University of Nevada, Reno

Identification of Protein Coding Regions in Microbial Genomes Using Unsupervised Clustering

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science

by

Jayashree Konda

Dr. Monica Nicolescu/Thesis Advisor

December, 2009

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

JAYASHREE KONDA

entitled

Identification of Protein Coding Regions in Microbial Genomes Using Unsupervised Clustering

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Dr. Monica Nicolescu, Advisor

Dr. Mircea Nicolescu, Committee Member

Dr. Eric Marchand, Graduate School Representative

Marsha H. Read, Ph. D., Associate Dean, Graduate School

December, 2009

Abstract

At present the genomes of many organisms have been sequenced, meaning that their nucleotide structure is known but the location of genes, and most importantly, the coding regions, are unknown. Identifying coding regions is of vital importance, as they code for proteins. Distinguishing between coding and non coding regions is a difficult undertaking and many research efforts have been studied. We describe here an unsupervised clustering algorithm to find out protein coding regions in microbial genomic DNA sequences. The algorithm is based on a simple measure called vector of frequencies of nucleotides in sliding window and uses an ab-initio iterative Markov modeling procedure to partition the genomic sequences into coding, coding on the opposite strand and non-coding regions. The algorithm is very efficient and it can be used for any type of microbial genomes and also for uncharacterized microorganisms. Based on a method developed by Audic and Claverie [18], we improved the accuracy of finding coding regions and also found the nearest transition point from one class to another with an accuracy matching and exceeding the level of the best currently used gene detection methods. The method was examined on 18 complete microbial genomes from Genbank which covers four classes of major phylogenic lineages (Gram negative, Gram positive, cyanobacteria, and archaea). The results showed an improvement in performance of predicting coding regions of microbial genomes.

Acknowledgements

First of all I would like to thank my advisor Dr. Monica Nicolescu for her help, encouragement and support in the completion of this thesis.

I would like to give special thanks to Dr.Mircea Nicolescu from Computer Science & Engineering Department and Dr. Eric Marchand from the Civil & Environmental Engineering Department for serving on my committee and for their valuable time.

I would like to thank my family members for their support and encouragement throughout my Masters. Also I would like to thank Sara Nasser and Adrienne Breland for their insight into the problem and guiding me.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

With the advent of whole genome sequencing projects, databases of DNA sequences have been increasing quickly. Extracting biological information from these long genomic sequences, known as annotation, becomes a crucial biological research problem. A primary goal of a genome annotation project is to locate all protein coding regions. Once a new genomic sequence is obtained, the most likely coding regions which encode proteins are identified and the predicted proteins are then subjected to a database similarity search. Prediction is an important component of bioinformatics. Assignment of structures to gene products is a first step in understanding how organisms implement their genomic information [2]. Computational gene identification methods are essential for automatic analysis and annotation of large genomic sequences, which speeds up the process to a great extent by reducing large amount of laboratory time and resources. With many more genomes to come in the near future, the methods of highly accurate DNA sequence interpretation, particularly gene finding, become increasingly important.

A number of gene finding methods have been developed for this purpose. Most of these methods use ordering of nucleotides in DNA (or, alternatively, of amino acids in proteins) and can be described using mathematical means. This ordering in genetic structures such as genes and regulatory regions has statistical patterns, which can be determined, modeled, and compared.

There are two major approaches to gene identification: intrinsic and extrinsic [3]. The extrinsic approach is based on finding similarities between protein or DNA sequences in the database to the sequence under analysis. If there is a similarity between a certain genomic region and protein, this similarity information can be used to infer the function of that region. The intrinsic approach, also called ab-initio statistical approach, uses gene structure as a template to detect genes. It uses statistical patterns of nucleotide frequencies and nucleotide ordering observed in a given genome. These patterns are not the same in protein coding and non coding DNA sequences of a given genome; hence a properly trained intrinsic method can recognize protein coding regions. The general drawback of the extrinsic approaches is that they are inherently database-dependent and may fall short of providing sufficient support for gene annotation in novel genomes. Therefore, improvement of the *ab initio* gene finding could provide a critically important resource for annotation of novel genomes. The majority of current computational methods use extrinsic methods that require some prior knowledge of the sequence statistical properties such as codon usage or positional preferences that have to be estimated from previously identified protein coding regions.

Significant methods have taken place in gene finding and the current methods are considerably accurate, reliable and useful than those available in the past. Ab-initio computational gene finding methods are initiated by the works of Fickett [19] and Staden *et al.* [22]. Frequently used techniques employ bayesian appraoch analyzing one sequence window at a time. In doing so, protein coding regions are represented by inhomogenous periodic Markov models, either of fixed order [15], [12] or interpolated

[17]. Many current gene prediction methods are highly accurate in detecting protein coding regions, but acceleration of microbial gene sequencing has led to a high need for gene finding methods using non supervised training. Non supervised training methods are known to be more robust and they provide a greater approximation of the results of the problem without use of external data. These methods are very straightforward and effortless compared to the supervised methods.

This thesis is devoted to further improvement in predicting coding regions using iterative Markov models. The new gene prediction method utilizes a non-supervised training procedure and can be used for a newly sequenced prokaryotic genome with no prior knowledge of protein or ribosomal ribonucleic acid (rRNA). This method is an improved version of the algorithm developed by Audic and Claverie [18]. The method has been sucessfully tested on 18 microbial genomes. This method also detects the nearest transition point from one class to another with an accuracy matching or exceeding the level of the best currently used gene detection methods.

This thesis is structured as follows: Chapter 2 presents background information on computational biology, Markov models and a survey on gene prediction methods. Chapter 3 presents the algorithm we have developed to predict the genes in microbial genomes. Chapter 4 presents the results and discussion. Chapter 5 presents the conclusions and future work.

Chapter 2: Background

This chapter describes fundamentals of computational biology and explains about the Markov models, which are basics of our current algorithm. It also gives a survey on gene prediction methods.

2.1 Fundamentals of BioInformatics

2.1.1 BioInformatics

Bioinformatics is a field that analyzes the biological data by using computer technology. Research in bioinformatics includes development of methods for storage, retrieval and analysis of data. It is a rapidly developing branch of biology and is highly interdisciplinary, using techniques and concepts from informatics, statistics, mathematics, chemistry, biochemistry, physics and linguistics [1].

2.1.2 DNA

Cells are the fundamental building blocks of every living system. All instructions that are needed for cell activities are contained within the chemical DNA which is called deoxyribonucleic acid. In all organisms DNA is made up of the same physical and chemical components. DNA consists of two long interwoven strands that form the "double helix". Each strand is built from a small set of constituent molecules called nucleotides. A nucleotide is made up of one phosphate group linked to a pentose sugar, which is linked to one of 4 types of nitrogenous organic bases: adenine, cytosine,

Guanine, and Thymine symbolized as A, C, G and T, respectively. These nucleotides form a bond between the 5' and 3' positions of consecutive molecules then makes DNA strand. The representation of resulting DNA is shown in Figure 2-1.

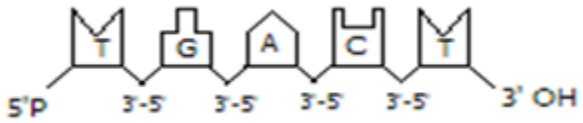


Figure 2-1 Chained nucleotides constituting a DNA strand

DNA is a double stranded molecule. Only specific bases on one strand are aligned with specific bases on other strand. The specific complementary pairs are A with T and G with C. That means thymine on one strand is always bonded to an adenine and guanine is always bonded to a cytosine. This is termed as complementary base pairing. The significance is that if the base sequence of either one of the strands of a DNA molecule is known, the sequence of the other strand can be deduced. Hydrogen bonds are formed between these complementary pairs. Two hydrogen bonds form between A and T, whereas three form between C and G. This makes C-G bonds stronger than A-T bonds.

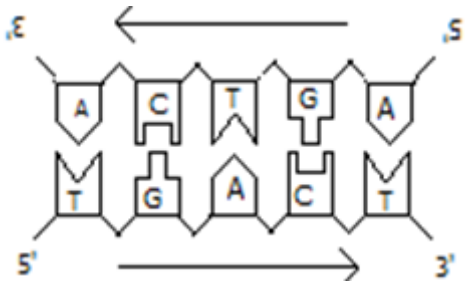


Figure 2-2 Two complementary strands of a complete DNA molecule

2.1.3 RNA

RNA is a ribonucleic acid which uses the sugar ribose instead of deoxyribose in its backbone. Uracil (U) is present in the structure of RNA instead of thymine (T). U is chemically similar to T, and in particular is also complementary to A. RNA can fold into three dimensional shapes because of two properties. First, it tends to be single-stranded in its "normal" cellular state. Second, because RNA (like DNA) has base pairing capability, it often forms intramolecular hydrogen bonds, partially hybridizing to itself with hydrogen bonds. RNA has some of the properties of both DNA and proteins. Storage capability is same as DNA due to its sequence of nucleotides and has ability to form three-dimensional structure like proteins.

2.1.4 Central Dogma of Molecular Biology

The central purpose of molecular biology is DNA transcribed into messenger RNA, which is in turn translated to protein [4]. The simplified version of central dogma is shown in Figure 2-3, in which DNA is duplicated through replication, transcribed to RNA, which is in turn translated to protein.

During the cell cycle, the DNA double strand is replicated where the replicated version serves as a template for the reproduction of a complimentary strand. DNA polymerase is an enzymatic protein, which splits the DNA double strand and synthesizes the complementary strand of DNA. Later, an enzyme called RNA polymerase copies one strand of DNA gene into a messenger RNA (mRNA) by incorporating U opposite A, A opposite T, G opposite C, and C opposite G. The RNA polymerase begins this

transcription at promoter region by binding to the DNA strand and the DNA double helix unwinds. Once the RNA polymerase reaches the coding region, it reads a single strand and builds the mRNA copy. As RNA polymerase reads each nucleotide it brings in the complementary nucleotide and bonds them together forming the mRNA strand. When it reaches the termination region called stop site, mRNA drops off from DNA strand. This mRNA is used to translate into proteins, responsible for most life functions. In the translation process, tRNA (called transfer RNA) can bind to three base pair codons on a messenger RNA which associates with the 20 common amino acids and controls the sequential binding of the amino acids (refer to appendix A for codon table). Since there are three base sequences, there are 4^3 or 64 possible codons, out of which 3 are used as stop codons and 1 as start codon to mark the end and start of translation respectively. The remaining codons are used as redundant representation of the amino acids.

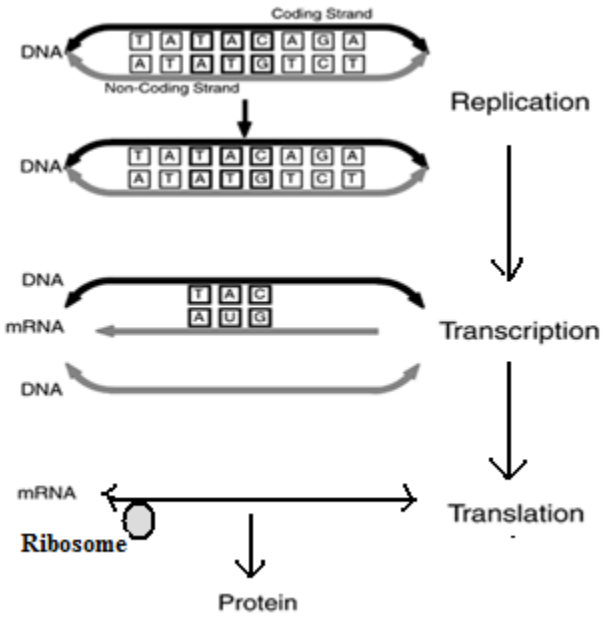


Figure 2-3 Simplified version of Central Dogma

2.2 Prokaryotes

2.2.1 Prokaryotic Genome Organization

Prokaryotes are microscopic single celled organisms. Their genome is a single, circular DNA molecule and size is in the order of a few million base pairs [$6*10^5$-$8*10^6$]. Typically 90% of the prokaryotic genome consists of coding regions. For instance, the *E. coli* genome has a size about 5 Mb and approximately 4300 coding regions, each of average length around 1000 base pairs (bp). The genes are relatively densely and uniformly distributed throughout the genome.

2.2.2 Prokaryotic Gene Structure

To solve the problem of gene prediction, the knowledge of gene structure is very important. The gene structure can be captured in terms of the following characteristics.

- **Promoter Elements**:

In genetics, the promoter is a DNA region where transcription initiation takes place. The sequence of a promoter is recognized by the sigma factor of the RNA polymerase [6]. In prokaryotes, the process of the gene expression begins with transcription where making of an mRNA copy of a gene takes place by an RNA polymerase. These prokaryotic RNA polymerases are actually assemblies of several different proteins which play an important role in the functioning of the enzymes. The promoter consists of two short sequences at positions -10 and -35 upstream from transcription start site, and is usually described as a consensus sequence. The sequence at -10 is called the Pribnow

box, which consists of six nucleotides TATAAT. This is essential to start transcription in prokaryotes. The sequence -35 elements consist of TTGACA. The presence of this sequence allows a very high transcription rate [9].

- **Open Reading Frames**:

The region of the nucleotide sequences from the start codon to the stop codon is called the open reading frame (ORF). In prokaryotes, gene finding starts from searching for an open reading frame. An ORF is a sequence of DNA that starts with start codon "ATG" (not always) and ends with any of the three termination codons (TAA, TAG, TGA). Depending on the starting point, there are six possible ways (three on forward strand and three on complementary strand) of translating any nucleotide sequence into amino acid sequence according to the genetic code. These are called reading frames.

- **Termination Sequences**:

There are some specific signals in prokaryotes to terminate the transcription process called intrinsic terminators like transcriptional start sites. Intrinsic terminators have two prominent structural features. The first is a sequence that will hybridize with it to form a base-paired stem-loop (hairpin) structure. The stem-loop is immediately followed by a consensus sequence of UUUUUUA. Formation of the hairpin loop causes transcription to pause temporarily. As the stem loop forms, its component nucleotides are pulled away from the template DNA somewhat prematurely, leaving the transcript attached to the template only by a string of relatively weak AU base pairs. In the absence of an adjacent

GC base pairing, this weak attachment may not be strong enough to keep the transcript attached to the template so transcription can continue [8].

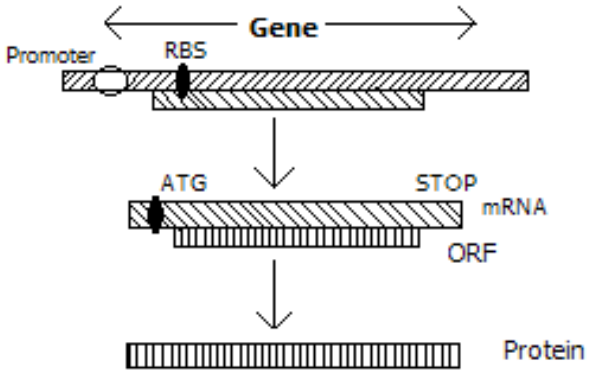


Figure 2-4 Relationship between gene, mRNA, and protein sequence for Prokaryotes [10].

2.3 Eukaryotes

2.3.1 Eukaryotic Genome Organization

Eukaryotes include a wide variety of organisms, from microscopic yeasts and fungi to elephants, plants, and humans. Their genomes consist of multiple linear pieces of DNA called chromosomes and their size is in the order of 10-670,000 million base pairs. Gene density is much lower than that for prokaryotes and it is approximately one human gene per 100,000 base pairs. Their genome contains many unusual parts and less than 5 percent of the human genome code for proteins [10].

2.3.2 Eukaryotic Gene Structure

The structure of eukaryotes is almost same as Prokaryotes, except these contains introns in between. Because of the presence of introns, Eukaryotic transcription is more

complex than Prokaryotic transcription. Transcription in the nucleus produces an RNA molecule called *pre-mRNA* that contains both the exons and introns. The introns are spliced out of the pre-mRNA by structures called *spliceosomes* to produce the *mature mRNA* that will be transported out of the nucleus for translation. A eukaryotic gene may contain numerous introns, and each intron may be many kilobases in size. One fact that is relevant to our computational gene prediction is that the presence of introns makes it much more difficult to identify the locations of genes computationally, given the genome sequence. Another important difference between prokaryotic and higher eukaryotic genes is that there can be multiple regulatory regions that can be quite far from the coding region, can be either upstream or downstream from it, and can even be in the introns.

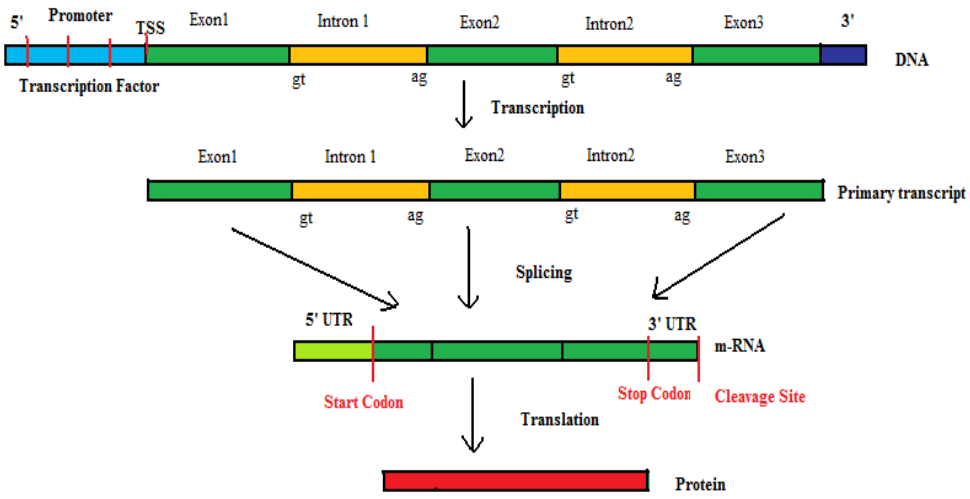


Figure 2-5 Relationship between gene, mRNA and protein in Eukaryotes

2.4 Issues in Gene Prediction

When the translation of mRNA into protein occurs, three types of posttranscriptional events influence the accuracy of gene prediction. These are:

1. The mRNA may be edited during the process of encoding into protein.
2. One tissue splices mRNA differently into another, which encodes into two similar proteins.
3. Genetic code may vary from the universal code. Because of these effects there will be issues in frame-shifts, insertions and deletions of bases, overlapping genes, genes on the complementary strand, etc.

Because of these events, basic solutions to predict coding regions in genomic sequences do not work when we need to consider all these issues.

2.5 Markov Chain Models

Markov chain models are one of the important statistical models used in fields such as statistics, physics and in queuing theory. These models have been extensively used for sequence analysis, and in the detection of protein coding regions in prokaryotes. They are at the core of the popular GenMark program [12], used to predict coding regions in prokaryotes. In this section first we describe the important and essential terms to understand Markov models and later describe the uniform and non uniform Markov models which are used as basic models for our current algorithm.

Definitions:

- *Nucleotide Frequencies*: The measure of occurrence of nucleotide pairs of a specified length in a given genome is termed as Nucleotide frequency and it is denoted by f_x. In other words, this is explained as the ratio of the number of

nucleotides of type x present in the given sequence to the total length of the sequence.

- *Codons*: A set of three nucleotides that code for amino acids. There are a total of 64 codons. For example in the sequence ACGGTCGAT the codons are ACG, GTC and GAT which code for amino acids threonine, valine, aspartate, respectively.
- *k-mer Measure*: The frequency counts of all k successive and overlapping words in a window. For example in a window of ACGTGCGTAGCTACGTCGTCAGTCGA, the counts of all dimer measures are shown in the Table 2-1.

Table 2-1 Dimer Measures

	A	**C**	**G**	**T**
A	0	2	2	0
C	1	0	5	1
G	1	2	0	5
T	2	3	1	0

- *Composition Measure*: f(b, i), where for each base b = A, C, G, T and for each codon position i = 1, 2, 3 is the frequency of b in position i.
- *Conditional Probability*: The magnitude of appearance of a nucleotide of type y after a nucleotide of type x is defined as conditional probability and is represented as P (y/x).

$$P(y/x) = N(xy) / N(x)$$

- *Markov Chain*: A Markov chain is a sequence of random variables X_i, where the probability distribution for X_i depends only on the preceding k variables X_{i-1}, …X_{i-k}, for some constant k. For DNA sequence analysis, a Markov chain models the probability of a given base b as depending only on the k bases immediately prior to b in the sequence. The most common type of Markov chain is a fixed order chain, in which the entire k-base context is used at every position. That means each position of a nucleotide depends on the preceding k variables. For example, a fixed 5^{th} order Markov chain model of DNA sequence comprises of 4^5 = 1024 probability distributions, one for each possible 5-mer context and each position of a DNA sequence depends on one of these 1024 contexts. Such fixed 5^{th} order models have proven effective at gene prediction in bacterial genomes [13]. Ideally larger values for k are always preferable. Unfortunately, because the training data available for building models is limited, we must limit k. In most collections of DNA coding, there is substantial variability in the frequency of different k-mers.
- *Transition matrix*: A Markov transition matrix is a square matrix describing the probabilities of moving from one state to another. In each row are the probabilities of moving from the state represented by that row, to the other states [11]. In DNA sequence analysis; this is defined as a matrix that consists of all the probabilities for any of the possible k words, 4^k of them, followed by one of nucleotides A, C, G, or T. Such probabilities are estimated simply by counting the

occurrence of each (k+1) - mers in the dataset. In first order Markov chain, there will be 4^1 words followed by A, C, G, T. That means it is a transition matrix of 16 possible words as shown in Table 2-2.

Table 2-2 First order transition matrix

	A	**C**	**G**	**T**
A	AA	AC	AG	AT
C	CA	CC	CG	CT
G	GA	GC	GG	GT
T	TA	TC	TG	TT

- *Uniform Markov models:* A uniform Markov chain of zero order is defined by the magnitudes of the probabilities of separate states P(x), where x represents nucleotide type A, C, G, or T [15].

$$P(x) = 1, \text{ if } x = A;$$

$$P(x) = 0, \text{ if } x \neq A \qquad (1)$$

Similarly, a uniform first order Markov chain will be specified by a vector of initial probabilities of states $P^0(a)$ and by a matrix of transitional probabilities P(b/a). This is represented as shown in the Equation 2.

$$P(xy) = P(y/x)\, P(x). \qquad (2)$$

A uniform second order Markov chain requires the specification of a vector of initial probabilities $P^0(ab)$ of 16 components, as well as a matrix of transitional probabilities P (c/ab), which is 4 x 16 in size [15].

$$P(xyz) = P(xy) * P(z/xy) \qquad (3)$$

In accordance with the uniform Markov model if we consider a window W of length L, in a DNA sequence for a given transition matrix represented as in Equation 4.

$$P(W/T) = p(s_0) * p(n_k/s_0)\ldots\ldots\ldots\ldots\ldots\ldots p(n_{L-1}/s_{L-1-k}) \qquad (4)$$

where s_i denotes the k-mer starting at the position i of the window, n_i is the nucleotide at position i, $p(s_i)$ the probability of the corresponding k-mer, and $p(n_k/s_i)$ the probability of nucleotide n, to follow the k-mer s_i.

For example if we take a sequence ACGTAG for a given transition matrix, the probability of this sequence is calculated as follows:

P (ACGTAG)/T = P(A) * P(C) * P(G) * P(T) * P(A) * P(G) – For the Zero order uniform Markov Chain

P (ACGTAG)/T = P(A) * P(C/A) * P(G/C) * P(T/G) * P(A/T) * P(G/A) – For the First order uniform Markov chain.

P (ACGTAG)/T = P (AC) * P (G/AC) * P (T/CG) * P (A/GT) * P (G/TA) – For the Second order uniform Markov chain.

- *Non – Uniform Markov models:* A non-uniform Markov model is defined as the determination of the statistical characteristics of the frequencies of mono- and di-nucleotides as a function of the position they occupy relative to the initiating codon in the coding regions of the genome [15]. All possible positions are broken down into 3 groups, termed as frames. For example, nucleotides located at

positions 1 + 3c where c = 0, 1,… form the first mononucleotide frame, nucleotides found in positions 2 + 3c, c = 0, 1…. form the second mononucleotide frame, while those found in 3 + 3c, c = 0, 1,…. form the third frame.

A non-uniform zero order Markov chain will be specified by three vectors $P^i(a)$ where i = 1, 2, 3. A vector with number i consists of the magnitudes of the probabilities of appearance of nucleotides in the i^{th} position of the codon in the coding frame. The transition matrix of non-uniform zero order Markov chain would look like as shown in Table 2-3.

Table 2-3 Transition matrix of non uniform zero order markov chain

	1st position	**2nd position**	**3rd position**
A	P^1 (A)	P^2 (A)	P^3 (A)
C	P^1 (C)	P^2 (C)	P^3 (C)
G	P^1 (G)	P^2 (G)	P^3 (G)
T	P^1 (T)	P^2 (T)	P^3 (T)

Similarly, a non-uniform first order Markov chain is defined by the three vectors of the initial probabilities $P^i_0(a)$ where i = 1, 2, 3, a = A, C, G, T, which correspond to the three P^i (a) vectors just mentioned above and by three matrices of transitional probabilities $P^i(b/a)$ where i = 1, 2, 3 [15]. This is represented as in Equation (5):

$$P^1(abc) = P^1(a) * P^1(b/a) * P^2(c/b) \qquad (5)$$

In the same way, the description of a non-uniform second order Markov chain requires specification of three vectors of initial probabilities of 16 components,

$P^i(ab)$ where i = 1, 2, 3, as well as three matrices of transitional probabilities of 4 x 16 size $P^i(c/ ab)$ where i = 1, 2, 3.

Prior research indicates that basic biological and chemical features of nucleic acids stand behind these frequencies and probabilities of nucleotides. In [15] , Mark used Markov models to find out the statistical properties of the *E. coli* genome and found that the statistical patterns of the alternation of nucleotides in the coding and non coding regions of the *E. coli* genome differ greatly. The most popular program GenMark [12] uses these models to recognize genes in both DNA strands simultaneously. GeneMark.hmm [16] was able to detect the majority of genes of all three *E. coli* classes using second order Markov models trained on the class III *E. coli* genes. The program Glimmer uses interpolated Markov models, i.e a series of Markov models with the order of increasing at each step, for finding genes in prokaryotic genomes [17]. Later GlimmerM, which is a modified version of Glimmer, was able to recognize genes in small eukaryotic genomes, such as malaria parasite *Plasmodium falciparum* [13]. All the methods described above use a training set to compute the Markov transitions matrices from which protein coding regions of other organisms are found. Stephane Audic and Jean Michel developed a self training method which uses an iterative Markov modeling procedure to find out the protein coding regions in prokaryotic genomes [18]. These models are also used to find eukaryotic promoters [14]. Thus, Markov models play a significant role in finding the important functional domains in genomes.

2.6 Gene Prediction Methods

Computational gene prediction methods are classified into two categories. One is the sequence similarity search approach and the other one is the ab-initio gene finding method. Sequence similarity search methods are also called extrinsic methods, which are based on finding similarities in gene sequences between proteins and other genomes and the input genome. The idea is that exons, present in the DNA sequence codes information for protein are more likely to be conserved whereas intergenic or intronic regions, gene segment between exons, can rapidly diverge. Once there is similarity by a certain genomic regions and DNA, or protein the similarity information can be used to infer gene structure or function of that region. The basic tools for detecting similarity between sequences are FASTA[33] and BLAST [32] which uses Smith-Watermann algorithm. The major drawback of these approaches is that nothing will be found if the database does not contain sufficient number of similar sequences. There is also a higher probability to miss small exons. Using only sequence similarity to sequences in databases may not be the best way to predict the location of genes. However, these methods can be combined with other methods to increase their total predictive power.

The second method is to use gene structure as a template to detect genes. Coding sequences have statistical properties that can be used to our advantage. Examples of these are codon counts, frequency of nucleotides, CpG islands, RNY codons where R represents purines and Y represents pyramidines, N represents purine or pyramidine. Researchers have found that there is a tendency to have the same nucleotide appearing

every 3,6,9 bp in open reading frames [19]. Programs such as GeneMark.hmm [16], Genscan [20], and Hmmgene [34] use this type of regularity as the basis for gene finding [21]. Another method is to use signal sensors. These gene finding programs could search for promoter elements, start and stop codons, splice sites or Poly-A sites. The representation of signals is offered by the positional weight matrices (PWMs), which indicate the probability that a given base appears at each position of the signal. It can also be optimized by neural network method.

In last few years, several methods have been proposed to detect genes in genomes. Some are explained here. The first method, proposed by Staden and Mc. Lachain [22], uses codon preference to find protein coding regions in long DNA sequences and also searches for ribosome binding sites which can be used to detect protein coding regions. Later Fickett [19] developed a TESTCODE property to distinguish coding from non coding, which is calibrated from eight statistical properties, among which four properties are related to the position of the nucleotide in codons and other four properties are related to the content of nucleotides in a fragment. Borodovsky and Sprizhitskii [15] analyzed the frequency of mono and di nucleotides of the *E. coli* genome and showed that DNA regions differ in functional properties like coding and non coding regions.

GENSCAN [20], developed by Chris Burge and Sam Karlin predicts complete gene structures, using probabilistic model of genomic sequence composition and gene structure. It can accurately make predictions for sequences representing either partial genes or multiple genes separated by intergenic DNA.

GRAIL, which stands for Gene Recognition and Analysis Internet Link [25]. There are two basic versions of GRAIL: GRAIL 1 makes use of a neural network approach to recognize potential coding windows in a fixed length window. A further refinement led to a second version, called GRAIL 2, in which variable length windows are used and contextual information like splice junctions, start and stop codons are considered.

Markov models played an important role to identify protein coding regions in genomes. GeneMark is a program utilizing a Markov model to identify genes in bacterial genomes. The main improvement in this method is finding genes in both strands of DNA rather than scanning one strand after another [12]. Later this method was partially improved by including information about ribosome binding site predictions [16]. GeneMark was further improved by addition of parameters to detect genes in eukaryotes. The parameters included in the program are splice sites, translaton initiation signals, exons and introns [23]. The Glimmer program [17] utilizes an interpolated Markov model which accounts predictions from first to eighth order Markov models rather than a fixed order Markov model to identify sequences likely to encode proteins. Glimmer was then used for the design of GlimmerM [13], and GlimmerHMM [24] for the prediction of genes in eukaryotes. Fgenesh and GeneParser methods use hidden Markov models to detect protein coding regions in eukaryotes [26].

Some methods use discriminant analysis, in which statistical pattern-recognition methods are used to categorize samples into two classes. There are two kinds discriminant analysis methods. One is linear discriminant analysis, where samples are

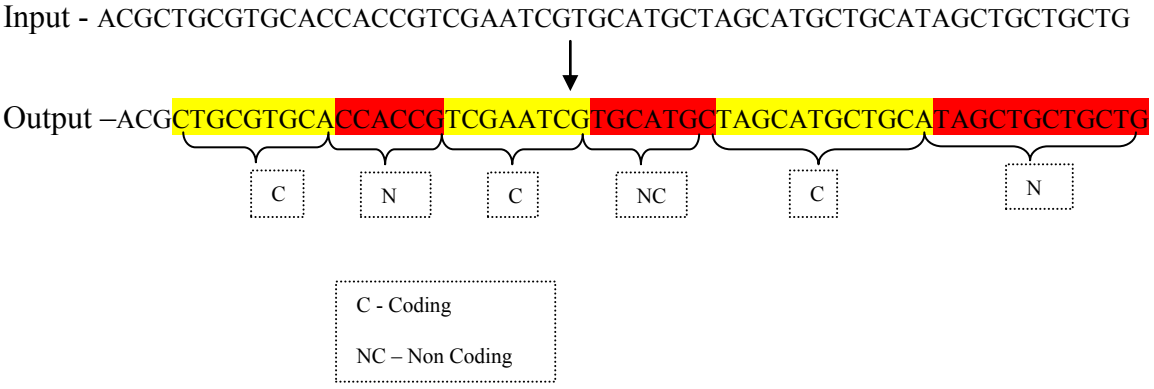
represented as points in space, finds an optimal surface that best have been represented as points that belong to two classes. Other one is quadratic discriminant analysis, finds an optimal curved surface instead [27].

The main drawback of the above methods is that they need predetermined training data to estimate all statistical properties. Collection of such large training sets represents a bottleneck in genome annotation and a practical challenge that restrains the use of ab-initio gene prediction algorithms. To avoid this difficulty, some researchers developed a gene finder able to extract statistical parameters from the original genomic sequence. Audic and Claverie [18] developed an unsupervised training method to identify protein coding regions in prokaryotes using uniform and non-uniform Markov models [18]. The GeneMarkS [29] gene prediction method utilizes a non-supervised training method and an iterative hidden Markov model to predict gene starts in microbial genome. Similar methods are developed in eukaryotes to find protein coding regions [30], [31] . However, there is still a significant need to develop more accurate methods for gene prediction.

Chapter 3: Unsupervised Clustering for Finding Coding Regions

This chapter presents the unsupervised clustering algorithm that is developed to find the coding regions in microbial genomes.

The method proposed here can be applied on any kind of prokaryotic sequences to find the coding and non-coding regions. Genomic sequences could be either complete, or be a large number of sub sequences of same organism, or be composed of many disjoined sequences resulting from a shot gun sequencing procedure. The current method is the modified version of the method developed by Audic and Claverie [18] where homogeneous Markov transition matrices are applied during the first part of the algorithm and inhomogeneous Markov transition matrices are applied during the second part of the algorithm. Instead of using non uniform Markov models, we have considered the sequences unclassified in the first part of the algorithm to improve the accuracy of the algorithm. The input of the algorithm is a prokaryotic genomic sequence of specified length taken from a GenBank and the output would be labeled coding and non coding regions of the sequence.



Below is the pseudo code for stage 1 of the algorithm, which computes the trained transition matrices from the original genome to use them in the second stage.

1. Take a prokaryotic genome of any length.
2. Divide the genome into non overlapping segments of a given length. Typically the length of the window is 100 nucleotides.
3. Randomly distribute segments among three classes: coding, coding on opposite strand, non-coding.
4. Find the Markov transition matrices for each class by finding frequency of nucleotides in each class.
5. For each iteration scan the entire genome sequence by performing steps from 6 to 11 until there are no differences among three classes. That means there should not be any difference in the number of genes from one dataset to another should be same and also the starting and ending positions of each sequence in the datasets.
6. Scan the genomic sequence by using a sliding window of W nucleotides.
7. Find the emitter class (M_1, M_2, M_3), which stand for coding on direct strand, coding on opposite strand and non coding class, for each window by using the Bayes equation

$$P(M_j/W) = \frac{P(\frac{W}{M_j})P(M_j)}{\sum_{r=1,2,\dots N} P\left(\frac{W}{M_r}\right)P(M_r)} \quad \text{---------- (6)}$$

where P(M_j) is the probability of the matrix M_j to correspond to any sequence before input. Here the genome is classified into 3 classes, therefore the value of P(M_j) is 1/3. $P(W/M_j)$ is calculated by using the following equation

$$P(W/M_j) = P(S_0) * \prod_{i=k}^{i=L-1} P(n_i/S_i - k) \quad \text{---------- (7)}$$

where S_i is the word of length k starting at position i in the sequence W, and n_i is the nucleotide occurring at position i.

8. For each sliding window follow steps 9 and 10 until you reach the end of the genomic sequence.
9. Shift the window over five positions to the right as shown in Figure 3-1 and use equations 1 & 2 to find the emitter class of this window.

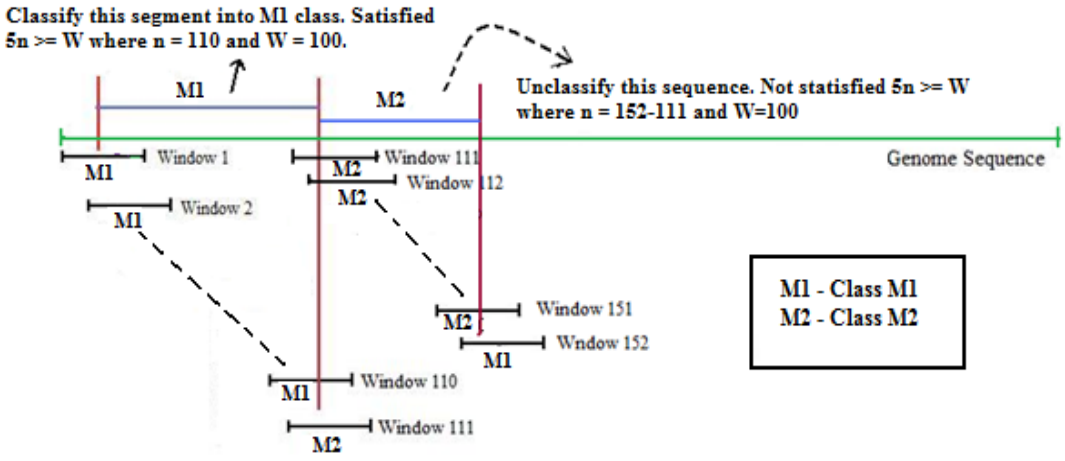


Figure 3-1 Schematic representation of Algorithm - 1

10. If ($5n \geq W$) where n is the number of successive windows categorized for the same class M_j, then classify the segment from the middle point of the first

window to the middle point of the last window into class M_j. Otherwise the segment is unclassified. Reset the window to the end of current segment.

11. Rebuild all three new Markov transition matrices after scanning the entire genomic sequence. Use these transition matrices in the next iteration to classify the sequence by using Bayes equation.

The procedure outlined below is the pseudo code for stage 2 to refine the sequences extracted from the stage 1 and also to find the nearest transition point from one class to another.

1. Use refined sequences from part 1 to calculate transition matrices.
2. For each iteration scan the entire genome sequence by following steps 3 to 11 until there are no changes among three datasets.
3. Scan the entire genomic sequence from the beginning by using sliding window of W nucleotides. Typically the length of the windows is 100 nucleotides.
4. For each sliding window of W nucleotides follow the steps from 5 to 9 until reaching the end of the genome sequence.
5. Find the emitter class (M_1, M_2, M_3) for each window by using Equation 6.
6. Shift window over five positions to the right and use equations 1 and 2 to find the emitter class of this window.
7. If there is a class transition between two windows, mark this as the current window and move 100 steps forward from the current window.

8. Slide backwards from this point using window of W nucleotides until there is no transition between two windows. Mark this window as a back window and classify the sequence from the starting position to the midpoint of current window and back window into M_j class. This is shown in Figure 3-2.

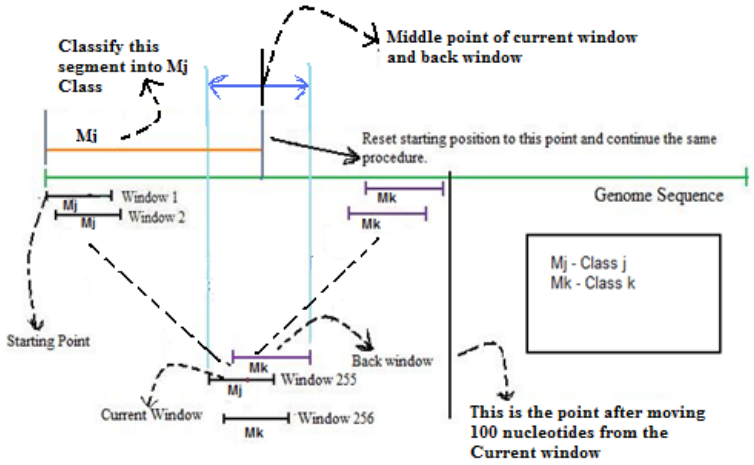


Figure 3-2 Schematic representation of Algorithm - 2

9. Reset the starting point to the midpoint of current window and back window.
10. The new transition matrices are rebuilt after analyzing the entre genome and use these transition matrices for the next iteration.

Below are details regarding the algorithm from stage 1. The stage 1 algorithm starts with the input of a prokaryotic genome of any length (Step 1). The entire genomic sequence is first randomly cut into non overlapping sequences of equal length (Step 2). Then, these pieces are randomly distributed among three classes assuming these classes are coding, coding on opposite strand and non-coding (Step 3). Markov transition

matrices are built from these sequences by using the frequencies and probabilities of 4 nucleotides adenine (A), cytosine (C), guanine (G) and thymine (T) which are basic building blocks of DNA (Step 4) (See Section 2.4). After finding the Markov transition matrices, the genomic sequence is scanned by using a sliding window of W nucleotides (Step 6). Within each window, the most likely emitter M_1, M_2, M_3 is calculated according to the Bayes equation, which is shown in equation 1 (Step 7). The window is then shifted over 5 positions to the right and the process is repeated to find the class of the sequence (Step 9). If the same class has been classified for n successive windows and the criteria $5n \geq W$ is met where n is the number of successive windows, then the genomic segment from the middle point of first window to the middle point of the last consistent one is classified into that class (Step 10). Otherwise, the current sequence segment is unclassified. After completing the analysis of the whole genomic sequence, 3 new data sets with three new classes are formed from which new transition matrices are rebuilt (Step 11). The process is iterated until there are no significant differences among 3 data sets from one iteration to another. That means there should not be any difference in the number of genes from one dataset to another and also starting and ending positions of each sequence in the datasets.

In the first part of the algorithm, there are some regions which are not classified. We refined the method by considering these unclassified portions in the second part of the algorithm that we proposed. Because of considering these regions, we are able to increase the percentage of truly predicted coding regions in microbial genomes and also improve the identification of the nearest transition point from one class to another. The

procedure from stage 2 starts with the data sets obtained from the first algorithm. Markov transition matrices are built from these data sets (Step 1). Then the genomic sequence is scanned by using a sliding window of W nucleotides (Step 3). The most likely emitter (M_1, M_2, M_3) is found out for each window (Step 5). The window is then shifted over five positions to the right and the process is repeated until there is no transition in classes from one window to another (Step 6). If there is a class transition then we mark the window as the current window and classify this into a class M_j at the transition point as shown in Figure 3-1 (Step 7). Then we move one hundred nucleotides forward and slide the window backwards from this point using sliding window of W nucleotides until there is no transition in classes from one window to another. If there is a transition from M_k to M_j, then mark down this as back window and calculate the midpoint of current window and back window (Step 8). Classify the genome segment from the starting point to the midpoint of current window and back window into M_j class and reset the starting point to current window (Steps 8 and 9). The procedure is repeated for the remaining genomic sequence. After analyzing the entire genomic sequence, three new data sets are formed. The process is iterated for entire genomic sequence with the new transition matrices until there are no changes in the data sets from iteration to other. The procedure was applied on 18 bacterial genomes and the results are improved in prediction of the coding regions with accuracy matching or exceeding the level of the best currently used gene detection methods.

Chapter 4: Results and Analysis

4.1 DataSet and Procedure

The sequence data used in the current study include the following genomes available in the GenBank database: *Haemophilus influenzae, Methanococcus jannaschii, Synechocystis PCC6803, Escherichia coli, Helicobacter pylori, Mycoplasma pneumoniae, Mycoplasma genitalium, Bacillus subtilis, Archeoglobus fulgidus, Methanobacterium thermoautotrophicum, Corynebacterium glutamicum, Acinetobacter baumannii, Candidatus Blochmannia pennsylvanicus, Candidatus Azobacteroides pseudotrichonymphae genomovar, Bartonella tribocorum, Rhodobacter sphaeroides, Yersinia pestis KIM*. All these genomes were selected randomly under prokaryotic genomes. Their length of the sequences and accession numbers, which are identificaton numbers for genomes in the GenBank database are shown in Table 4-1.

We applied the first part of the original method developed by Audic and Claverie [18], which is uniform Markov models on all the above sequences to extract the datasets for three classes coding on direct strand, coding on opposite strand and noncoding. Then we ran our algorithm as described in previous sections on the same datasets to refine the sequences obtained in the first part. We then compared the accuracy of the method develped by Audic and Claverie [18] to our method on detecting coding and non-coding regions. Comparison results are showed in Table 4-2.

Table 4-1 Data set used in the current study

	Genome	Acession number	Length in base pairs
G1	Haemophilus influenzae	NC_000907	1,830,138 bp
G2	Methanococcus jannaschii	L77117	1,664,977 bp
G3	Synechocystis PCC803	NC_000911	3,573,470 bp
G4	Escherichia coli k12	NC_000913	4,638,858 bp
G5	Helicobacter pylori	NC_011333	1,667,867 bp
G6	Mycoplasma pneumoniae	NC_000912	816,394 bp
G7	Mycoplasma genitalium	NC_000908	580,073 bp
G8	Bacillus subtilis	NC_000964	4,214,814 bp
G9	Archeoglobus fulgidus	AE000782	2,178,400 bp
G10	Methanobacterium thermoautotrophicum	NC_000916	1,751, 377 bp
G11	Corynebacterium glutamicum	NC_009342	3,314,179 bp
G12	Acinetobacter baumannii	NC_009085	3,976,747 bp
G13	Candidatus Blochmannia pennsylvanicus	NC_007292	791,654 bp
G14	Candidatus Azobacteroides pseudotrichonymphae genomovar	NC_011565	1,114,206 bp
G15	Bartonella tribocorum	NC_010161	2,619,061 bp
G16	Rhodobacter sphaeroides	NC_007493	3,188,609 bp
G17	Yersinia pestis KIM	NC_004088	4,600,755 bp
G18	Helicobacter pylori 26695	NC_000915	1,677,867 bp

4.2 Results

Table 4.2 summarizes the comparison results of the self training method developed by Audic and Claverie [18] with our current algorithm on 10 bacterial genomes which cover all phylogenetic areas. The results of genome sequences were obtained using fifth order Markov modeling of window size 100 and step 5 for 35 iterations. In Table 4-2, Column 2 represents the number of nucleotides annotated as coding on direct strand (C+) and coding on opposite strand (C-) in Genbank for that particular genome. Column 3 represents the number of coding nucleotides predicted correctly using the original method developed by Audic and Claverie [18] and column 4 represents the number of coding nucleotides predicted correctly using our current algorithm.

Table 4-2 Comparision results of two methods

	Genbank annotation of coding nucleotides in a given sequence	**Self training method developed by Audic & Claverie**	**Current Algorithm**
G1	C + 744,614 nt	636,824 nt (86%)	658,060 nt (88%)
	C- 775,845 nt	662,504 nt (85%)	687,270 nt (86%)
G2	C + 759,425 nt	661,740 nt (87%)	701,420 nt (92%)
	C- 679,908 nt	587,600 nt (86%)	606,095 nt (89%)
G3	C + 1,621,700 nt	1,279,612 nt (79%)	1,335,040 nt (82%)
	C- 1,471,880 nt	1,178,788 nt (80%)	1,283,165 nt (87%)
G4	C + 1,994,205 nt	1,582,668 nt (79%)	1,679,100 nt (79%)

	C- 2,084,634 nt	1,665,355 nt (80%)	1,720,875 nt (80%)
G5	C + 722,915 nt	634,753 nt (87%)	645,945 nt (89%)
	C- 780,576 nt	663,511 nt (85%)	693,821 nt (89%)
G6	C + 299,312 nt	284,212 nt (95%)	283,880 nt (95%)
	C- 414,027 nt	377,941 nt (91%)	379,987 nt (92%)
G7	C + 285,729 nt	266,667 nt (93%)	267,830 nt (93.5%)
	C- 226,875 nt	215,916 nt (95%)	220,836 nt (97%)
G8	C + 1,797,237 nt	1,438,578 nt (80%)	1,549,650 nt (86%)
	C- 1,877,565 nt	1,447,961 nt (77%)	1,522,960 nt (81%)
G9	C + 1,008,654 nt	838,321 nt (83%)	839,937 nt (83%)
	C- 1,010,811 nt	870,252 nt (86%)	907,725 nt (90%)
G10	C + 777,122 nt	671,485 nt (86%)	691,185 nt (89%)
	C- 810,068 nt	675,861 nt (83%)	676,795 nt (83.5%)

The results indicate that our current algorithm improves accuracy in finding protein coding regions. There are two parts in our algorithm. We have used the iterative uniform Markov modeling procedure in the first part of our algorithm, which is described in section 2.4. In this part of the algorithm, there are some regions which are not classified. We refined the method by considering these unclassified portions in the second part of the algorithm that we proposed. Because of considering these regions, we are able to increase the percentage of truly predicted coding regions in microbial genomes and also to improve the identification of the nearest transition point from one class to another.

The input of the algorithm is a genome sequence taken from a GenBank and the output is the number of nucleotides that falls into coding region on direct strand, coding region on the opposite strand and non coding region. These are the numbers are shown in the Table 4.2 as C+ for coding region on direct strand and C- for coding on opposite strand. This procedure depends on two parameters: the window size W, and the order of Markov chain. The convergence of sequences is satisfactory for Markov chain order 5. Markov models of order 5 are built from hexamer frequencies. Because of having importance to hexamer frequencies in discriminating coding and non coding regions [15], more coding nucleotides are classified successfully for Markov chain order 5. The other parameter is the window size W. The convergence behavior was observed for window sizes 50, 100, 200. But the best convergence is found for window 100. Thus a window size of 100 and a Markov order of 5 were used for all genomes. We also observed that coding regions started converging after the tenth iteration and there was no difference among 3 data sets after 35 iterations. The approximate time took to run each genome of size 3-4 million base pairs is 2.5 hours and for genomes around 1-2 million base pairs is 1.5 hours.

We also ran this algorithm for other 8 genomes from GenBank. The results are shown in Table 4-3. This table summarizes the results of the partition of genome sequences using our current algorithm. The first column for each genome represents the number of nucleotides partition in each class. Out of which number of nucleotides predicted correctly in coding regions on direct strand and coding regions on opposite

Table 4-3 Classified genome sequence segments are collected in three automatically defined data sets: C+ (Predicted coding), C- (Reverse coding) and non-coding.

Total predicted	Coding	Reverse coding	Non-Coding
	Corynebacterium glutamicum (3,314,179 nt)		
3,313,807 nt	1,470,945 nt	1,403,792 nt	439,442 nt
C+ pred. (1,420,493 nt)	**1,198,465 (81%)**	86,824	135,204
C- pred. (1,395,890 nt)	81,579	**1,168,720 (83%)**	145,591
No pred. (497,424 nt)	139,277	142,475	215,672
	Acinetobacter baumannii (3,976,747 nt)		
3,976,085 nt	1,386,647 nt	1,476,832 nt	1,113,268 nt
C+ pred. (1,256,473 nt)	**1,155,645 (83%)**	51,526	49,302
C- pred. (1,385,373 nt)	23,880	**1,270,945 (86%)**	90,548
No pred. (1,334,239 nt)	893,525	350,683	90,031
	Candidatus Blochmannia pennsylvanicus (791,654 nt)		
790,619 nt	262,794 nt	343,798 nt	185,062 nt
C+ pred. (272,610 nt)	**249,225 (94%)**	2,433	20,952
C- pred. (370,110 nt)	27,873	**310,585 (90%)**	31,652
No pred. (147,899 nt)	44,750	50,712	52,437
	Candidatus Azobacteroides pseudotrichonymphae genomovar (1,114,206 nt)		
1,108,934 nt	399,583 nt	393,057 nt	321,566 nt
C+ pred. (381,510 nt)	**366,212 (91%)**	4,682	10,616
C- pred. (399,583 nt)	6,675	**367,167 (93%)**	25,741
No pred. (327,841 nt)	173,843	15,087	138,911
	Bartonella tribocorum (2,619,061 nt)		
2,616,973 nt	1,033,771 nt	883,174 nt	702,116 nt
C+ pred. (946,385 nt)	**882,365 (85%)**	23,737	40,283
C- pred. (882,365 nt)	118,048	**736,062 (84%)**	28,255
No pred. (788,223 nt)	636,062	73,477	78,684
	Rhodobacter sphaeroides (3,188,609 nt)		
3,187,049 nt	1,499,199 nt	1,338,449 nt	350,961 nt
C+ pred. (1,440,150 nt)	**1,258,099 (84%)**	40,883	141,168
C- pred. (1,325,245 nt)	116,966	**1,110,410 (83%)**	97,869
No pred. (421,654 nt)	208,654	187,151	25,849
	Yersinia pestis KIM (4,600,755 nt)		
4,599,856 nt	1,686,315 nt	1,942,633 nt	971,807 nt
C+ pred. (1,652,890 nt)	**1,332,895 (79%)**	205,196	114,799
C- pred. (1,865,588 nt)	178,930	**1,508,163 (77%)**	178,495
No pred. (1,082,277 nt)	354,198	357,199	370,880

strand are shown bold in columns 2 and 3 respectively. The last column represents the non-coding region nucleotides. The association between each class and coding status is a priori unknown and varies from one run to the next. For example if we consider the *Corynebacterium glutamicum* genome, 81% of coding regions on the direct strand and 83% of coding regions on the opposite strand were predicted correctly when compared to annotated regions in GenBank.

The result of our current algorithm demonstrates that this method can predict coding and non-coding regions with good accuracy. In addition, there is no significant difference in the performance of our algorithm and the most popular gene prediction algorithms like Genemark [12] and Glimmer [13] which are developed using training sets. Our algorithm does not depend on any training set and will work equally well for any new bacterial genome available in the future.

Chapter 5: Conclusions and Future Work

This thesis presents an approach to identifying the coding regions in microbial genomes, which has vital importance as they code for proteins. Many methods have been proposed in the past. But majority of these methods need a training set of same species to predict coding regions. The method described in this thesis does not need any training set, uses original genome to find the statistical properties of genome and can apply equally well on any bacterial genome available in future. The results we got showed an improvement in accuracy of finding coding region. Future work will be applied to improve the accuracy in predicting the coding regions of eukaryotes by considering more parameters like splice sites, translation initiation signals, exons and introns.

Bibliography

[1] http://bioinfcpcri.org/resources/definition.htm, Nov -2009.

[2] S. Ignacimuthu, [2005], *Basic Bioinformatics*, Alpha Science International, Ltd.

[3] Borodovsky. M, Rudd. K.E, and Koonin. E.V, (1994) *Intrinsic and extrinsic approaches for detecting genes in a bacterial genome*. Nucleic Acids Research, **22**, pp. 4756-4747.

[4] Bryan. B, (2002), *Bioinformatics computing*, Prentice Hall PTR.

[5] Catherine. M, Marie-France. S, Thomas. S, and Pierre. R, (2002) *Current methods of gene prediction, their strengths and weaknesses.* Nucleic Acids Research, 30, No. 19, pp. 4103-4117.

[6] http://www.web-books.com/MoBio/Free/Ch4C1.htm, Nov- 2009.

[7] Bandyopadhyay. S, Maulik. U, Roy. D, (2008), *Gene Identification: Classical and Computational Intelligence Approaches*, IEEE Transactions on Systems, Man, and Cybernetics. Part C, Applications and reviews, 38, No.1, pp. 55-68.

[8] http://www.colorado.edu/MCDB/MCDB2150Fall/notes00/L0004.html, Nov -2009.

[9] http://en.wikipedia.org/wiki/Promoter, Nov- 2009.

[10] Jean-Michel. C, Cedric. N, (2006), *Bioinformatics for Dummies, For Dummies* , 2 edition.

[11] http://economics.about.com/od/termsbeginningwithm/g/markov_transition_matrix.htm, Nov -2009.

[12] Borodovsky. M, Mc Ininch, J, (1993), *GenMark: Parallel Gene Recognition For Both DNA Stands, Computers Chemistry*, 17, No 2, pp 123-133.

[13] Arthur L. D, Douglas. H, Simon. K, Salzberg. S, and Owen. W, (1999), *Improved microbial gene identification with Glimmer*, Nucleic Acids Research, 27, No. 23, pp 4636-4641.

[14] Audic. S, Jean-Michel. C, (1997), *Detection of eukaryotic promoters using Markov transition matrices*, Computers Chemistry, 21, No. 4, pp 233-227.

[15] Borodovsky. M, Yu. A. Sprizhitskii, E. I. Golovanav, A. A. Aleksandrov, (1986), *Statistical Patterns in the Primary Structures of functional regions of the genome in Escherichia Coli*., 20, pp 833- 840.

[16] Lukashin. AV, Borodovsky. M, (1998), *GeneMark.hmm: new solutions for gene finding,* Nucleic Acids Research, 26, No. 4, pp 1107-1115.

[17] Salzberg. S, Delcher. A, Kasif. S and White. O, (1998), *Microbial gene identification using Interpolated Markov models*, Nucleic Acids Research, 26, No. 2, pp 544-548.

[18] Audic. S, Jean-Michel. C, (1998), *Self identification of protein coding regions in microbial genomes*, Proc. Natl. Acad. Sci. USA, 95, pp. 10026–10031.

[19] J.W Fickett (1982), *Recognition of protein coding regions in DNA sequences*, Nucleic Acids Research, 10, 5303-5318.

[20] Chris. B, and Samuel. K, (1997), *Prediction of complete Gene Structures in Human Genomic DNA*, J. Mol. Biol. (1997) 268, pp. 78-94.

[21] Rogic. S, Mackworth. A and Ouellette. F, (2001), *Evaluation of Gene Finding Programs on Mammalian Sequences*, Genome Research, 817-832.

[22] Staden. R, McLachlan. A.D, (1982), *Codon preference and its use in identifying protein coding regions in long DNA sequences*, Nucleic Acids Research, 10, No. 1, pp. 141-156.

[23] Besemer. J, and Borodovsky. M, (2005), *GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses*, Nucleic Acids Research, 33, pp. 451-454.

[24] Majoros. W.H, Pertea. M, Salzberg. S, (2004), *TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene finders,* Bioinformatics, 20 , pp. 2878-2879.

[25] Uberbacher. E. C., Xu. Y., Murl. R. J., (1996), *Discovering and understanding genes in human DNA sequence using GRAIL*, Methods Enzymol, 266, pp. 259-281.

[26] Salamov. A, Solovyev. V,(2000), *Ab initio gene finding in Drosophilia genomic DNA*, Genome Research, 10, pp 516-522.

[27] Zhang. M. Q, (1997), *Identification of protein coding regions in the human genome by Quadratic discriminant Analysis methods*, Proceedings of Natural Academic Science, 94, pp 565-568.

[28] McLachlan. G. J, (1992), *Discriminant Analysis and Statistical Pattern Recognition*, Wiley, New york.

[29] Besemer. J, Lomsadz. A, and Borodovsky. M, (2001), *GeneMarkS: A self training method for prediction of gene starts in microbial genomes*, Nucleic Acids Research, No. 29, pp. 2607-2618.

[30] Vardges. H, Alexandre. L, Yury. C and Borodovsky. M, (2008), *Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training*, Genome Research, 18, pp. 1979-1990.

[31] Alexandre. L, Vardges. H, Yury. C, and Borodovsky. M, (2005), *Gene identification in novel eukaryotic genomes by self training algorithm,* Nucleic Acids Research, 33, pp. 6494-6506.

[32] Altschul, S.F, Gish. W, Miller. W, Myers. E.W & Lipman. D.J (1990) *Basic local alignment search tool*, J. Mol. Biol. 215, pp. 403-410.

[33] Pearson, W.R & Lipman, D.J (1988) *Improved tools for biological sequence comparison*, Proc. Natl. Acad. Sci. USA 85, pp. 2444-2448.

[34] www.molecularstation.com/bioinformatics/link-87.html, Oct -2009.

Appendix:

Appendix A: Genetic Code

The sequence of nucleotides adenine, cytosine, guanine, and thymine along a particular strand of DNA specifies the genetic information. A gene is a sequence of nucleotides along the DNA that codes for one protein chain. Since there are 4 letters in the DNA sequence and there are 20 different kinds of amino acids which make up all known proteins. The minimum size of the word in the DNA sequence that is necessary to code for all 20 amino acids is 3. 3 letter words gives rise to $4^3 = 64$ possible combinations. It turns out that 3 out of the 64 possible codons are reserved for stop signal to specify the end of the gene and one is reserved for start signal. Remaining is used for amino acids. Therefore a particular amino acid can have more than one, and some have up to four codons.

Table A: Codon table

First	U	C	A	G	**Last**
U	UUU - Phenylanine	UCU - Serine	UAU - Tyrosine	UGU -Cysteine	**U**
	UUC - Phenylanine	UCC - Serine	UAC - Tyrosine	UGC - Cysteine	**C**
	UUA - Leucine	UCA - Serine	**UAA -Stop** (Ochre)	**UGA - Stop** (Umber)	**A**
	UUG - Leucine	UCG - Serine	**UAG - Stop** (Amber)	UGG -Tryptophan	**G**
C	CUU - Leucine	CCU - Proline	CAU - Histidine	CGU - Arginine	**U**
	CUC - Leucine	CCC - Proline	CAC - Histidine	CGC - Arginine	**C**
	CUA - Leucine	CCA - Proline	CAA - Glutamine	CGA - Arginine	**A**
	CUG - Leucine	CCG - Proline	CAG - Glutamine	CGG - Arginine	**G**
A	AUU - Isoleusine	ACU - Threonine	AAU - Asparigine	AGU - Serine	**U**
	AUC - Isoleusine	ACC - Threonine	AAC - Asparigine	AGC - Serine	**C**
	AUA - Isoleusine	ACA - Threonine	AAA -Lysine	AGA - Arginine	**A**
	AUG - Methoionine	ACG - Threonine	AAG - Lysine	AGG - Arginine	**G**
G	GUU - Valine	GCU - Alanine	GAU - Aspartate	GGU -Glycine	**U**
	GUC - Valine	GCC - Alanine	GAC - Aspartate	GGC - Glycine	**C**
	GUA - Valine	GCA - Alanine	GAA - Glutamate	GGA - Glycine	**A**
	GUG - Valine	GCG - Alanine	GAG - Glutamate	GGG - Glycine	**G**