.[9]

At this moment, we have two choices to apply the subtractive clustering, the first one is by training all the possible radiuses which are enclosed between 0 and 1. The second alternative is to test a determinant set of radiuses that gives us a different number of clusters. In this project we choose the second alternative, where a set of 4 radiuses are tested and the results will be evaluated later!

3.5. Features Selection and Evaluation

To get the best clustering results, the features which should enter in the clustering process must have the most effect in it. It is not possible to try all the combinations between all the features manually, to solve this problem we use genetic algorithm, for features selection.

The eleven features in Table 3.4 are represented as a string of 0s and 1s, where the 0 value means that this feature will not be used in the clustering process, and the features represented by 1 will be used.

The goal of the genetic algorithm here is to optimize the clustering using deferent combinations of features, by iterating and changing the combinations "genetically" until the best local combination is reached.

The evaluation of the features combinations requires an evaluation function (also called Fitness Function). In general the fitness function returns a rating value (between 0 – 100 for example), which allows the genetic algorithm to decide best way to go through. In our case the fitness function reflects the success of the clustering based on a combination of features; by evaluate the results of clustering process for each combination of features.

Another point must be taken into consideration is that the biological features must be used in the same evaluation. To do so, we need to calculate the correlation between the clusters -which resulting from the clustering process (bacteria clustering) - and the biological features from the other side.

Since we have a biological feature vector with the length of five features, we find that the correlation process is to be done five times in each iteration, and the best result between the correlations with each of the biological features is taken to be returned as fitness.

Finally we have to convert all of this information to a number between 0% and 100%.

Equation no 12[‡‡] is the fitness equation which evaluates the correlation between the biological features and the clusters, where K is the number of clusters, and N is the number of samples in the population. In the cluster k the number of samples which is correlated correctly to a biological feature is C, and M is the number of samples which is not correlated correctly.

$$f1 = \left[\frac{\sum_{i=0}^{k} max(C_i, M_i)}{N} \times 100\%\right] \qquad (12)$$

Equation 12 will give us a value f between 50 and 100, which mean that the best fitness will be 100 and the lowest is 50. To normalize this value between 0 and 100 in order to make the minimization process we use the equation number 13:

[‡‡] The minimum and maximum value of this equation will be 50 and 100 if the biological feature has only two possible values. But actually it will have different ranges depending on the possibilities of the biological feature.

$$f2 = (100 - f) \times \frac{Bi}{Bi - 1} \quad (13)$$

Where Bi is the Biological Feature Possibilities, it represents the number of possible values for each feature.

Ex: We have a population size with 80 samples, and the clustering divide them into three clusters, the fitness is calculated below:

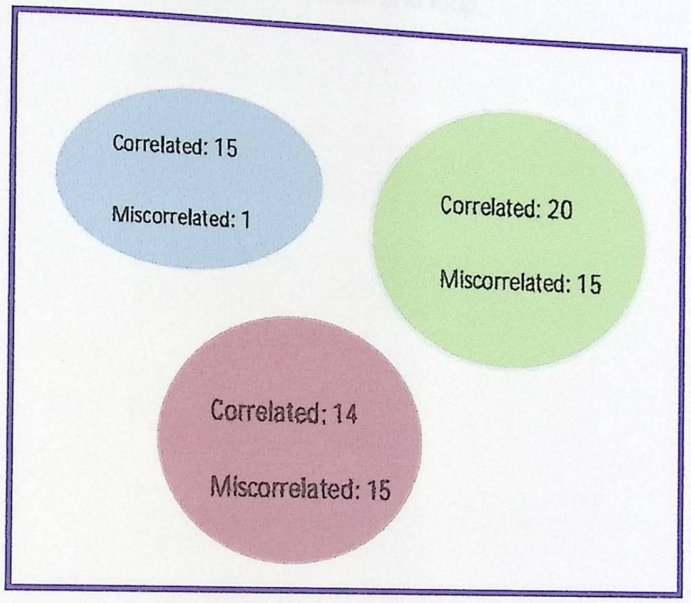

Figure 3.6: Biological feature correlation

In Figure 3.6 the population size is 80 and the biological feature possibilities are 2:

$$f1 = \left[\frac{\max(15,1) + \max(20,15) + \max(14,15)}{80} \times 100\%\right]$$

$$f1 = 62.5\,\%$$

$$f2 = (100 - 62.5) \times \frac{2}{2 - 1}$$

$$f2 = 75\%$$

We can summarize the genetic and the fitness by the following algorithm:

- Start with feature combination X from the genetic algorithm.
- Apply the clustering function (Fuzzy C-Means) on the samples using the X features
- Calculate the correlation with each biological feature
- Calculate the fitness for this combination
- Return the fitness to genetic function and loop...

3.6. The General Methodology Flowchart

This flowchart describes and summarizes the implementation of the methodologies used in this research. All of the steps are described and explained in this chapter previously.

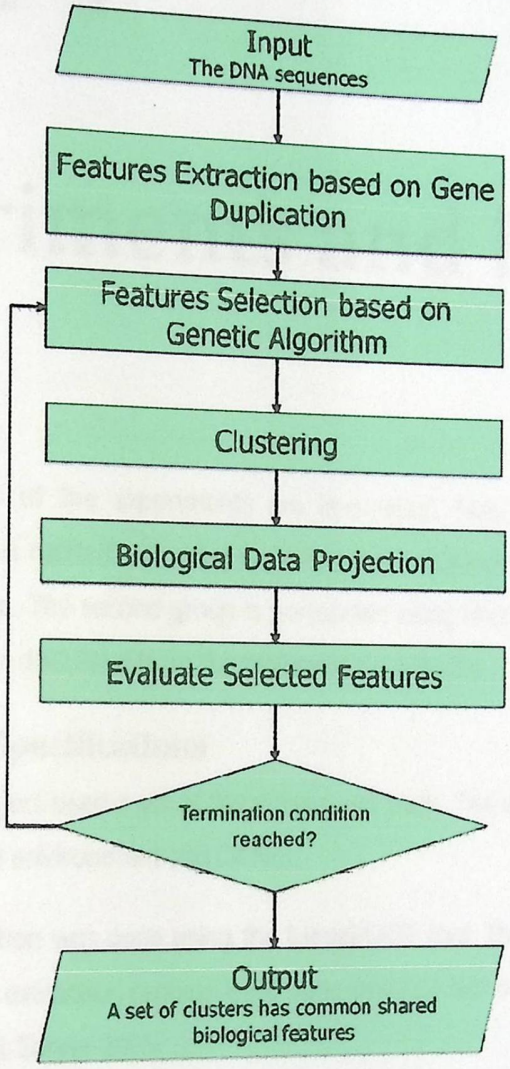

Figure 3.7: General Methodology Flowchart

3.7. Summary

In this chapter the methodologies that we used in project are described, the needed tools and algorithms and all of the needed data are listed. The way that we extract the data and why these data are selected.

Chapter 4

Experiments and Results

In this chapter the results of the experiments are illustrated. Two groups of experiments were conducted. The first group is carried out using Fuzzy C-Means clustering. The features selection was done using genetic algorithm. The second group is performed using Hierarchical Clustering. The results of these experiments are also discussed from the biological perspective.

4.1. Approaches and Specifications

The methodology of the project used a set of algorithms and tools. The application of these algorithms was done under the MATLAB environment and C#.Net.

The gene duplication extraction was done using the MegaBLAST tool. The MegaBLAST output filtering, and the duplication features extraction process were done using C#.Net code. The data were stored and managed using Microsoft SQL Server 2005.

The MATLAB Fuzzy C-Means algorithm (fcm), which is a part of Fuzzy Logic Toolbox, was used in the experiments. The main parameter for the 'fcm' is the number of the clusters, which is estimated using the subtractive clustering approach.

The Subtractive algorithm is executed using the 'subclust' function under the Fuzzy Logic Toolbox in MATLAB too; the main parameter of the 'subclust' is the radius of the clusters in addition to the feature vector.

The genetic algorithm is implemented using the 'ga' function under the Genetic Algorithm and Direct Search Toolbox in MATLAB.

In the genetic algorithms, each chromosome is an 11 binary value representing selecting/deselecting each of the duplication features. A population size of 50 was randomly selected and maintained during the experiments. The number of generations was set to 500. A single point crossover was used for reproducing new solutions. A mutation was also implemented by flipping a randomly chosen binary value. The fitness function was implemented as in Equations 12 and 13.

The Hierarchical Clustering was done using the following steps:

1- First the Euclidean distances between each pairs of features is computed using the 'pdist' function in MATLAB under the Statistics Toolbox.
2- Create a tree from the calculated distances using 'linkage' function. An example of such trees is described in section 2.7.1.
3- Finally the outputs are drawn using the 'dendrogram' functions under the Statistics Toolbox.

4.2. Data Specifications

As it is mentioned before, the input of the Fuzzy C-Means and the Subtractive clustering are a vector of features, each row of this vector represents the features of a sample. In our case the feature vector we are using is composed of 11 fields, each field or se of fields represent certain or a set of characteristics of the duplications. Each row in that vector represent on bacteria sample. The over all of the bacteria samples are 75 samples, all of the features are normalized to a number between 0 and 1, except the length of the genome.

In the duplication filter process (section 3.2.2) the selected threshold was $\tau = 95\%$.

In the feature of duplication distribution histogram the number of bins which is selected is 4 bins, so the genome is divided into four quarters.

On the other hand the process of calculating the correlation between the generated clusters and the biological data, clearly need a set of biological features. In our experiments the biological features consist of 5 features. Each feature is represented in a discreet value, where each of the biological features has a set of possible values. For example the growth rate can take one of four values (Very Rapid, Rapid, Slow, And Very Slow) each of these possibilities are represented by an integer.

Table 4.1 shows the five biological features used in these experiments with the number of possible values for each of these features:

Table 4.1: Selected Biological Features

Feature	Num Possible Values
Gram stain	2
Respiration	8
Extracellular/Intracellular	5
Generation time	4
Topology	3

4.3. Experiments

Four experiments are done each experiment include subtractive and the genetic for the Fuzzy C-Means. Each of the experiments has a fixed radius as input for the Subtractive clustering, where we used the radiuses (0.75, 0.80, 0.90, and 0.95).

Each of these radiuses has been used as input to the Subtractive Clustering, and each one has a deferent output where the number of clusters are (5, 4, 3, and 2) respectively. When using each of these outputs (number of clusters) with the Fuzzy C-Means we get a deferent combination of features and a deferent distribution of samples on the clusters.

Note that if we execute the same operations again on the same data, the results may change slightly because of the factor of randomization in the genetic algorithm and the Fuzzy C-Means.

4.4. Results and Discussion

The results are divided into two sections: Hierarchical Clustering and genetic algorithm with Fuzzy C-Means.

4.4.1. Hierarchical Clustering

After applying the Hierarchical Clustering on each of the duplication features, we got the following results where:

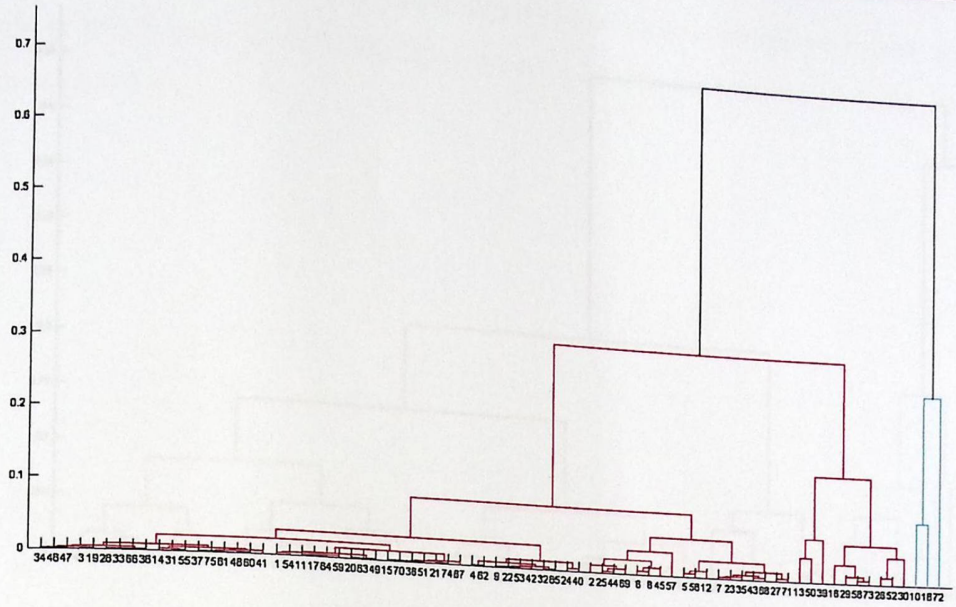

Figure 4.1: Number of Duplications Dendrogram

In Figure 4.1 the dendrogram shows the distribution of the bacteria according to the number of duplications in it. The blue colored lines represent the bacterium that has a large number of duplications, while the red color lines represent the samples that has small number of duplications.

The result that appeared are in agree with a report that was published by (Klasson et al 2009) in PNAS[§§] [21]. And the results of this research are agreed with the results in Figure 4.2, where the bacteria samples that has a high number of duplications has an moderate to small average length of these duplications.

[§§] PNAS: Proceeding of National Academy of Sciences of the United State of America

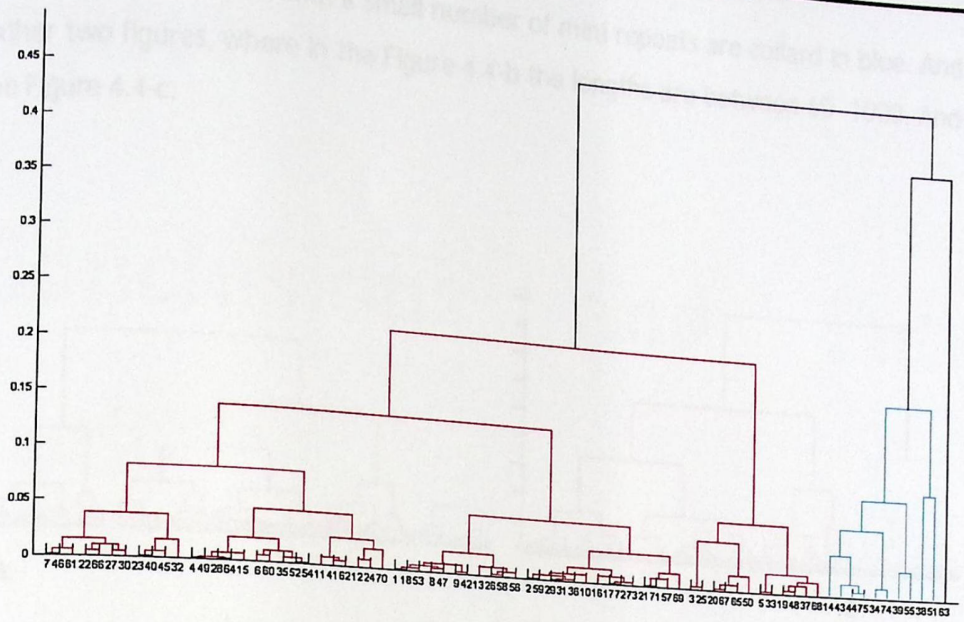

Figure 4.2: Average Length of duplications Dendrogram

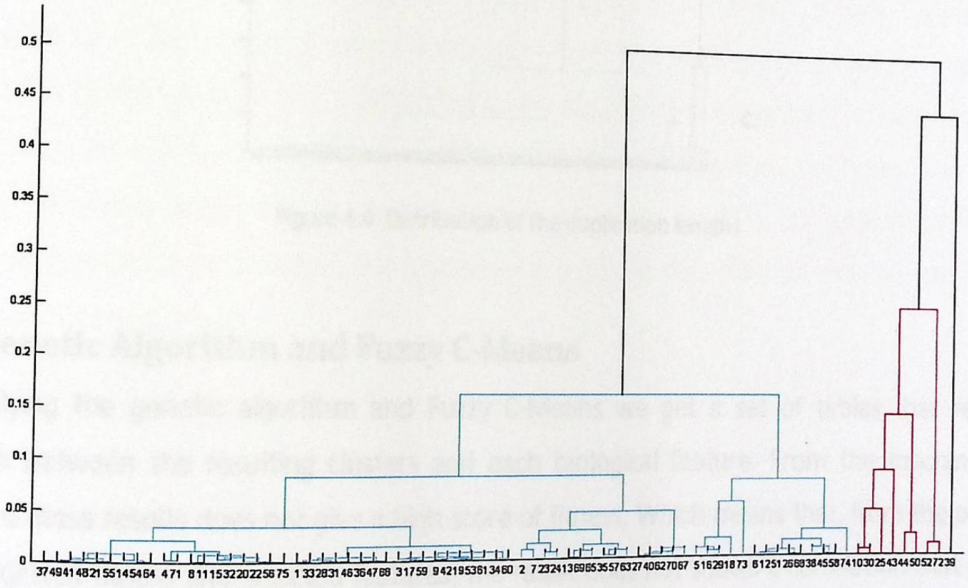

Figure 4.3: Duplication Density Dendrogram

The dendrogram in Figure 4.3 shows the distribution of the samples according to the duplication density feature, the red lines group represent the samples with high rate of the genome is duplicated 46% and more. And the blue lines group shows the samples with a rate less than 46%.

Figure 4.4 shows the distribution of samples depending on the number of each length class. Which means, in Figure 4.4-a the samples that have a large number of mini repeats duplications (28 -60) are

colored in red, and the samples with a small number of mini repeats are collard in blue. And the same with the other two figures, where in the Figure 4.4-b the lengths are between 60 -1000. And 1000 and more in the Figure 4.4-c.

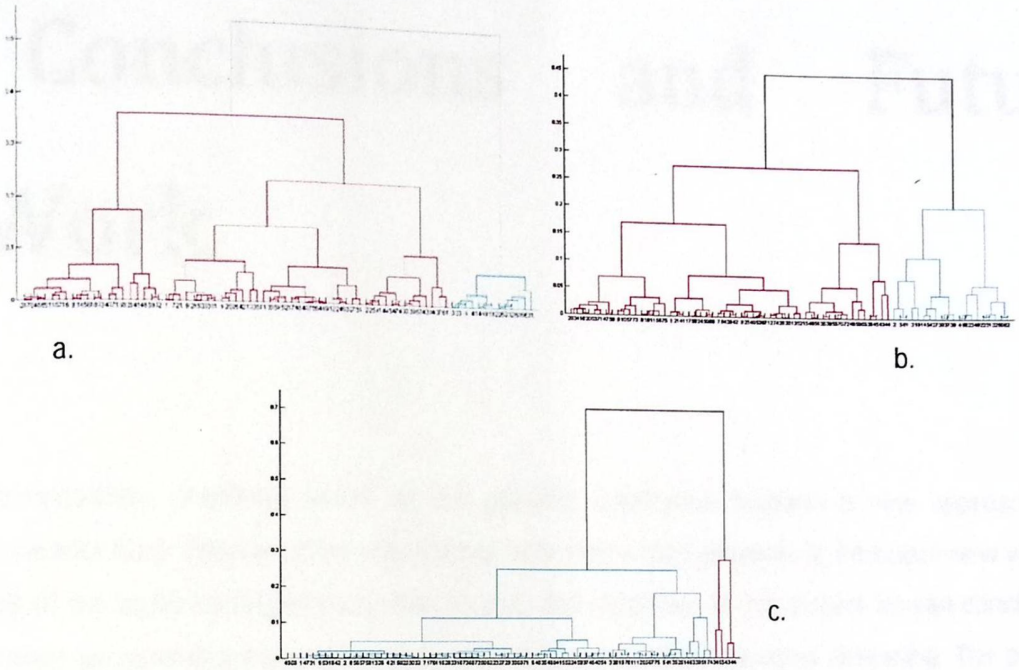

Figure 4.4: Distribution of the duplication lengths

4.4.2. Genetic Algorithm and Fuzzy C-Means

After applying the genetic algorithm and Fuzzy C-Means we get a set of tables that reflects the correlation between the resulting clusters and each biological feature. From the machine learning perspective these results does not give a high score of fitness. Which means that, from the perspective of the suggested fitness and selected features; the result does not shows a correlation that can highly determines a certain biological feature.

From the biological perspective, the results need further analyses by the biological specialist, where these results may lead to a useful data about the relationship between the genomic duplication and the biological features of the microorganisms.

The tables of results are listed in appendix A.

Chapter 5

Conclusions and Future Work

A microorganisms clustering based on the genomic duplication features is new approach in the bioinformatics field. This project is one of these new efforts that attempts to introduce new way in the analysis of the biotechnological data. According to the objectives of this project we can conclude that, the project succeed in introducing a new methodology in microorganisms clustering. The process of statistically clustering pathogenic and non-pathogenic bacterium based on its genomic duplications; is failed to get a significant evaluation. But it provides results and data that need further studying which may lead to important results.

In the future many changes could help in obtaining better results, such as, the choosing of new duplication features. So the clustering process may lead to higher correlation with the biological features. From another hand the choosing of new set of biological features that reflect other characteristics and behaviors of the microorganisms, may lead to better results. The changing of the selected fitness function may lead to better evaluation of the clustering and correlation with the biological data.

In the future work, genomic duplication can be analyzed to find out the duplications that are located in a functional gene or in a nonfunctional area. This will discard duplications that are not functional and limits the studying to functional duplications.

Appendix A

The resulting tables, from the Genetic algorithm and the Fuzzy c-Means algorithm. The results are for the radius 0.75 and 0.95.

Radius : 0.75			Number of Clusters : 5	
Output Feature Vector:		0 1 0 0 0 0 0 0 0 0 1		
Biological Feature:		Gram stain		
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples		
		-1	0	1
1	19	1	16	2
2	14	0	12	2
3	6	0	1	5
4	4	0	3	1
5	32	4	23	5
Fitness			32.00	

Radius : 0.75			Number of Clusters : 5						
Output Feature Vector:		0 0 1 1 1 1 0 0 1 0 1							
Biological Feature:		Respiration							
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples							
		-1	0	1	2	3	4	5	6	7
1	11	0	3	0	7	0	1	0	0	0
2	16	1	9	2	2	0	1	0	0	1
3	13	0	4	5	3	0	0	1	0	0
4	19	1	4	2	12	0	0	0	0	0
5	16	2	4	0	7	0	3	0	0	0
Fitness			52.50							

Radius : 0.75				Number of Clusters : 5			
Output Feature Vector:			0 1 1 1 1 0 1 0 0 1 1				
Biological Feature:			Extracellular/Intracellular				
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples					
		-1	0	1	2	3	4
1	18	1	0	7	1	8	1
2	9	2	0	4	1	2	0
3	18	6	1	8	1	2	0
4	15	3	0	6	0	4	2
5	15	1	0	3	0	10	1
Fitness			62.40				

Radius : 0.75				Number of Clusters : 5		
Output Feature Vector:			0 0 1 0 0 0 1 1 0 0 1			
Biological Feature:			Generation time			
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples				
		-1	0	1	2	3
1	23	6	5	3	5	4
2	14	2	1	1	9	1
3	20	2	2	10	1	5
4	5	0	1	3	1	0
5	13	2	4	6	1	0
Fitness			68.33			

Radius : 0.75			Number of Clusters : 5		
Output Feature Vector:		0 1 0 1 1 0 1 1 0 1 0			
Biological Feature:		Topology			
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples			
		-1	0	1	2
1	7	1	5	1	0
2	0	0	0	0	0
3	16	0	11	5	0
4	24	2	20	2	0
5	28	1	26	1	0
Fitness			23.11		

Radius : 0.95			Number of Clusters : 2	
Output Feature Vector:		0 1 0 0 0 0 0 0 0 0 1		
Biological Feature:		Gram stain		
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples		
		-1	0	1
1	11	0	4	7
2	64	5	51	8
Fitness			34.00	

Radius : 0.95			Number of Clusters : 2							
Output Feature Vector:		1 0 1 0 1 0 1 0 0 1 0								
Biological Feature:		Respiration								
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples								
		-1	0	1	2	3	4	5	6	7
1	40	3	17	1	13	0	5	0	0	1
2	35	1	7	8	18	0	0	1	0	0
Fitness			60.00							

Radius : 0.95		Number of Clusters : 2					
Output Feature Vector:		1 0 1 0 1 1 1 0 0 1 1					
Biological Feature:		Extracellular/Intracellular					
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples					
		-1	0	1	2	3	4
1	39	4	0	10	2	20	3
2	36	9	1	18	1	6	1
Fitness			59.20				

Radius : 0.95		Number of Clusters : 2				
Output Feature Vector:		0 1 0 0 0 1 0 0 0 1 1				
Biological Feature:		Generation time				
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples				
		-1	0	1	2	3
1	48	9	7	19	4	9
2	27	3	6	4	13	1
Fitness			71.67			

Radius : 0.95		Number of Clusters : 2			
Output Feature Vector:		0 0 1 1 0 1 1 1 1 0 0			
Biological Feature:		Topology			
Cluster No.	Cluster size	Biological Feature Possibilities/ Number of Samples			
		-1	0	1	2
1	40	1	33	6	0
2	35	3	29	3	0
Fitness			23.11		

Bibliography

1. Biotechnology. *WebRef.org*. [Online] http://www.webref.org/biotechnology/b/biotechnology1.htm.

2. 123 biotechnology. [Online] http://www.123biotech.com/.

3. Greg, Krukonis and Tracy, Barr. *Evolution for Dummies*. Canada : Wiley Publishing, 2008.

4. The Structure of DNA. *Genetic Home Reference*. [Online] http://ghr.nlm.nih.gov/handbook/illustrations/dnastructure.

5. Mizuta, Satoshi, Koshino, Michimasa and Shimizu, Toshio. Seeking Genomic Duplication in Prokaryotic Genomes.

6. Belkum, Alex Van, et al. *Short-Sequence DNA Repeats in Prokaryotic Genomes* . s.l. : American Society for Microbiology, 1998 .

7. Data Clustering. Wiki*pedia*. [Online] http://en.wikipedia.org/wiki/Data_clustering.

8. *Engineering*. V., Hogg R. and Ledolte, J. 1987, MacMillan.

9. MATLAB User Help.

10. *Optimizing of Fuzzy C-Means Clustering*. Alata, Mohanad, Molhim, Mohammad and Ramini, Abdullah. s.l. : WASET, 2008, PROCEEDINGS OF WORLD ACADEMY OF SCIENCE, Vol. 29, pp. 224-229 .

11. Hammouda, Khaled and Karray, Fakhreddine. A *Comparative Study of Data Clustering*. Ontario, Canada : University of Waterloo.

12. Mrazek, Jan. *Analysis of Distribution Indicates Diverse Functions of Simple Sequence Repeats in Mycoplasma Genomes*. s.l. : Oxford University Press, 2009.

13. Sequence Alignment. Wiki*pedia*. [Online] http://en.wikipedia.org/wiki/Sequence_alignment.

14. *Basic Local Alignment Search Tool*. Altschul, Stephen F., et al. 1990, J Mol Biol , pp. 403–410.

15. BLAST. Wikipedia. [Online] http://en.wikipedia.org/wiki/BLAST.

16. MaxWelling. Support Vector Machines. Toronto : University of Toronto.

17. Clustering of the Self-Organizing Map. Vesanto, Juha and Alhoniemi, Esa. 2000, IEEE TRANSACTIONS ON NEURAL NETWORKS, p. 586–600.

18. K-Means Algorithm. Wikipedia. [Online] http://en.wikipedia.org/wiki/K-means_algorithm.

19. Data Clustering. JAIN, A.K., MURTY, M.N. and FLYNN, P.J. 1999, ACM Computing Surveys, pp. 264-323.

20. Genetic Algorithm. Wikipedia. [Online] http://en.wikipedia.org/wiki/Genetic_algorithm.

21. Skinner, Michael. Genetic Algorithms Overview. Genetic Algorithms Warehouse. [Online] http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html.

22. The mosaic genome structure of the Wolbachia wRi strain infecting Drosophila simulans. L, Klasson, et al. 2009, PNAS, pp. 5725-5730.

23. FASTA Format Description. NCBI. [Online] http://www.ncbi.nlm.nih.gov/blast/fasta.shtml.