ABSTRACT

PONNALA, LALIT Algorithmic Approach for finding Convolutional Code generators for the Translation Initiation of Escherichia coli K-12. (Under the direction of Professor Donald L. Bitzer and Professor Winser E. Alexander).

Using error-control coding theory, we parallel the functionality of the translation of mRNA into amino acids to the decoding of noisy parity streams that have been encoded using a convolutional code. This enables us to model the ribosome as a table-based convolution decoder. In this work, we attempt to find plausible convolutional code generators for the translation initiation of *Escherichia coli* K-12. We choose the g-mask from the exposed part of the 16S rRNA. We develop an algorithmic approach to calculate the generators from the g-mask. We assign plausibility to the generators based on their ability to produce encoded sequences which exhibit a clear distinction between the translated and non-translated sequences. We also explore the construction of g-masks based on binding patterns, and evaluate the performance of the corresponding generators.

Algorithmic Approach for finding Convolutional Code generators for the Translation Initiation of Escherichia coli K-12

by

Lalit Ponnala

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science

Department of Electrical and Computer Engineering

Raleigh

2003

Approved By:

Dr. E. E. May

Dr. M. A. Vouk	Dr. W. E. Snyder
Dr. W. E. Alexander Co-Chair of Advisory Committee	Dr. D. L. Bitzer Co-Chair of Advisory Committee

To Dr. K. N. Hari Bhat, who is always a source of inspiration.

Biography

Lalit Ponnala was born in Hyderabad, India. He did his schooling at St. Patrick's High School, Secunderabad. He attended Little Flower Junior College, Hyderabad for his Intermediate studies, and went on to pursue a Bachelors Degree in Electronics and Communication Engineering at the National Institute of Technology (formerly known as Karnataka Regional Engineering College), Surathkal, India. He joined North Carolina State University for his graduate studies in the year 2001.

Acknowledgements

I would like to thank my advisors, Dr. Bitzer and Dr. Alexander without whom this thesis would not have been possible. I am indebted to Dr. Vouk for introducing me to this research area. I would like to thank Dr. Snyder and Dr. May for serving on my committee. I would also like to thank Dr. Ramasubramanian for his help. I am thankful to many friends at NC State for their continued support, my room-mates Gurucharan Patwal, Harish Vishwanathan and Srivatsa Chivukula, my good friends Gaurav Mehta, Yatin Tawde, Kiran Seth, Mangesh Dalvi and Raviraj Mahatme. I value the days at Mechatronics Research Laboratory, to Karthik Santhanagopalan and Yogesh Ramdas I owe many thanks. I thank all the members of 'Aalaap' for the countless hours of music and fun. Special thanks to all members of AID Raleigh, especially Shatakshi, Seema and Mayank Shekhar, Mrinmoyee Sanyal, Chinmoyee and Sudhir Deshpande, Sanjeev Singh, Yoganand Saripalli and Pallav Sudarshan. A hearty thanks to members of my research group, Cranos Williams, Gary Charles, Jamie Walls and Senanu Ocloo. A big thank you to Bobby Ninan, Sai Oruganti, Srikant Nalatwad, Kamala Subramaniam, Vaidyanathan Ramadurai, Belinda and Chuanhua Xing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the advent of diverse genome projects, the amount of sequence data has grown exponentially. With the increased availability of genomic data, the research emphasis has shifted from sequence compilation techniques to genomic sequence and system analysis. Research in genetics has had a highly positive impact on many facets of agriculture and medicine. DNA biotechnology is opening new avenues for research. It has had a profound influence on identifying the causes of inherited diseases, has led to the mass production of gene products, and will be the basis for gene therapy.

The topic of genomic sequence analysis has been broken down into several parts, and methods of communications and signal processing used in electrical engineering have been applied to understand it [3][19][26]. In this thesis, we attempt to model the process of translation of genetic sequences that ultimately code for protein. The specific approach that will be followed in this thesis is that of error-control coding. Redundancy exists in genomic sequences. Modelling protein synthesis as an information processing system allows DNA sequences to be analyzed as messages.

Transfer of biological information can be modelled as a communication channel with the DNA sequence as the input and the amino acid sequence which forms protein as the channel output [23]. The genetic translation system (most likely the genetic system as a whole) permits, if not requires, some degree of error. Therefore it must provide some method of error detection and error correction.

The DNA encoded message, in its double-helix form, is doubly redundant. Hence, we could consider un-replicated DNA as a half-rate systematic code, i.e., for each item (base) there is a corresponding parity item (complementary base). At the nucleic-acid level, messenger RNA (mRNA) sequences are mapped to amino-acids by grouping three nucleic acid bases together to form a codon. A codon, or three base nucleic acid vector, is mapped to a single amino acid information item. This process which occurs during translation, could be viewed as the decoding of a rate one-third code, i.e., each amino acid is encoded using three "parity" items, or it could be viewed as a code of different (non-systematic) rate depending on how much error detection capability is assumed to be present.

It is also known that leader regions of the messenger RNA (mRNA), and other prokaryotic regulatory regions contain consensus sequences which in some way signal or control translation. Examples are Shine-Dalgarno (SD) sequence in bacterial mRNA, and the TATA box and the Pribnow box in double helix DNA [1]. Specifically, the SD sequence always occurs before the start codon (usually AUG) of a translating mRNA sequence in *Escherichia coli* (*E.coli*). Our hypothesis is that, if there is a method in place that checks for the validity of the leader sequence (which includes SD), the ribosome would somehow have a way of recognizing it. Assuming there is a validating relationship among the leader sequence bases and/or codons, and assuming that the ribosome has an exposed region which is in contact with the mRNA leader for validation purposes, one may conjecture that the leader sequence may have embedded in it (or may be modeled as) an error detecting code such as a block code or a convolutional code.

The evidence is quite clear that Shine-Dalgarno type interactions within coding sequences do in fact affect the efficiency of translation. This work seeks to develop plausible coding theory-based models for translation initiation in prokaryotes, taking *E.coli* as the model organism. Convolutional coding captures the inter-relation between bases in the encoding process, and is ideal for genetic communication. The decoding of messenger RNA (mRNA) by the ribosome maps triplets of bases to an amino acid, since the ribosome moves in steps of three [44]. This is analogous to the decoding of a rate-1/3 convolutional code. We attempt to find *plausible* generators that describe this coding process.

The DNA sequences may be classified into three categories based on their behavior: *certains* or *coding sequences*, *hypothetical* coding sequences, and *non-coding* sequences. Our hypothesis is that if the generators can retain the properties of the coding sequences, and that of the non-coding sequences, they can be said to be *plausible*. This means that when the DNA is encoded using these generators and then decoded using the *genetic* g-mask, there should be a clear distinction between the translated and non-translated sequences. A generator which satisfies this property can be said to be *plausible*, and describes the translation process well.

In order to fully understand the implications of the model based on the convolutional coding theory, we need to be able to back-track from the syndrome former (which appears to be physically expressed as SD sequence in the case of bacteria), to the codes that actually can produce that syndrome former. In general, this is a many to one relationship. We would expect that to be the case since the DNA information engine may employ errors to control translation efficiency and to "lock-in" into the translation frame [44].

Chapter 2 gives an overview of the state-of-the-art with regard to information-based models for genomic sequence analysis. Chapter 3 discusses table-driven convolutional coding in some detail. Chapter 4 develops the method for finding the generators of the convolutionally encoded genetic sequence, based on mRNA leader binding properties. Chapter 5 discusses the results obtained. Chapter 6 summarizes the research results and concludes the thesis.

Chapter 2

Overview

The central premise of genetics is that genetic information is perpetuated in the form of nucleic acid sequences but functions once expressed as proteins [1]. Various investigators have developed models that attempt to capture different information related aspects of the genetic system. Most of these models are based on the information contained in DNA sequences.

In this chapter, the basic understanding of nucleic acids and their relationships to protein production is outlined. This will be followed by an overview of information theory based approaches to genetic sequence and system analysis is given. The state of research with regard to error-correction coding theory methods for evaluating the genetic translation initiation system is explored.

2.1 A brief overview of Genetics

This section provides a cursory overview of the basic mechanics of protein production and supplies the definitions of various terms used in this work. The material presented here is available in standard texts on molecular genetics [1][2].

2.1.1 The Chemical Structure of Genetic Material

Genes and chromosomes are the fundamental units in the chromosomal theory of inheritance, which explains the transmission of genetic information controlling phenotypic traits. Deoxyribonucleic acid (DNA) found in the chromosomes of all eukaryotes and most prokaryotes occurs in the form of macromolecules. There are four different kinds of nucleotides that make up the basic structural units of the DNA molecules. Variation in the chains of nucleotides generates millions of DNA molecules.

Each nucleotide consists of a nitrogenous base, a phosphate and a sugar group. The base combines with the deoxyribose sugar to form a nucleoside, which, in turn, is attached to the phosphate group to produce deoxyribonucleotide. The different nucleotides in the long chain polymer are linked through phospho-diester linkages.

The nucleotides in a DNA molecule differ in the nature of the nitrogenous base. The bases are of two types: purines and pyrimidines. Adenine (A) and Guanine (G) are the purines and Cytosine (C) and Thymine (T) are the pyrimidines. There is conclusive evidence that genes are segments of DNA, and the DNA molecule formed by polymerization of the four types of nucleotides can generate all the genetic variability that exists within and among different species. Watson and Crick proposed the complementary base-pairing of DNA bases in the duplex structure, and this proved to be in excellent agreement with the experimental observation of Chargaff on the composition of DNA molecules.

2.1.2 Replication

Replication is an essential function of the genetic material. Though it is not error-free, it requires high precision, and an extremely accurate system of DNA replication has evolved in all organisms. In the case of the human genome, there are about 3 billion base pairs, and even an uncorrected error-rate of 10^{-6} would create 3000 errors during each replication cycle! The phenomenon by which DNA replicates is called *semiconservative replication.* Watson and Crick [5] suggested that because of the arrangement of nitrogenous bases, each DNA strand could serve as a template for synthesis of its complement. Due to the stability of the hydrogen bonds between complementary bases, A would attract T, and C would attract G. This would result in the production of two identical DNA double strands, each containing one "old" and one "new" strand. The fidelity of synthesis is increased by the action of DNA polymerases, which detect and excise mismatched nucletides. This process is called *exonuclease proofreading.*

2.1.3 Transcription

The first step in gene expression involves the transfer of information present in the template strand of the DNA into RNA by the process of transcription. This is brought about by the action of RNA polymerase, which resembles the DNA polymerase except in that the nucleotides contain the ribose rather than the deoxyribose forms of the sugar. Initiation of transcription is dependent on an upstream (5') DNA region, called the *promoter*, that represents the initial binding site for RNA polymerase. Promoters contain specific sequences, such as the TATA box, that are essential to polymerase binding. The genes of eukaryotic organisms contain internal nucleotide sequences that are not expressed in the amino acid sequences of the proteins they encode. These regions are called *introns* and those that are retained and expressed are called *exons.* The synthesized RNA contains the coded information, and is hence called the messenger RNA (mRNA). In eukaryotes, the intron regions are removed before the mature mRNA is translated. In contrast, the genes of prokaryotic organisms do not contain any intron regions.

The mRNA code contains triplets of ribonucleotides, called codons, each specifying one amino acid. A given amino acid can be specified by more than one codon (for 18 of the 20 amino acids), which makes the code *degenerate.*

2.1.4 Translation

The final product of gene expression, in almost all instances, is a polypeptide chain consisting of a linear series of amino acids whose sequence has been prescribed by the genetic code. Translation is the polymerization of amino acids into polypeptide chains. It is a complex energy-requiring process that also depends on transfer RNA (tRNA) molecules that adapt specific triplet codons in mRNA to their correct amino acids. The tRNA molecules are first charged under the direction of *aminoacyl tRNA synthetases*, and then chemically linked to their respective amino acids.

Since prokaryotes do not have nuclei, translation takes place as soon as mRNA is created. On the other hand, organisms with nuclei, eukaryotes, ship mRNA outside the nucleus before producing proteins. The process of translation is best described by dividing it into three phases: initiation, elongation and termination.

2.1.5 Initiation

This process involves the small subunit of the ribosome (called the 16S rRNA in prokaryotes), a specific charged initiator tRNA, and a number of proteinaceous initiation factors (IFs), which are required to enhance binding affinity. In bacteria, the binding of the ribosome to the mRNA involves a sequence of up to six ribonucleotides (AGGAGG) which *precedes* the initial mRNA start codon. This sequence, called the Shine-Dalgarno sequence, base-pairs with the exposed part of the 16S rRNA, facilitating initiation. In prokaryotes, the initiation codon of mRNA - AUG - codes for the modified amino acid *formylmethionine (fMet)*. During initiation, the reading frame is set, so that all subsequent ribonucleotide triplets are translated accurately. Once initiation is completed, the IFs are released.

2.1.6 Elongation

This is the second phase of translation. Once the ribosome is assembled with the mRNA, binding sites are formed for two charged tRNA molecules - the P, or peptidyl and the A, or aminoacyl site. The charged initiator tRNA binds to the P site, provided the AUG triplet of the mRNA is in the correct position of the ribosomal subunit. The second mRNA triplet indicates which charged tRNA molecule will position itself at the A site. Once it is present, a peptide bond will be formed linking the two amino acids together. The increase of the polypeptide chain by one amino acid is called elongation. The uncharged tRNA from the P site is then released, and the entire mRNA-tRNA complex shifts downstream by a distance of three nucleotides. The third mRNA triplet is now ready to accept a charged tRNA. The above sequence of events is repeated over and over, and each time, an additional amino acid is added to the growing polypeptide chain.

2.1.7 Termination

Termination, the third phase of translation, occurs when one of three codons: UAG, UAA or UGA, enter the A site. These codons, called stop codons or nonsense codons, stall protein production since they neither specify an amino acid, or call for a tRNA in the A site. The polypeptide chain is detached from the tRNA at the P site, and the ribosome breaks back down into the 30S and 50S subunits.

2.2 Function of coding sequences

Rodolphe and Mathe [6] state that the sequence that codes for a protein cannot be determined from its function because of the vast amount of degeneracy present in the genetic code. Coding sequences must satisfy several constraints such as amino acid requirements, genomic mutational biases, constraints imposed by DNA structure, translational processes, transcriptional processes, mRNA stability and structure and the necessary presence of motifs. So, besides neutral polymorphism, genetic code degeneracy is certainly widely *used* to satisfy such constraints. Rodolphe and Mathe used Markov Chain models to show that codon choices are dependent, and this dependence induces preferences different from those induced by all of the transcription and translation process. They also infer the existence of constraints other than those related to the transcription and translation process acting on coding sequences everywhere.

Orlov et al [7], attempted to model genetic text without incorporating any prior information, thus providing a way of assessing the information measure of the genetic code. They used variable memory Markov models for the generation of symbols based on a stationary source. Using the method of tree sources, they studied dependencies of the symbols upon the preceding context in DNA sequences. They found that, surprisingly, the entropy reduction for DNA sentences due to correlations is only about 1-5%.

Jong [8] states that there is a pressing need to develop formal methods and computer tools for the modelling and simulation of genetic regulatory networks, which involve interactions between DNA, RNA, proteins and small molecules. The argument being that since such systems involve several components connected through interlocking positive and negative feedback loops, an intuitive understanding of their dynamics is hard to obtain.

Schultz and Yarus [15] contradict the view that the genetic code is “frozen” and immutable. It was thought to be so because the transformation of codons into lethal amino acids could terminate an organism. The authors argue that codon reassignment through an intermediary tRNA is more probable than schemes requiring the disappearance of a codon. This means that all genomic instances of a codon must be replaced by synonyms or undergo mutation. The authors state that “variation from the universal genetic code may usually depend on a selected extension of tRNA coding properties”.

2.3 DNA Computation

DNA computation involves manipulating a set of DNA strands using certain *bio-operations* such as hybridization and denaturation. It is imperative that, for such operations to be performed efficiently, we have a set of DNA codewords with desirable properties. Kari and Konstantinidis [10] deal with the problem of finding sets of codewords that do not form undesirable bonds with each other. A framework for DNA computation has been developed based on the theory of formal languages. The authors have developed a language of codewords that can detect an error occurring anywhere in a sequence of eight consecutive nucleotides.

According to Marathe [13], designing equi-length words over the DNA alphabet A, C, G, T that satisfy certain combinatorial constraints can enable efficient DNA computation. Building on classical results from coding theory, the author calculates upper and lower bounds on the size of DNA codewords which satisfy the additional constraint of similar free energy and enthalpy. Large codewords are desirable because they can store more information. The author emphasizes the need for experimental work to validate the use of combinatorial constraints in the design of DNA codes.

In this thesis, we lay emphasis on the process of translation initiation. The DNA sequence is encoded using the "generators" of our model, and the mRNA sequence is formed. Assuming that the code at the translation-initiation stage is a convolutional code, a suitable model that represents this code will be found. Since the output of the translation stage is the coding mRNA sequence, a code that governs this process must be able to provide a clear distinction between the translated and non-translated sequences. Our specific task will be to find plausible convolutional code generators that describe efficiently, the translation initiation process.

2.4 Information-based Models

In [9], MacDonaill recognizes that the process of replication is one of information transmission, and informatics plays an important role in shaping the nucleotide alphabet. He approaches the problem of alphabet composition from the perspective of error control coding theory. Based on a digital representation of hydrogen donor/acceptor (D/A) patterns, he states that there are a total of 16 possible informationally distinct nucleotides, using parity-check codes. The nucleotides of the natural alphabet A, U, C, G belong to a set of even-parity codes, which lends support to the fact that the natural alphabet is structured as a parity code. He says "Nature's choice of just four nucleotides may lie in a requirement for *hardware integrity*, that is, a measure of chemical robustness and pattern stability". Other patterns are eliminated based on considerations of tautomeric instability of nucleotide structure and vulnerability to hydrolysis.

Houen [16] observes that RNA is translated into proteins by ribonucleoproteins, while proteins influence other information processing steps such as DNA replication, transcription and protein folding. From this, it can be inferred that "the evolution of DNA as an information storage medium was a secondary event, unrelated to the evolution of the genetic code."

Yockey [18] compares the genetic logic system to a Turing machine, with the DNA as the input tape. The genetic message is recorded, and the system moves through the "internal states" tRNA, mRNA and synthetases. The proteins specified by the genetic message are analogous to the output tape.

The Central Dogma, according to Yockey, states that the information may be transferred from DNA to DNA, DNA to mRNA, and mRNA to protein. There are three transfers of information that the Central Dogma prohibits: from protein to protein, protein to DNA and protein to mRNA. The Central Dogma indicates a unidirectional flow of information from DNA to protein. In fact, the genetic code does have a Central Dogma since the source (DNA) and receiver (protein) have different entropies. There is information loss in the conversion from mRNA, which is made up of an alphabet of 64 letters, to the protein which has 20 alphabets.

Yockey states that Shannon's conditional entropy is the proper measure of genetic noise, and that information content or complexity of proteins can be measured by the mutual entropy of the protein sequence. He explains that the reduction of redundancy present in DNA during its conversion to protein, and the redundancy present in the protein sequence could allow more than one genetic message to be encoded in one reading frame of the single-stranded DNA. Thus, overlapping genes illustrate the use of all the information present in the DNA sequence to record genetic messages. The genetic code is instantaneously decodable and optimal, in the coding theory sense. It makes the most optimal use of its alphabet, namely, the 4 nucleotides.

Yockey also explains the phenomenon of aging using error-control coding. He cites the case of the organism coelacanth, which has preserved its morphology for 4×10^8 years, to illustrate that the protein synthesis process is highly accurate. Chemical mutagens, UV rays and X-rays cause errors in DNA replication, but many of these errors are removed by “repair” enzymes. The overall error rate in DNA ranges from 10^{-7} to 10^{-12}. It is shown, based on approximate calculations of the error frequency that the protein generating system can tolerate, that simplest organisms must have had high error-protection.

Yockey states that genetic noise occurs due to two factors: mischarged tRNA moelcules, and single base interchanges. Using the communication analogy, he postulates that as time goes by, some sources forget the original “sense” codewords and use codewords less protected from noise. This is similar to the effect of mutations that create less protected, yet functionally equivalent amino acids.

2.5 Error Correcting Codes in Genomics

Battail [12] says that error-correction coding needs to be incorporated in Dawkin's model of evolution, and suggests nested encoding (where information is encoded several times over) as a plausible coding scheme. The author observes that the extremely small error rate of genome replication seems impossible in the absence of some form of error-correction, since molecules are vulnerable to thermal noise and degradation on account of cosmic rays and natural radioactivity. According to the author, organisms that have a short genome replicate very fast, and this serves the purpose of keeping their genetic information intact in the absence of error-correction coding. In contrast, eukaryotes have longer genomes and protect themselves against aging by means of error-correction coding, in spite of low replication speed.

Leibovitch [17] tries to find if there is an error-correcting block code structure in the DNA. He specifically explores the possibility of an (n,n-1) code, i.e., a code in which there is ONE parity bit which is expressed as a linear combination of the information bits. The DNA bases are mapped to a finite field of 4 elements, and a Gaussian reduction method to solve for the coefficients x_i that make up the equation

$$a_1x_1 + a_2x_2 + ... + a_nx_n = 0 \qquad (mod\ \ 4) \tag{2.1}$$

where a_i are the DNA bases. By dividing the DNA sequence of the *cytochrome-c* gene into consecutive base-strings of length 3 to 8, solutions to the above equation were computed. Non-trivial solutions were found in a very small number of cases, which led to the conclusion that the DNA sequence is devoid of a linear block-coding structure. The author urges the research community to analyze DNA as a language, and suggests the possibility of more complex error-correcting codes being present in DNA.

Rosen [11] asks the question: "Is DNA a code in the information theoretic sense?", and investigates redundancy in the coding structure of DNA using parity check codes. The algorithm used modifies the vector search method using a subspace partitioning algorithm, which is a general technique for detecting redundancy in a linear code. Based on the results obtained, Rosen concludes that there is no underlying code in the DNA, but attributes any fallacies to the difference in mutation rates between the coding and non-coding regions. In other words, the same code may not be present throughout the genome, and may vary depending on the function of the particular region. The author also states that, from a coding perspective, intron regions (in eukaryotes) may contain important error-control information.

Arques [14] identified three subsets of nucleotide triplets that can be used to recover the reading frame of a coding sequence. These codes are called *circular codes* and have favorable properties such as complementarity, circular permutation, rarity and concatenation. By excluding the three codons having similar nucleotide composition, the author divided the remaining 60 codons into 20 classes of three each so that, in each class, the three codons are deduced from each other by circular permutations. Almost all autocorrelation functions applied in the protein genes of prokaryotes and eukaryotes have a modulo 3 periodicity.

Viewing protein synthesis as an information-processing system allows nucleotide sequences to be analyzed as messages [23]. May et al.'s model [24] defines a genetic channel as the DNA replication, transcription and translation process during which errors may be introduced, detected and possibly corrected. Figure 2.1 depicts May et al.'s initial coding theory view of translation initiation.

The transfer of biological information can be modelled as a communication channel, with the DNA sequence as input, and the polypeptide amino acid sequence as the output. As in error-protected information channels, redundancy occurs within genomic sequences. The DNA encoded message, in its double-helix form, is doubly redundant. Hence, we could consider un-replicated DNA as a half-rate systematic code, i.e., for each item (base) there is a corresponding parity item (complementary base). At the nucleic-acid level, messenger RNA (mRNA) sequences are mapped to amino-acids by grouping three nucleic acid bases together to form a codon. A codon,

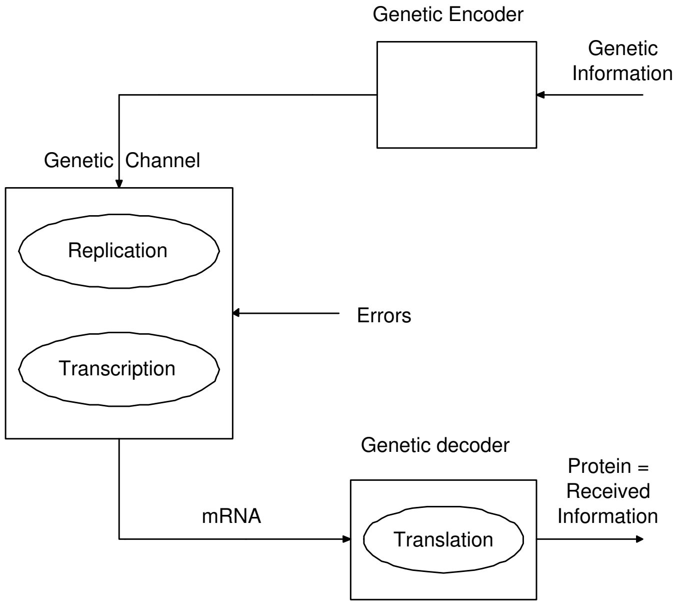

Figure 2.1: Initial Coding theory view of Genetics.

or three base nucleic acid vector, is mapped to a single amino acid information item. This process which occurs during translation, could be viewed as the decoding of a rate one-third code, i.e., each amino acid is encoded using three “parity” items, or it could be viewed as a code of different (non-systematic) rate depending on how much error detection capability is assumed to be present.

May et al.’s initial communication view of the genetic system [24] has been modified as follows: (1) The replication process represents the error-introducing channel; (2) Assuming a nested genetic encoder, the genetic decoding process occurs over three levels: transcription, translation initiation, and translation elongation plus termination. Figure 2.2 depicts our final coding theory view of translation initiation.

May et al.’s hypothesis is that if the redundancy, or extra information, contained in the genetic sequence is used to protect the organism against errors, then it would be feasible to use principles of error control coding theory to interpret the genetic translation initiation mechanism [45]. If the hypothesis is realizable then there may exist a valid set of table-based, convolutional, error-control codes that:

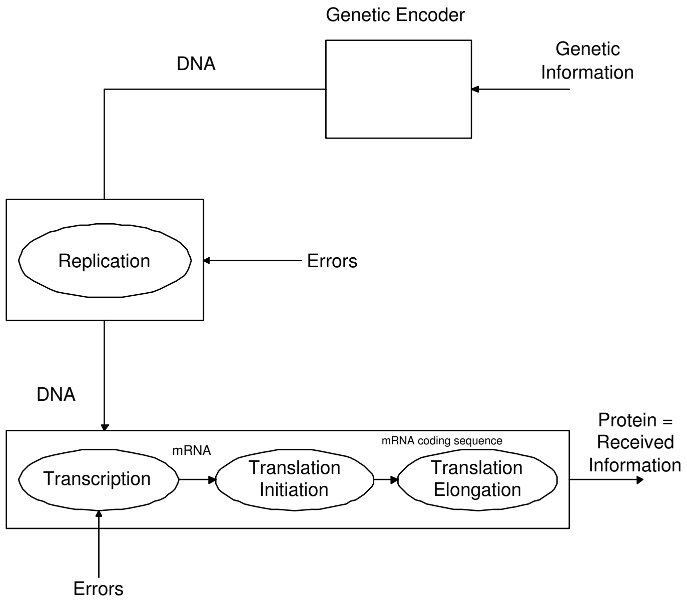

Figure 2.2: Modified coding theory view.

- Have decoding masks (gmasks) that are similar to the 3' end of the 16S rRNA
- Have decoding masks which identify key sites on the mRNA leader (5' untranslated) region that are involved in translation initiation
- Have decoding masks that can be used to detect valid and invalid initiation sites.

Chapter 3

Convolutional Codes

This chapter explains the fundamentals of convolutional codes, and their application to genomics. The material on convolutional coding can be found in coding theory texts [32][33], while the material on table-driven techniques may be found in [34][35][36][37].

3.1 Error control coding

There are two basic types of codes: block and convolutional. Block codes independently encode fixed length blocks of information. Convolutional codes differ from block codes in that each encoding operation depends on current as well as a number of past information bits. This number is known as the *constraint length* of the code. It is desirable to have large constraint lengths (or block lengths) for efficient encoding. However, the decoder complexity increases in proportion to the constraint (or block) length.

Codes can also be categorized as error-detecting and error-correcting codes. Both block codes and convolutional codes can be used as error-detecting or error-correcting

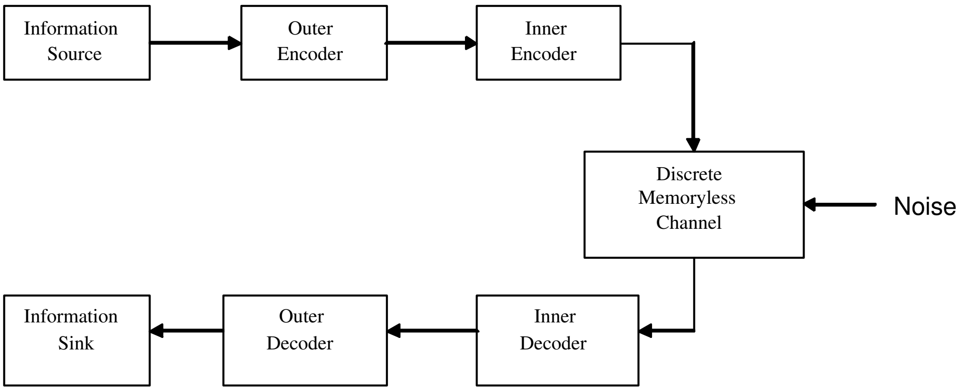

Figure 3.1: A concatenated coding system.

codes. Further, both block and convolutional codes can be designed to have simultaneous error-detecting and correcting capabilities. Yet another coding technique using multiple encoding and decoding stages was proposed by Forney [38], and is known as *concatenated coding.* The general model of a concatenated coding system is shown in Figure 3.1.

The genetic coding system may be modelled as a concatenated code. Assuming a nested genetic encoder, the genetic decoding process occurs over three levels: transcription, translation initiation, and translation elongation plus termination.

3.2 Convolutional Coding

We will now illustrate with an example, how a convolutional code works. We will assume that binary data is being encoded. Consider the circuit shown in Figure 3.2. The square boxes are the delay elements (or memory elements), and the circles are modulo-2 adders. The information sequence $\mathbf{u}$ is shifted in from the left one bit at a time, and two encoded sequences $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ are the outputs of the modulo-2 adders.

Thus, at time unit i, the current information bit is u_i, the contents of the two delay elements are u_{i-1} and u_{i-2}, respectively, and the two outputs are ${v_i}^{(1)}$ and ${v_i}^{(2)}$.

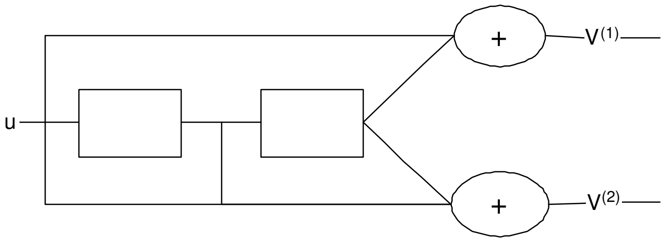

Figure 3.2: A convolutional encoder.

The first output ${v_i}^{(1)}$ is the modulo-2 sum of u_i and u_{i-2}, while the second output ${v_i}^{(2)}$ is the modulo-2 sum of u_i, u_{i-1} and u_{i-2}. The sequence of encoded bits is multiplexed and transmitted over the channel.

In general, at time i, a block of k information bits is shifted in and a block of n encoded bits is generated. The encoded block at time i depends not only on the information block at time i but also on m previous information blocks. Thus, a convolutional encoder requires memory. The encoder above has encoding rate R $= k/n$.

A message to be sent over the communication channel can be viewed as a sequence of discrete source symbols and is called an *information sequence*. It is input to a channel encoder, and the output of the encoder is called the *encoded sequence* (or the *codeword*). The set of encoded sequences corresponding to all possible information sequences drawn from a specific alphabet is called a *channel code*. Channel coding should, at the very least, allow the possibility of recovering (or *decoding*) the original information unambiguously from the *received sequence* in the absence of channel noise. That is, a one-to-one correspondence must exist between the sets of information and encoded sequences.

A few formal definitions related to convolutional codes will now be given [33].

Definition 1. *An (n,k) convolutional code C over a finite field F is a k-dimensional subspace of n-dimensional vector space F((D))^n.*

Definition 2. *An encoder is a $k \times n$ convolutional encoder over F if the mapping $F^k((D)) \to F^n((D))$ realized by the encoder can be represented by $\mathbf{V}(D) = \mathbf{U}(D)\mathbf{G}(D)$, where $\mathbf{G}(D)$ is a $k \times n$ matrix of rank k with entries in the subset F[D] of F((D)).*

Definition 3. *Let C be an (n,k) convolutional code. Then its dual code $C^\perp$ is an (n-k)-dimensional subspace of $F((D))^n$ consisting of all sequences $\mathbf{V}^\perp(D)$ orthogonal to all encoded sequences $\mathbf{V}(D) \in C$.*

The dual code $C^\perp$ is itself an *(n,n-k)* convolutional code. It is generated by any rate-*(n-k)/n* encoder $\mathbf{H}(D)$ such that $\mathbf{G}(D)\mathbf{H}^T(D)=\mathbf{0}$. The matrix $\mathbf{H}(D)$ is called the *parity-check matrix* of the code *C*. An algorithm to determine $\mathbf{H}(D)$, given $\mathbf{G}(D)$, is discussed later in this chapter.

Definition 4. *The n-input, (n-k)-output linear sequential circuit whose transfer function matrix is $\mathbf{H}^T(D)$ is called a syndrome-former, and has the property that $\mathbf{V}(D)\mathbf{H}^T(D)=\mathbf{0}$ if and only if $\mathbf{V}(D) \in C$.*

That is, when the n outputs of an encoder or generator $\mathbf{G}(D)$ are connected to the n inputs of the corresponding syndrome-former $\mathbf{H}^T(D)$, the *n-k* outputs of the syndrome-former are zero for all time.

Let $\mathbf{R}(D)$ be the received sequence when the codeword $\mathbf{V}(D)$ is transmitted over a channel. The sequence $\mathbf{R}(D) \in F((D))^n$ may be different from the transmitted codeword $\mathbf{V}(D)$ since the channel may introduce errors. Let $\mathbf{E}(D)$ be the error sequence. Then,

$$\mathbf{R}(D) = \mathbf{V}(D) + \mathbf{E}(D) \tag{3.1}$$

Definition 5. *The syndrome vector* $\mathbf{S}(D)$ *is defined as*

$$\mathbf{S}(D) = \mathbf{R}(D)\mathbf{H}^T(D) \tag{3.2}$$

3.3 Decoding of Convolutional Codes

Various algorithms are available for decoding convolutional codes. The Viterbi algorithm has received considerable attention [39][40]. This algorithm is maximum-likelihood and optimum for the decoding of convolutional codes. A difficulty with Viterbi decoding is the fixed amount of computation always required per decoder information block for a given code constraint length, and that this effort grows exponentially with the code constraint length. Under low noise conditions, a more flexible (adaptive) algorithm may be desirable.

Sequential decoding represents an alternative procedure [32][41]. The performance of sequential decoding is slightly less than optimal, but the decoding effort is basically independent of the code constraint length, so large constraint lengths can be used, and very low error probabilities can be achieved. An algebraic approach called majority-logic or threshold decoding can also be applied to convolutional codes [32][42]. Majority-logic decoding differs from Viterbi and sequential decoding in the fact that the error detection process is data-independent, and that the final decision in an information block is based only on one constraint length of the received blocks rather than on the entire received sequence. Because of the latter, majority-logic decoding usually results in inferior performance when compared with Viterbi or sequential decoding where the correction decisions are made based on at least five constraint lengths.

In all three approaches, some of the more important design parameters are the code constraint length which heavily determines the ability to detect errors, the coding

Base	**Numeric Symbol**
Adenine (A)	1
Guanine (G)	2
Cytosine (C)	3
Thymine (T)	4

Table 3.1: Mapping of DNA base-pairs

rate, i.e. the applied information redundancy, and the number of receiver quantization levels which can provide further enhancement such as weighting of each bit change by the signal-to-noise ratio for that bit, i.e. soft decision [43].

An ideal decoder would have performance approaching maximum-likelihood, but would have hardware complexity and speed comparable to a majority logic decoder. Table-driven decoding is one approach that provides such a combination.

3.4 Table-driven encoding and decoding

In the above sections, the concepts of convolutional encoding and decoding have been illustrated. It has been assumed that the source alphabet is binary. But it is possible that the source alphabet could belong to any finite field. A finite field is a field with a finite field order (i.e., number of elements), also called a Galois field. The order of a finite field is always a prime or a power of a prime. For the purpose of our genetic computations, we map the DNA bases to a field of five elements as shown in Table 3.1. So all the arithmetic related to coding must be performed in GF(5).

We will now explain the table-driven coding technique using an example based on rate-1/2 non-systematic codes. However, the technique can be used with any coding rate and with both systematic and non-systematic codes, although the performance is superior when non-systematic codes are used. A rate *k/n* convolutional code will have *n* generators, each of which operate on L input symbols at a time. The length L, which is also the length of each generator, is called the *constraint length.* The

Iteration	**No.of data elements**	**No.of encoded elements**
0	L	n
1	L+k	n+n
2	L+2k	n+2n
	...	
j	L+jk	n+jn

Table 3.2: One-to-one mapping in table-driven coding

encoded data is also referred to as *parity*.

The table-driven decoding method is based on the existence of a one-to-one mapping of a set of encoded elements and a set of data elements. Though the encoded elements are generated at a rate different from the data rate in rate-k/n coding, it is possible to find a relationship between the two which involves the same number of elements.

Consider the production of the encoded symbols during rate-k/n coding with constraint length L, as shown in Table 3.2. The first L data symbols produce n parity symbols. Each additional data symbol produces n more encoded bits. A **necessary** condition for the one-to-one mapping is that the number of data symbols and encoded symbols be the same. By requiring the number of data and encoded symbols in the jth iteration to be equal, and solving for j, we have

$$L + jk = n + jn \tag{3.3}$$

$$j = (L - n)/(n - k) \tag{3.4}$$

A parameter called the window length w is defined as

$$w = L + jk = n(L - k)/(n - k) \tag{3.5}$$

Therefore, w data and encoded symbols are needed for uniquely resolving a message encoded with code of length L, and w symbols must be taken at a time for it to be possible to have a one-to-one mapping. The method of calculating the parity sequence will now be illustrated with an example, for a rate-1/2 code, L = 3.

Let the generators be

$$\mathbf{g_1} = \begin{bmatrix} 1 & 2 & 4 \end{bmatrix}.$$

and

$$\mathbf{g_2} = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix}.$$

Let the input information be

$$\mathbf{u} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 2 & 0 & 0 \end{bmatrix}.$$

Each generator is applied on the input data, and the encoded symbols are calculated as follows.

1	2	4
0	0	1

$$v_0 = (1 \times 0) + (2 \times 0) + (4 \times 1) \quad mod - 5 \quad = \quad 4 \tag{3.6}$$

1	3	2
0	0	1

$$v_1 = (1 \times 0) + (3 \times 0) + (2 \times 1) \quad mod - 5 \quad = \quad 2 \tag{3.7}$$

Next, the generators are moved by k (in this case $k = 1$) positions to the right, and the procedure is repeated.

1	2	4
0	1	1

$$v_0 = (1 \times 0) + (2 \times 1) + (4 \times 1) \quad mod-5 \quad = \quad 1 \tag{3.8}$$

1	3	2
0	1	1

$$v_1 = (1 \times 0) + (3 \times 1) + (2 \times 1) \quad mod-5 \quad = \quad 0 \tag{3.9}$$

Repeating this procedure till the end of the data sequence, we get the following parity stream:

$$\mathbf{v} = \begin{bmatrix} 4 & 2 & 1 & 0 & \dots & 4 & 1 & 2 & 2 \end{bmatrix}.$$

The syndrome-former, or g-mask enables one to check if the received parity stream is correct or not. The g-mask is *and*-ed with the received parity bits and the result is summed modulo-5, giving the syndrome. The procedure is repeated after shifting the g-mask n positions along the parity stream. A *correct* g-mask would yield a syndrome of all zeros when applied over the received sequence, provided the received sequence has no errors.

The g-mask can be calculated from the generator coefficients using an algorithmic approach. The procedure will now be explained using the generators for the rate-1/2, L = 3 code. We now denote the generators as

$$\mathbf{g}^{(1)} = \begin{bmatrix} g_2^{(1)} & g_1^{(1)} & g_0^{(1)} \end{bmatrix}$$

and

$$\mathbf{g}^{(2)} = \begin{bmatrix} g_2^{(2)} & g_1^{(2)} & g_0^{(2)} \end{bmatrix}$$

The set of generators can also be described by the *generator matrix*, which contains the individual generators row-wise.

$$\mathbf{G}_{n\times L} = \begin{bmatrix} g_2^{(1)} & g_1^{(1)} & g_0^{(1)} \\ g_2^{(2)} & g_1^{(2)} & g_0^{(2)} \end{bmatrix}$$

We first choose a data vector **u** with an element 1 in a *unique* position as shown below. Note that the number of leading and trailing zeros should be sufficiently large.

$$\mathbf{u} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By applying the generators on the data sequence, we get the following parity sequence:

$$\mathbf{v} = \begin{bmatrix} \ldots & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & 0 & \ldots \end{bmatrix}$$

The g-mask **a** (or syndrome-former) has 6 elements for the present code.

$$\mathbf{a} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \end{bmatrix}^T$$

When the g-mask is applied over the *correct* parity elements, shifting appropriately each time, a syndrome of all zeros should result. We obtain the following equations when we apply the g-mask over the parity vector **v**

$$g_0^{(1)}a_4 + g_0^{(2)}a_5 = 0 \tag{3.10}$$

$$g_0^{(1)}a_2 + g_0^{(2)}a_3 + g_1^{(1)}a_4 + g_1^{(2)}a_4 = 0 \tag{3.11}$$

$$g_0^{(1)}a_0 + g_0^{(2)}a_1 + g_1^{(1)}a_2 + g_1^{(2)}a_3 + g_2^{(1)}a_4 + g_2^{(2)}a_5 = 0 \tag{3.12}$$

$$g_1^{(1)}a_0 + g_1^{(2)}a_1 + g_2^{(1)}a_2 + g_2^{(2)}a_3 = 0 \tag{3.13}$$

$$g_2^{(1)}a_0 + g_2^{(2)}a_1 = 0 \tag{3.14}$$

The above equations may be re-arranged in matrix form by defining a matrix $\mathbf{C}$ as follows

$$\mathbf{C}_{(w+k)\times(w+n)} = \begin{bmatrix} 0 & 0 & 0 & 0 & g_0^{(1)} & g_0^{(2)} \\ 0 & 0 & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} \\ g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} \\ g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & 0 & 0 \\ g_2^{(1)} & g_2^{(2)} & 0 & 0 & 0 & 0 \end{bmatrix}$$

The g-mask is related to $\mathbf{C}$ as follows

$$\mathbf{Ca} = \mathbf{0} \tag{3.15}$$

Invertibility of $\mathbf{C}$ guarantees that the code is non-catastrophic [37].

Using the example generators for the rate-1/2 code and solving (3.15), we get the set of possible g-masks to be

$$\mathbf{a}_1 = \begin{bmatrix} 1 & 4 & 3 & 3 & 2 & 1 \end{bmatrix}^T$$

$$\mathbf{a}_2 = \begin{bmatrix} 2 & 3 & 1 & 1 & 4 & 2 \end{bmatrix}^T$$

$$\mathbf{a}_3 = \begin{bmatrix} 3 & 2 & 4 & 4 & 1 & 3 \end{bmatrix}^T$$

$$\mathbf{a}_4 = \begin{bmatrix} 4 & 1 & 2 & 2 & 3 & 4 \end{bmatrix}^T$$

There are four g-masks since we are dealing with elements in GF(5). The all-zero g-mask is excluded, since it is a trivial solution. Only one of the g-masks is “unique” and the others are just its multiples. When the elements of $\mathbf{a}_1$ are multiplied (modulo-5) by 2, we obtain the elements of $\mathbf{a}_2$. When the elements of $\mathbf{a}_1$ are multiplied by 3 and 4, we get the elements of $\mathbf{a}_3$ and $\mathbf{a}_4$ respectively.

In the absence of any errors, it can be verified that the syndrome is all zero. The above g-masks when applied over the parity sequence $\mathbf{v}$, shifting by n (here, 2) positions each time, gives a syndrome of zero at every location.

Chapter 4

Algorithm for finding Generators

As stated previously, the transfer of biological information can be modelled as a communication channel, with the DNA sequence as input and the polypeptide amino acid sequence as the output. As in error-protected information channels, redundancy occurs within genomic sequences. The DNA encoded message, in its double-helix form, is doubly redundant. Hence, we could consider un-replicated DNA as a half-rate systematic code, i.e., for each item (base) there is a corresponding parity item (complementary base). At the nucleic-acid level, messenger RNA (mRNA) sequences are mapped to amino-acids by grouping three nucleic acid bases together to form a codon.

A codon, or three base nucleic acid vector, is mapped to a single amino acid information item. This process which occurs during translation, could be viewed as the decoding of a rate one-third code, i.e., each amino acid is encoded using three "parity" items. We assume that the mRNA has been encoded as a rate 1/3 code with constraint length L = 5. This assumption is the same as in previous work [45], but the method of constructing the model differs. In this work, we have developed an algorithmic approach for finding the generators of a convolutional code when the g-mask (syndrome former) is known. This approach will be illustrated in this chapter with an example.

4.1 Theoretical Basis

It is also known that leader regions of the messenger RNA (mRNA), and other prokaryotic regulatory regions contain consensus sequences which in some way signal or control translation. Examples are the Shine-Dalgarno (SD) sequence in bacterial mRNA, and the TATA box and the Pribnow box in double helix DNA [1]. Specifically, the SD sequence always occurs before the start codon (usually AUG) of a translating mRNA sequence in *E.coli.* Our hypothesis is that, if there is a method in place that checks for the validity of the leader sequence (which includes SD), the ribosome would somehow have a way of recognizing it. Assuming there is a validating relationship among the leader sequence bases and/or codons, and assuming that the ribosome has an exposed region which is in contact with the mRNA leader for validation purposes, one may conjecture that the leader sequence may have embedded in it (or may be modelled as) an error detecting code such as a block code or a convolutional code. In *Escherichia coli* (*E.coli*), the exposed part of the 16S ribosomal RNA (rRNA) binds with the mRNA leader sequence during the initiation of translation. The same exposed part appears to remain in contact with mRNA (in addition to P and A sites and possibly other ribosomal regions) during the translation process.

The specific form of coding theory we apply in the present analysis is called table-driven convolutional coding. A brief overview of this theory has been covered in Chapter 3. The mathematics of coding is carried out over a finite field, also referred to as Galois Field (GF), using a set of discrete source symbols [32]. Since Uracil (U) replaces Thymine (T) in messenger RNA, the mRNA bases are mapped to the field of five, as shown in the Table 4.1.

This calls for arithmetic operations such as addition and multiplication to be

Base	**Numeric Symbol**
Inosine (I)	0
Adenine (A)	1
Guanine (G)	2
Cytosine (C)	3
Uracil (U)	4

Table 4.1: Mapping of bases to GF(5) elements

carried out in GF(5). The assignment of numerical values to the nucleotides is consistent with the bonding characteristics of the base pairs. In digital communication theory, checking the validity of an encoded message corresponds to matching of the message, piecewise, to a syndrome former (a sequence of symbols also known as the g-mask). A correct message is expected to yield zero syndrome when matched against the syndrome former. The syndrome former symbols (which will be the same symbols used to encode the sequence, A, U, G , C and I in our case) are “multiplied” (in this case base five) with the corresponding portion of the encoded sequence, and these products are then “added” (base five) to form one symbol of the syndrome. The syndrome former is then moved downstream by a code-specific distance and the process is repeated. If the resulting syndrome is zero, the message is assumed to be error-free. However, the syndrome can contain any of the five operational symbols. Also, it must be noted that if the number of “errors” exceeds the error-detecting capability of the code, the message may be flagged as correct even when it is not. It is interesting to note that this reminds us very much of what happens in the case of cancers where an erroneous genetic code is allowed to translate.

It has been stated that constructing the g-mask is one of the primary challenges of table-based decoding, as applied to genetic coding systems [27]. A good g-mask which gives nearly zero-syndrome patterns at appropriate locations along the mRNA leader sequence will prove to be a viable tool for calculating the generators.

In previous research [26][27], the g-mask has been constructed from the exposed part of the 16S rRNA. A plausible distinction between the syndrome patterns for

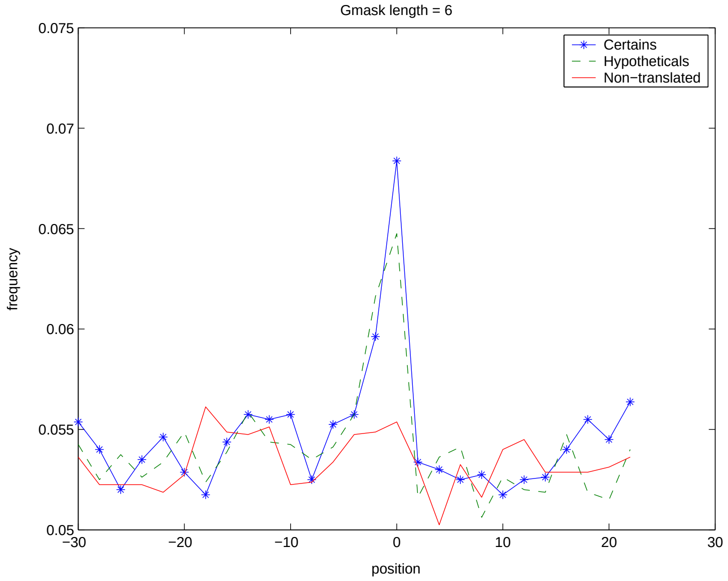

Figure 4.1: Syndrome patterns obtained using g-mask

translated and non-translated sequences has been obtained. This has led us to believe that the convolutional code is an acceptable coding paradigm for genetic sequences.

We will now demonstrate the validity of the convolutional code hypothesis by considering a rate 1/2 code with constraint length L = 3. This is similar to May et al.'s previous work [27]. As per the rules of table-driven coding, this code must have a g-mask of length 6. The results of sliding a g-mask of length 6, chosen from the exposed part of the 16S ribosome, over the mRNA leader sequences are shown in Figure 4.1.

The genetic g-mask is formed from subsets of contiguous bases on the exposed part of the 16S rRNA. There are 8 possible g-masks of length 6 that can be formed from the 13-base long exposed part. Each g-mask is applied on three different sets of sequences: "certains" or translated sequences, "hypotheticals" or hypothetically-translated sequences and non-translated sequences. At every position, we measure how well a g-mask decodes the leader sequence at that position by calculating the syndrome at that position for all participating sequences (1000 in our case). Then we consider groups of adjacent syndrome values as a “distance pattern” [27]. Finally, at each position, we find the most frequent “distance pattern” and compute its relative frequency. If the pattern were truly random we might expect that relative frequency to be around 0.04. In Figure 4.1, relative frequency is shown on the vertical axis. The larger the frequency, the more the g-mask favors the sequence. The horizontal axis indicates position in the mRNA leader sequence, with zero corresponding to the location of the first base of the start codon (usually AUG). A clear peak is observed around position zero in the translated and hypothetically-translated sequences, the peak being more prominent in the former case. This indicates that the convolutional coding model may have the ability to distinguish between the three types of sequences. The results do differ from previous work [27] since we find the frequency of syndrome symbol pairs obtained from *all* the g-masks at each position. In order to fully understand the implications of the model based on the convolutional coding theory, we need to be able to back-track from the syndrome former (which can be chosen complementary to the SD sequence in the case of bacteria), to the codes that actually can produce that syndrome former.

In general, this is a many to one relationship. We would expect that to be the case since the DNA information engine may employ errors to control translation efficiency and to “lock-in” into the translation frame [44]. We do assume that in the present study, the most important feature of a good code is its ability to distinguish the translated sequences from the hypothetical and non-translated sequences. Genetic algorithms have been used with considerable success in constructing convolutional code models for translation initiation [45]. The optimal codes possess the following features: high similarity of the g-mask to the 3’ end of the 16S rRNA, ability of the g-mask to recognize key regions on the mRNA leader such as the non-random and the Shine-Dalgarno domains, and potential to detect valid and invalid leader sequences. The genetic algorithms-based method searches the space of all possible codes of a given description, to find optimal generators. In this work, instead of using search-based methods, we apply analytical methods to find good generators.

4.2 Method of finding generators from g-mask

We now describe a numerical algorithm for finding the generators of a convolutional code, when the g-mask is known. The algorithm is based on concepts of matrix theory and linear algebra, and is computationally simple. The syndrome former and the generators of a convolutional code have the following relationship:

$$\mathbf{Ca} = \mathbf{0} \tag{4.1}$$

C is the matrix containing the generator coefficients, and **a** is the g-mask. An example of forming the above equation is described in the previous chapter. For a catastrophic code generator, **C** is not invertible [37]. This serves as a test for catastrophicity, and will hence be referred to as the "rank test". If r = k/n is the rate of the code, and L is the length of each generator, number of generator coefficients is (nL).

When the coefficients of the g-mask **a** are known, equations (3.10)-(3.14) may be re-arranged in the form

$$\mathbf{Ac} = \mathbf{0} \tag{4.2}$$

The matrix **A** contains elements of the g-mask.

$$\mathbf{A}_{(w+k)\times(w+n)} = \begin{bmatrix} 0 & 0 & a_0 & 0 & 0 & a_1 \\ 0 & a_0 & a_2 & 0 & a_1 & a_3 \\ a_0 & a_2 & a_4 & a_1 & a_3 & a_5 \\ a_2 & a_4 & 0 & a_3 & a_5 & 0 \\ a_4 & 0 & 0 & a_5 & 0 & 0 \end{bmatrix}$$

The vector **c** contains the generator coefficients.

$$\mathbf{c} = \begin{bmatrix} g_0^{(1)} & g_1^{(1)} & g_2^{(1)} & g_0^{(2)} & g_1^{(2)} & g_2^{(2)} \end{bmatrix}^T$$

Given the g-mask coefficients, (4.2) may be solved to find the generator coefficients. This gives us an elegant way to find the generators of the convolutional coding system, when the g-mask is known.

For the rate-1/2 code discussed in the previous chapter and the g-mask $\mathbf{a}_1$, the $\mathbf{A}$ matrix is

$$A = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 1 \\ 0 & 2 & 3 & 0 & 1 & 3 \\ 2 & 3 & 1 & 1 & 3 & 4 \\ 3 & 1 & 0 & 3 & 4 & 0 \\ 1 & 0 & 0 & 4 & 0 & 0 \end{bmatrix}.$$

Solving (4.2) using this matrix yields the set of four possible generators

$$\mathbf{G}_1 = \begin{bmatrix} 3 & 1 & 2 \\ 3 & 4 & 1 \end{bmatrix}$$

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 2 \end{bmatrix}$$

$$\mathbf{G}_3 = \begin{bmatrix} 4 & 3 & 1 \\ 4 & 2 & 3 \end{bmatrix}$$

$$\mathbf{G}_4 = \begin{bmatrix} 2 & 4 & 3 \\ 2 & 1 & 4 \end{bmatrix}$$

The set of generators which we used to find the g-mask are in $\mathbf{G}_2$. There is always a one-to-one correspondence between the generators and the g-mask. A “good” g-mask would give strongly correlated patterns of zero-syndrome when applied over the portions of the leader sequence that distinguish coding from non-coding sequences. Techniques for constructing optimal g-masks will be illustrated in the next chapter.

Chapter 5

Results

In this chapter, the results of this research will be presented. Two important methods for constructing the *genetic* g-mask will be discussed. The generators of the convolutional code model will be calculated using these g-masks. The coding theory model states that the generators encode the genetic information. We assume that the DNA represents the "correct" genetic information. The phenomena of replication and transcription explain the action of *encoding* the genetic information, and produce mRNA as the output.

We will investigate the plausibility of each generator by encoding the DNA sequences using them, and decoding using the appropriate set of g-masks.

The most plausible generators will be able to produce encoded sequences which, when decoded, would yield a clear distinction between the translated and non-translated sequences.

5.1 G-mask from 16S ribosome

One method of obtaining the g-mask is from contiguous locations on the exposed part of the 16S ribosomal RNA. This in fact corresponds to reality, since the rRNA binds to the mRNA leader sequences during translation initiation. The plausibility of using such g-masks has been demonstrated previously [26][27].

The *E.coli* K-12 strand [47] is used as a model organism to analyze our coding model. The length of the g-mask for a rate 1/3 code, L= 5, is

$$gmlen = 9$$

The exposed part of the 16S rRNA that is involved in binding is of length

$$lexp16S = 13$$

So the number of g-masks is given by

$$numgmask = lexp16S - gmlen + 1$$

In this case, the number of possible g-masks that can be formed are 13-9+1 = 5. There are two g-masks in a rate 1/3 code, so we choose all possible pairs of g-masks from the available five sequences of length nine.

There is one caveat here! All the possible g-masks cannot be *inverted* to obtain the generators. In order to find the generator coefficients **c**, the matrix **A** should be reducible into row-echelon form. In this case, it is found that all the g-masks are invertible. We calculate one non-catastrophic generator from each g-mask pair.

The following procedure illustrates the approach:

- Form a list of g-masks from consecutive locations of the exposed part of the 16S ribosome. Use pairs of g-masks in the computations, since the rate 1/3 code has 2 g-masks.
- Test whether the matrix **A** is invertible in each case. Select only those g-masks

which can be inverted to calculate the generators.

- For each g-mask pair that can be inverted, form a set of possible generators. Test each generator for non-catastrophicity.
- Calculate one non-catastrophic generator from each g-mask pair. This gives the set of generators to be used in encoding the DNA sequence.

The entire genome of *E.coli* has 4288 protein-coding genes [46]. The *E.coli* genome [47] consists of sequences which are denoted as CDS. This means that the particular sequence referred to is, hypothetically, a coding sequence. Out of a total of 4288 sequences which are denoted CDS, only 1172 are known to code for protein [44]. These *certain* sequences are first extracted and stored in a file. A list of *certain* coding sequences in *E.coli* is given in the appendix. The remaining CDS sequences, i.e. *hypotheticals* are also stored in a separate file. The genome is parsed to find other sequences which possess a start codon (AUG) but are not denoted as CDS. These *non-coding* sequences are also stored separately. Out of the 1172 *certain* coding sequences, the first 1000 are chosen to construct our model. Correspondingly, the number of *hypotheticals* and *non-coding* sequences is chosen to be 1000, and the length of each sequence is 60, centered around the start codon. The base A of the AUG start-codon triplet is assigned position 0.

With this data available, the encoding of the DNA sequences is performed using the calculated generators. Table-driven encoding is used to obtain the encoded sequences [34]. The decoding of these sequences is performed using the g-masks extracted from the exposed part of the ribosome. There are five g-masks in this case, and all of them are used in decoding the obtained mRNA sequences.

The results of decoding are given in Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6, Figure 5.7, Figure 5.8, Figure 5.9 and Figure 5.10. The x-axis shows the position of the base on the mRNA leader sequence (-30 to +29).

At each position, the syndrome distance obtained from decoding using the g-mask is calculated. There are totally 20 syndrome distances in the mRNA leader sequence

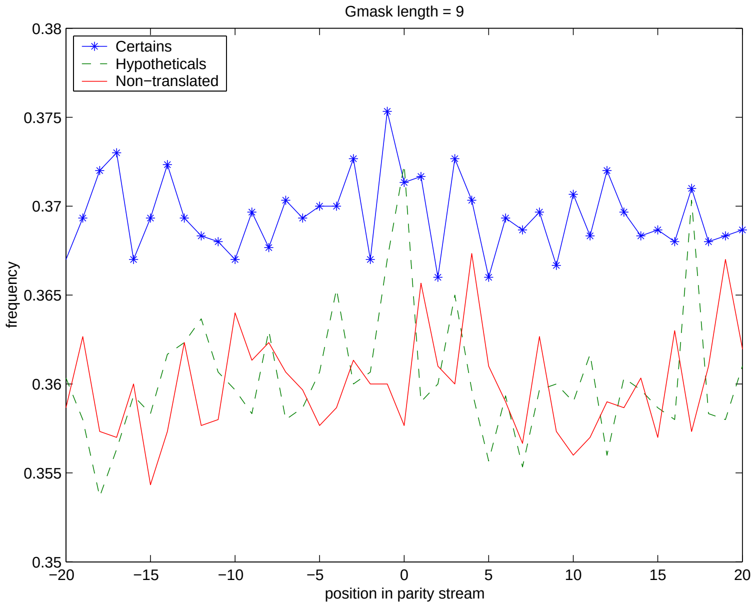

Figure 5.1: Decoding the generator_1 sequences: exposed part of ribosome

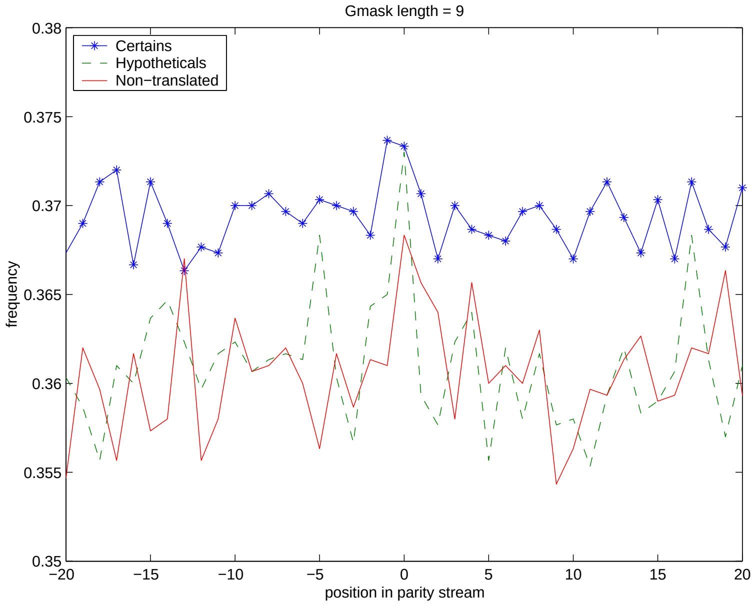

Figure 5.2: Decoding the generator_2 sequences: exposed part of ribosome

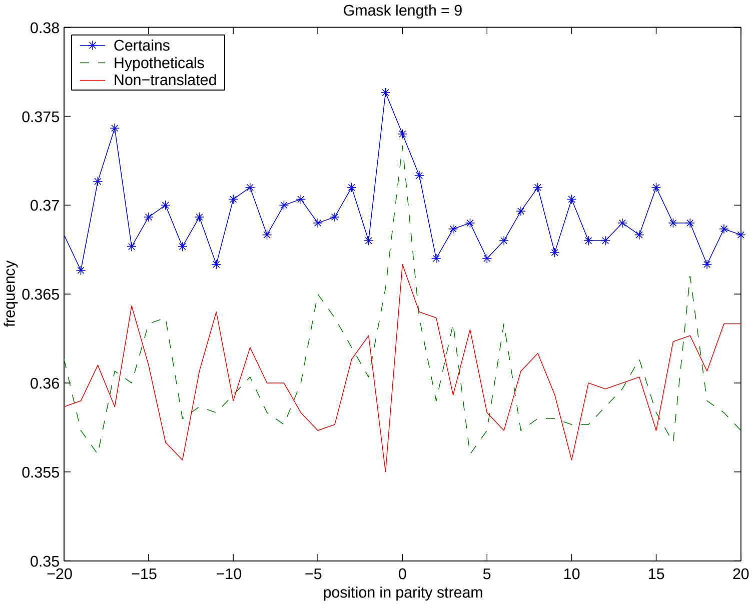

Figure 5.3: Decoding the generator$_3$ sequences: exposed part of ribosome

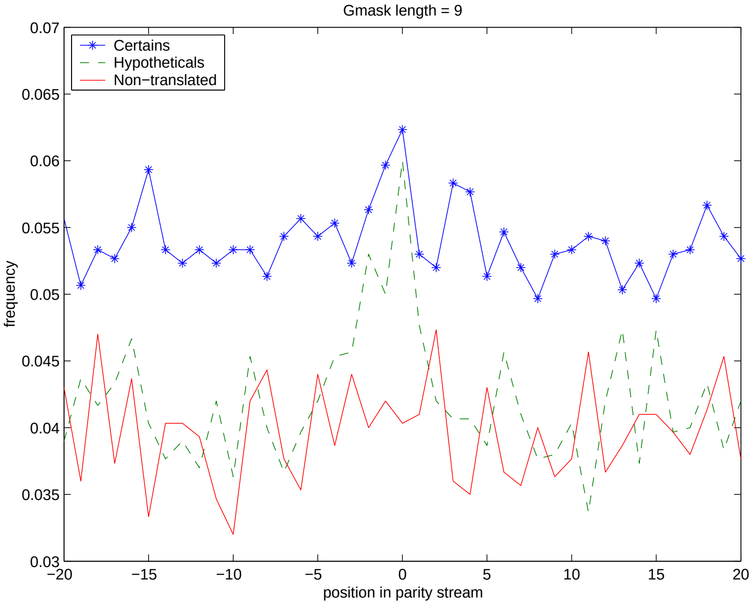

Figure 5.4: Decoding the generator$_4$ sequences: exposed part of ribosome

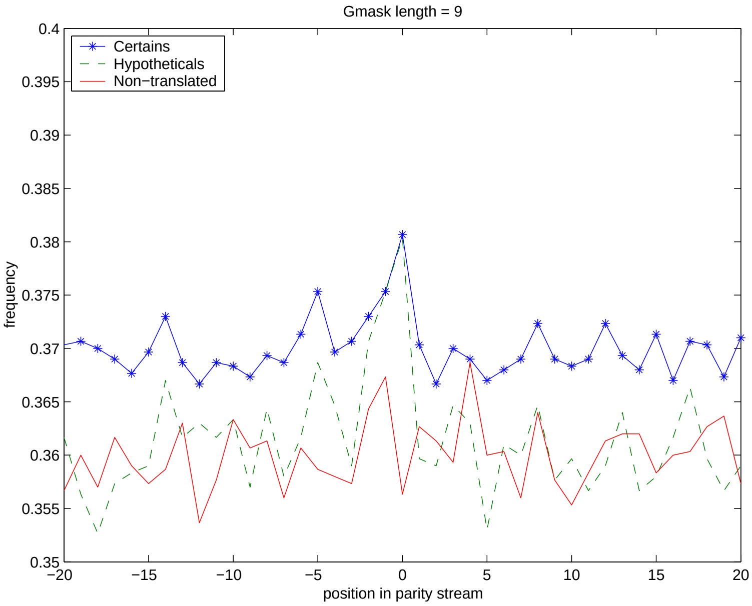

Figure 5.5: Decoding the generator$_5$ sequences: exposed part of ribosome

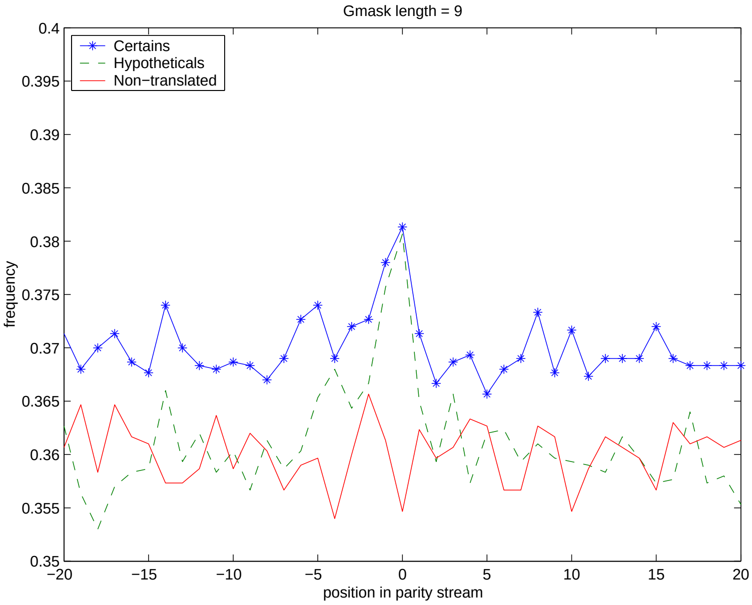

Figure 5.6: Decoding the generator$_6$ sequences: exposed part of ribosome

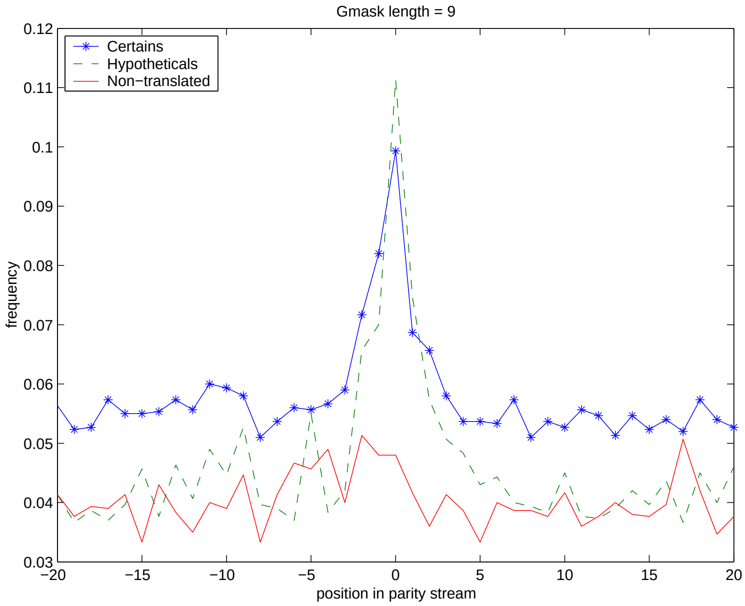

Figure 5.7: Decoding the generator$_7$ sequences: exposed part of ribosome

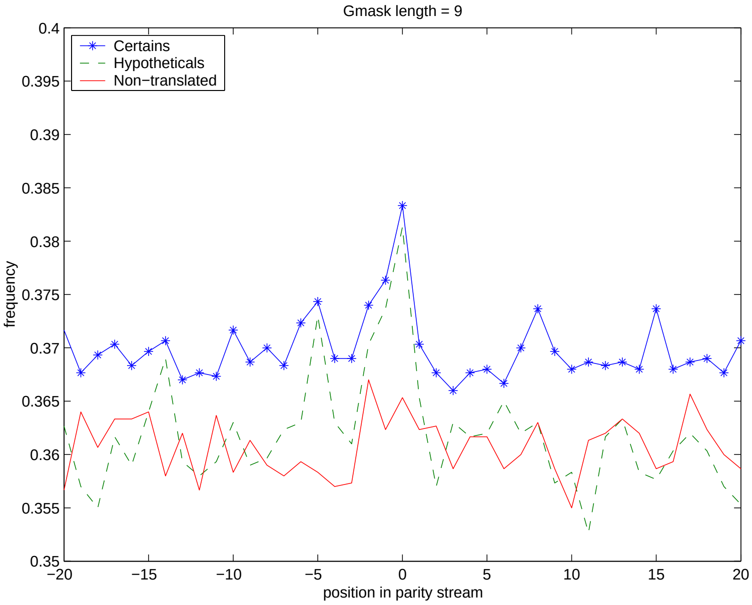

Figure 5.8: Decoding the generator$_8$ sequences: exposed part of ribosome

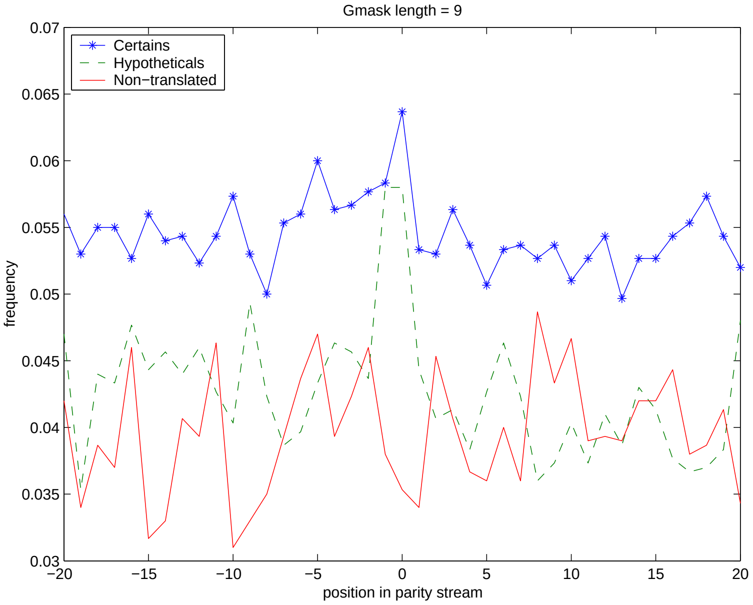

Figure 5.9: Decoding the generator$_9$ sequences: exposed part of ribosome

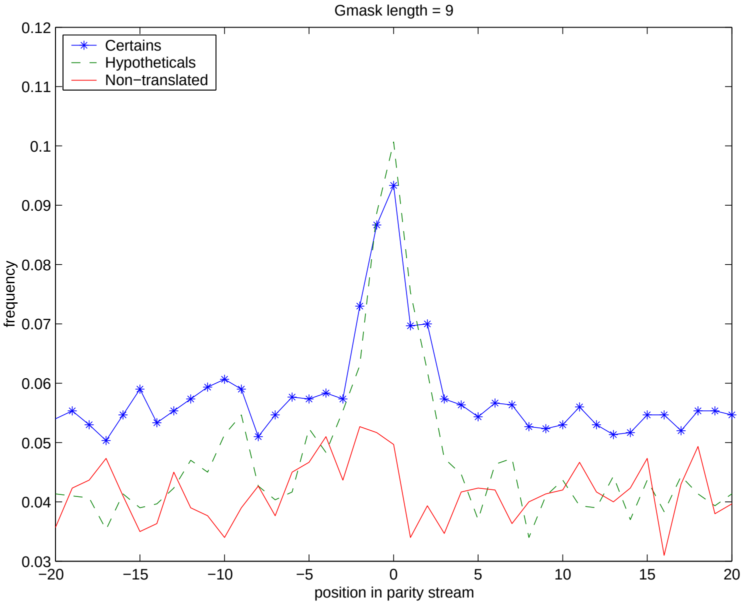

Figure 5.10: Decoding the generator$_{10}$ sequences: exposed part of ribosome

of length 60, since the g-mask moves in steps of n = 3. Each of the 1000 sequences are decoded using each g-mask. The coding sequences have tandem repeats such as the Shine-Dalgarno sequence a few bases upstream of the start codon. This causes the syndrome distance values to be identical in this range, since they are decoded with the same g-mask. The maximum frequency of 2-symbol distance patterns will show a prominent peak in this range. Two-symbol syndrome patterns obtained at each position from all the decoded sequences are used in finding the fequency. The y-axis shows the maximum frequency of 2-symbol syndrome distance patterns [27] at each position.

We observe that there is a marked distinction in the graph around zero, which is to be expected, based on the biology of the system. The coding sequences have a higher value of maximum frequencies than the non-coding sequences.

5.2 Constructing G-mask from binding patterns

We will now illustrate a method of calculating the g-mask based on binding patterns between the ribosome and the exposed mRNA leader sequences. A true test of a g-mask's goodness is its ability to distinguish the coding from the non-coding sequences. In coding theory, we consider zero syndrome as an indicator of correctness. This means that the g-mask which produces consistent patterns of zero syndrome for the coding sequences would be considered an *ideal* g-mask.

The theory of binding patterns has been presented in [44] [45]. The basic idea is that coding sequences can be identified from the way in which they bind to the mRNA leader sequences. The sequences which possess high complementarity to the 3' end of the exposed part of the 16S rRNA have a higher probability of coding for protein. This is because greater binding energy is released if there are consecutive bonds between the ribosome and the mRNA leader.

It is known, from an analysis of binding patterns, that the largest difference between the consecutive binding patterns between coding and non-coding sequences occurs at position -14 [45]. We first find a list of sequences having strong binding patterns in this region. These are the *most certain* coding sequences. We choose the g-mask to be complementary to the mRNA leader sequence, centered around position -14. We then apply these g-masks on the other coding sequences, and find five g-masks that give the least overall syndrome.

The following procedure illustrates the method of finding the g-mask based on binding patterns:

- Select *certain* coding sequences from the *E.coli* genome. For each sequence, find the position-wise binding vectors.
- In each binding vector, denote the presence of an exact complement as 1, and absence as 0.
- Calculate the maximum number of consecutive ones in each binding vector. Sum this number, say M1 over all positions along the coding sequence. Select the sequences having the highest values of this sum. These can be characterized as *most favorable* sequences for binding with the exposed part of the ribosome.
- Choose the g-mask to be complementary to the coding sequence, centered around position -14. The hypothesis is that this g-mask would surely indicate, by means of the low syndrome values, that the corresponding sequence is a *certain* coding sequence.
- Apply the chosen set of g-masks on another set of coding sequences, and find the syndrome at each location. Sum the overall syndrome per sequence. Choose a fixed number of g-masks which give least overall syndrome.

Each of these g-masks is now used to calculate the generators of the corresponding rate 1/3 code. The results of decoding the mRNA leaders using these g-masks

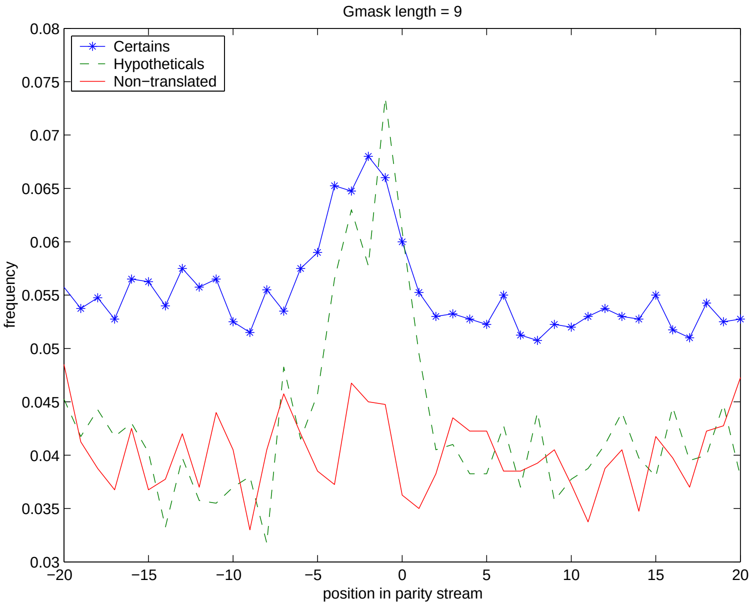

Figure 5.11: Decoding the generator$_1$ sequences: binding patterns

are presented in Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, Figure 5.15, Figure 5.16, Figure 5.17.

In accordance with our predictions, the generators of the convolutional code model show a clear distinction between translated and non-translated sequences. We will now evaluate the *fitness* of each generator based on its ability to distinguish between translated, hypothetically translated and non-translated sequences. The best g-mask does not necessarily produce an all-zero syndrome. This leads to the generators being non-ideal in most of the cases. We therefore need a different metric to evaluate the performance of the generators.

The metric we compute is the ratio of average frequencies in the range -10 to +10 along the mRNA leader, since this is where we observe the greatest distinction between the translated and non-translated sequences. This ratio can be taken as a measure in evaluating the fitness of each generator. We define two metrics, Fitness1 and Fitness2, to evaluate the plausibility of each generator. A higher value of both these parameters indicates greater plausibility of the generator.

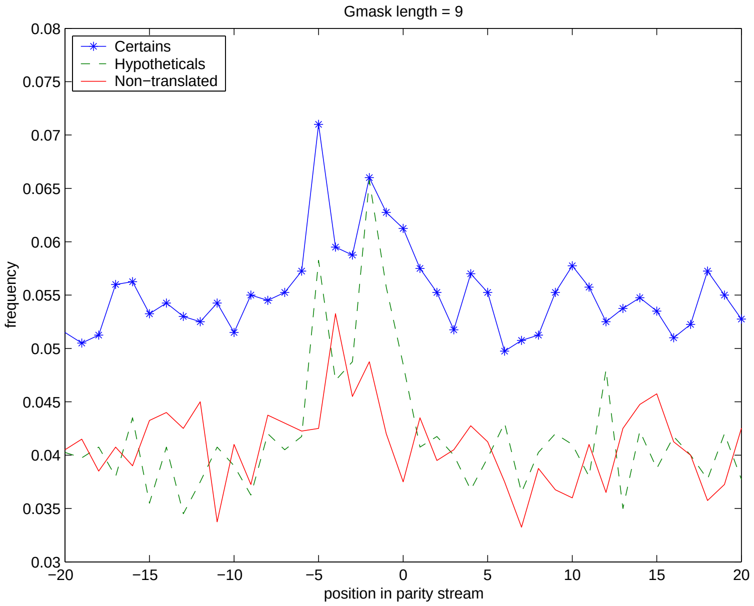

Figure 5.12: Decoding the generator$_2$ sequences: binding patterns

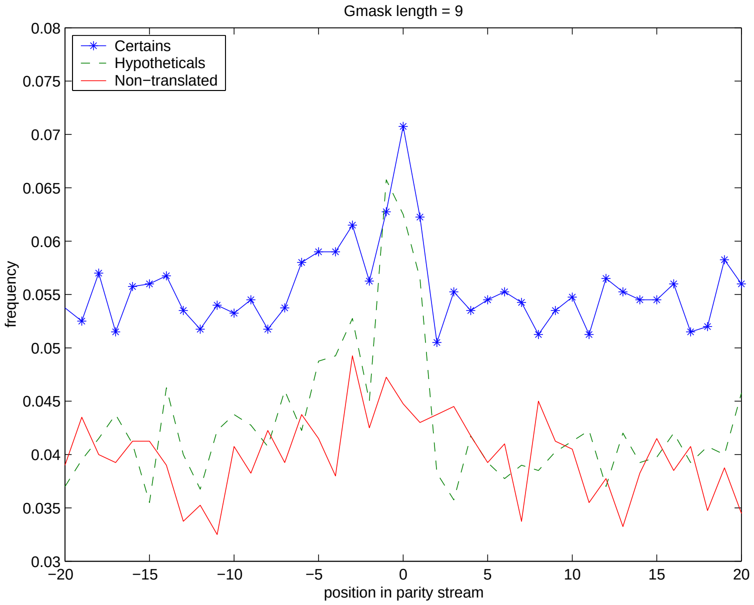

Figure 5.13: Decoding the generator$_3$ sequences: binding patterns

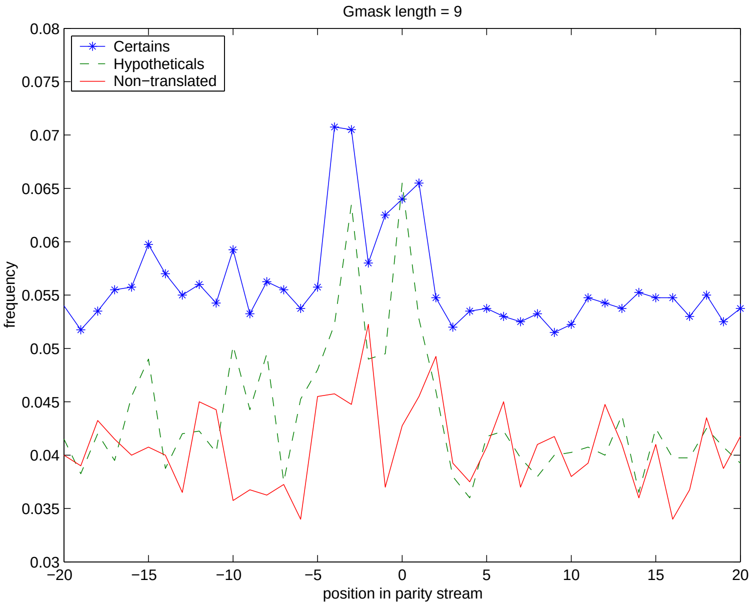

Figure 5.14: Decoding the generator$_4$ sequences: binding patterns

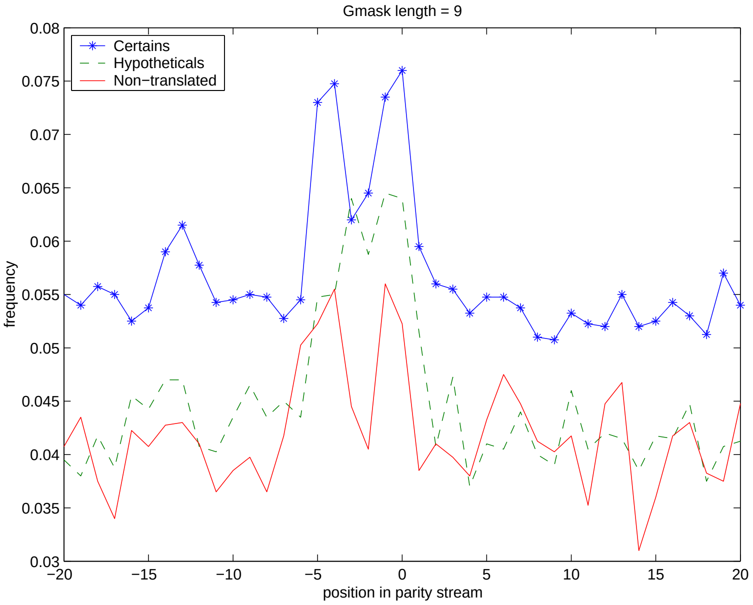

Figure 5.15: Decoding the generator$_5$ sequences: binding patterns

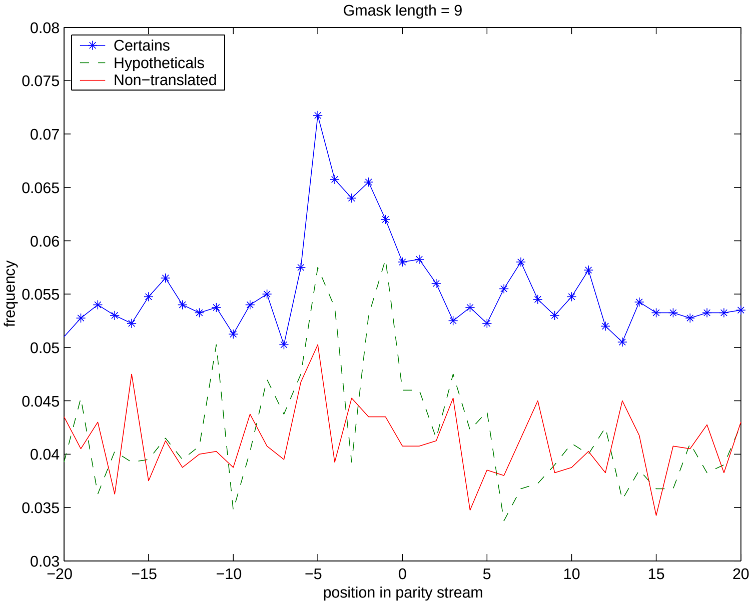

Figure 5.16: Decoding the generator$_6$ sequences: binding patterns

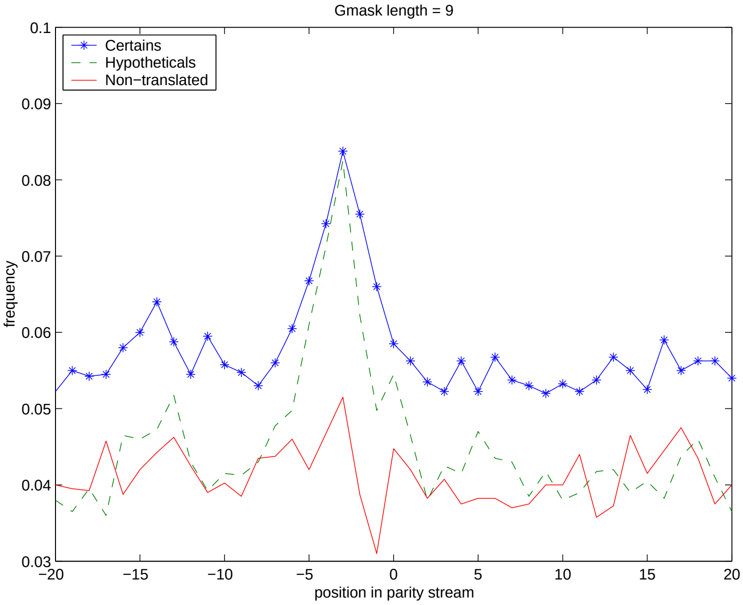

Figure 5.17: Decoding the generator$_7$ sequences: binding patterns

Let us denote

$$f_a^{cert} = average \;\; frequency \;\; of \;\; certain \;\; coding \;\; sequences \tag{5.1}$$

$$f_a^{hyp} = average \;\; frequency \;\; of \;\; hypothetical \;\; coding \;\; sequences \tag{5.2}$$

$$f_a^{non} = average \;\; frequency \;\; of \;\; non-coding \;\; sequences \tag{5.3}$$

$$Fitness1 = f_a^{cert}/f_a^{hyp} \tag{5.4}$$

$$Fitness2 = f_a^{cert}/f_a^{non} \tag{5.5}$$

Based on these metrics, the fitness graphs shown in Figure 5.18, Figure 5.19 are obtained.

The hypotheticals have an average frequency which is close to that of the certains, since the value of Fitness1 is close to 1. A higher value of Fitness2 indicates that the generator is good, since such a generator would clearly distinguish the coding regions from the non-coding regions. In this regard, among the generators calculated from the exposed part of the 16S shown in Figure 5.18, Generator$_{10}$ can be considered the best, since it has a reasonably high value of Fitness1 and the highest Fitness2 value.

In the case of generators calculated from binding patterns, as shown in Figure 5.19, we follow a similar procedure for evaluation of fitness. Generator$_4$ may be considered the best in this case, since it has the highest value of Fitness2 and a reasonably high value of Fitness1.

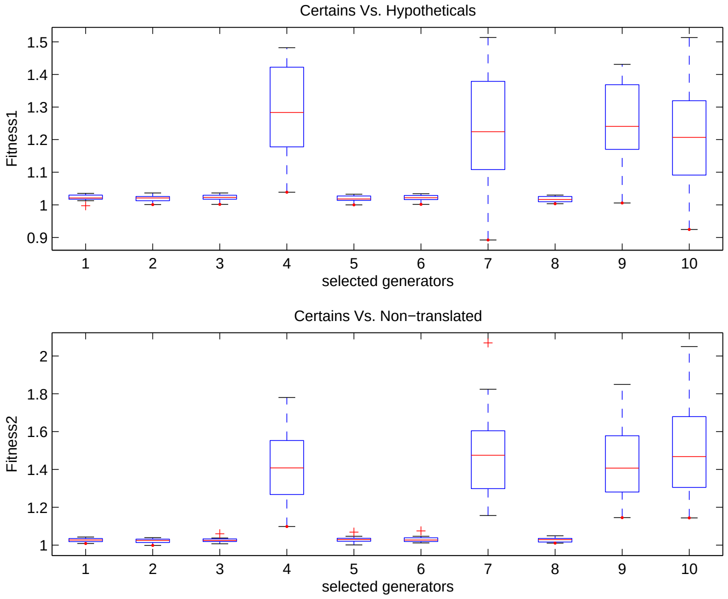

Figure 5.18: Fitness of generators, with confidence intervals, calculated from exposed part of ribosome

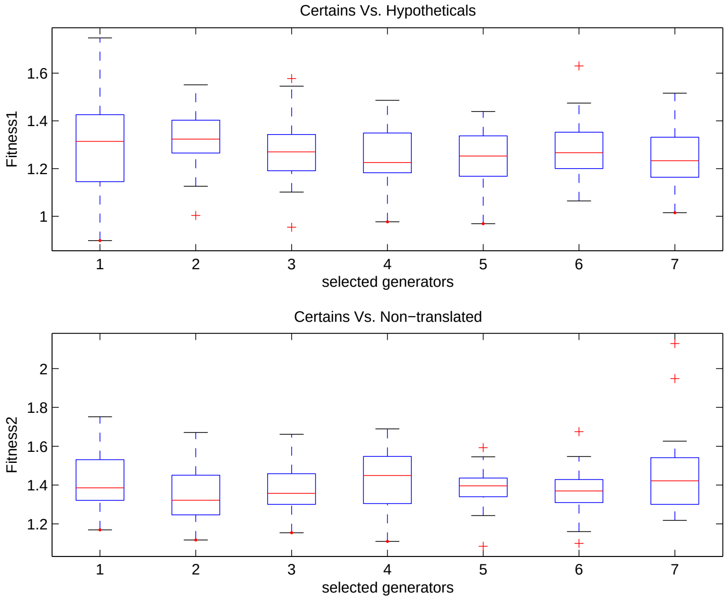

Figure 5.19: Fitness of generators, with confidence intervals, calculated from binding patterns

Chapter 6

Summary and Conclusions

We will now discuss the significance of the results presented in the previous chapter. To start with, let us analyze the method of calculating the generators.

Using the g-masks from the exposed part of the ribosome, or calculating them from binding patterns makes the method computationally less intensive. Several possible generators can be computed and their fitness or *plausibility* can be measured easily using our method.

For the analysis presented, we have used 1000 sequences in each category, i.e., a 1000 *certain* coding sequences, a 1000 *hypothetical* coding sequences, and a 1000 *non-coding* sequences.

The main research contributions presented in this thesis are:

- A method to find effective *genetic* g-masks based on the theory of binding patterns.
- An efficient method to algebraically calculate the generators of the convolutional code of desired rate from the g-mask
- Development of metrics to evaluate the efficiency of the calculated generators

The fitness plots in figures Figure 5.18 and Figure 5.19 show that the generators calculated from the exposed part of the 16S ribosome have a smaller variance than those calculated based on binding patterns. This shows that generators calculated from the exposed part of the ribosome perfom better.

A sequence-based model for prokaryotic translation initiation has been presented using the theory of convolutional codes. We have devised a novel method for finding the generators of the convolutional code model, using table-based coding techniques. The performance of each generator has been evaluated based on its ability to produce a clear distinction between translated, hypothetically-translated and non-translated sequences.

In this thesis, we attempted to find plausible convolutional code models based on the interactions at the translation initiation stage. We assume that the code is one-dimensional and has a fixed rate and constraint length. Future work in this direction would be to investigate the concatenated coding approach, and develop appropriate coding models for each step of the genetic coding process. Coding models may also be used to predict secondary and tertiary structure of proteins, based on their one-dimensional sequences. It would be interesting to construct coding models in higher dimensions for this purpose.

Bibliography

[1] Benjamin Lewin, *Genes V*, Oxford University Press, New York, NY, 1995.

[2] Klug and Cummings, *Concepts of Genetics*, Prentice Hall, Upper Saddle River, NJ, 5 edition, 1997.

[3] D. Anastassiou, "Frequency-Domain Analysis of Biomolecular Sequences," *Bioinformatics*, vol. 16, no. 12, December 2000, pp. 1073-1081.

[4] D. Anastassiou, "DSP in Genomics," *Proceedings, IEEE International Conference ICASSP 2001*, Salt Lake City, Utah, May 2001.

[5] Watson J. D. and Crick F. H. C., "Molecular structure of Nucleic Acids", *Nature* 171, 737-738 (1953).

[6] Francois Rodolphe and Catherine Mathe, "Translation Conditional Models for Protein Coding Sequences," *Journal of Computational Biology*, Volume 7, Numbers 1/2, 2000.

[7] Orlov Yu.L., Filippov V.P., Potapov V.N., Kolchanov N.A. "Complexity: Software Tools for Analysis of Information Measures of Genetic Texts", *Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, 2002.

[8] Hidde de Jong, "Modeling and Simulation of Genetic Regulatory Systems: A Literature Review", Journal of Computational Biology, Volume 9, Number 1, 2002.

[9] D. A. Mac Donaill, “The Role of Error-Coding in Shaping the Nucleotide Alphabet: Nature’s Choice of A, U, C and G”, 25^{th} *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2003.

[10] L. Kari, S. Konstantinidis, “Static and Dynamic Properties of DNA Languages”, 25^{th} *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2003.

[11] Rosen G.L., Moore J.D., “Investigation of coding structure in DNA”, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, Volume: 2 , April 6-10, 2003 Page(s): 361 -364

[12] G. Battail, “Does Information theory explain biological evolution?”, *Europhysics Letters*, 40(3), pp. 343-348 (1997).

[13] Amit Marathe, Anne E. Condon, Robert M. Corn, ”On Combinatorial DNA Word Design”, *Journal of Computational Biology*, Volume 8, Number 3, 2001, pp. 201-219.

[14] Didier G. Arques and Christian J. Michel, “A code in the protein coding genes,” *BioSystems*, vol.44, pp.107–134, 1997.

[15] D.W. Schultz, M. Yarus, “Transfer RNA mutation and the Malleability of the Genetic Code”, *Journal of Molecular Biology*, 235 (1994) pp. 1377-1380.

[16] G. Houen, “Evolution of the genetic code: the nonsense, antisense and antinonsense codes make no sense”, *BioSystems* 54 (1999), pp. 39-46.

[17] Leibovitch L.S., Tao Yi, Todorov A.T., Levine L., “Is there an Error Correcting Code in the DNA?”, *Biophysical Journal*, Vol. 71, pp. 1539-1544, 1996.

[18] Hubert P. Yockey, “Information theory and molecular biology” Cambridge University Press, 1992

[19] Thomas D. Schneider, Gary D. Stormo, Larry Gold, and Andzej Dhrenfeucht, “Information Content of Binding Sites on Nucelotide Sequences,” *Journal of Molecular Biology*, vol. 188, pp. 415–431, 1986.

[20] Thomas D. Schneider, “Information content of individual genetic sequences,” *Journal of Theoretical Biology*, vol. 189, pp. 427–441, 1997.

[21] Manfred Eigen, “The origin of genetic information: viruses as models,” *Gene*, vol.135, pp. 37–47, 1993.

[22] Richard E. Blahut, *Theory and practice of error control codes*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1983.

[23] Ramon Roman-Roldan, Pedro Bernaola-Galvan, and Jose L. Oliver, “Application of information theory to DNA sequence analysis: a review,” *Pattern Recognition*, vol.29, no.7, pp. 1187–1194, 1996.

[24] Elebeoba E. May, “Comparative Analysis of Information Based Models for Initiating Protein Translation in Escherichia coli K-12,”, M.S. thesis, NCSU, December 1998.

[25] G. Battail, “Does information theory explain biological evolution?,” *Europhysics Letters*, vol.40, no.3, pp.343–348, November 1997.

[26] Elebeoba E. May, Mladen A. Vouk, Donald L. Bitzer, and David I.Rosnick, “Coding Model for Translation in E. coli K-12,” *First Joint Conference of EMBS-BMES*, 1999.

[27] Elebeoba E. May, Mladen A. Vouk, Donald L. Bitzer, and David I.Rosnick, “The Ribosome as a Table-Driven Convolutional Decoder for the Escherichia coli K-12 Translation Initiation System,” *World Congress on Medical Physics and Biomedical Engineering Conference*, 2000.

[28] Lila Kari, Rob Kitto, and Gabriel Thierrin, “Codes, Involutions and DNA Encodings,” University of Western Ontario, London, Ontario, Canada. Submitted.

[29] Nikola Stambuk, "On circular coding properties of gene and proteinsequences," *Croatica Chemica ACTA*, vol.72, no.4, pp.999–1008, 1999.

[30] Nikola Stambuk, "On the genetic origin of complementary proteincoding," *Croatica Chemica ACTA*, vol.71, no.3, pp.573–589, 1998.

[31] Elwyn R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill Book Company, New York, NY, 1968.

[32] Shu Lin, Daniel J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Inc., 1983.

[33] Ajay Dholakia, *Introduction to Convolutional Codes with Applications*, Kluwer Academic Publishers, 1994.

[34] Donald L. Bitzer and Mladen A. Vouk, "A Table-Driven(Feedback) Decoder", *Proceedings of the Tenth Annual International Phoenix Conference on Computers and Communications*, 1991, Page(s): 385-392.

[35] Dholakia A., Lee T.M., Bitzer D.L., Vouk M.A., Wang L., and Franzon P.D., "An Efficient Table-Driven Decoder for One-Half Rate Convolutional Codes", *Proc. 30th ACM Annual Southeast Conference*, pp. 116-123, 1992.

[36] Dholakia A., Vouk M.A., and Bitzer D.L., "Table based decoding of rate one-half convolutional codes," IEEE Trans. on Communications, Vol. 43(2-4), pp. 681-686, 1995.

[37] Bitzer D.L., A. Dholakia, H. Koorapaty, and M.A. Vouk, "On Locally Invertible Rate-1/n Convolutional Encoders," *IEEE Transactions on Information Theory*, Vol 44 (1), pp. 420-422, January 1998.

[38] G.D.Forney, Jr., *Concatenated Codes*, M.I.T. Press, Cambridge, MA 1966.

[39] A.J.Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, IT-13, pp. 260-269, April 1967.

[40] G.D.Forney, Jr., “The Viterbi Algorithms,” *Proc IEEE*, 61, pp. 268-278, March 1973.

[41] J.M. Wozencraft and B. Reiffen, *Sequential Decoding*, MIT Press, Cambridge, Mass., 1961.

[42] J.L. Massey, *Threshold Decoding*, MIT Press, Cambridge, Mass., 1963.

[43] G. Ungerboeck, “Trellis-Coded Modulation with Redundant Signal Sets Parts I and II,” IEEE Communications Magazine, Vol. 25, No. 2, pp. 5-21, February 1987.

[44] David I. Rosnick, *Free energy periodicity and memory model for genetic coding*, PhD thesis, North Carolina State University, Raleigh, NC, 2001.

[45] Elebeoba E. May, *Analysis of Coding Theory Based Models for Initiating Protein Translation in Prokaryotic Organisms*, PhD thesis, North Carolina State University, Raleigh, NC, 2002.

[46] The Complete Genome Sequence of *Escherichia coli* K-12, Science. 1997 Sep 5;277(5331):1453-74.

[47] The complete genome sequence of *Escherichia coli* K-12, available online at $ftp : //ftp.ncbi.nih.gov/genbank/genomes/Bacteria/Escherichia\_coli\_K12/U00096.gbk$.