SAND REPORT

SAND2003-3963
Unlimited Release
Printed October 2003

Detection and Reconstruction of Error Control Codes for Engineered and Biological Regulatory Systems

Elebeoba E. May, Anna M. Johnston, William E. Hart, Jean-Paul Watson, Richard J. Pryor, and Mark D. Rintoul

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-3963
Unlimited Release
Printed October 2003

Detection and Reconstruction of Error Control Codes for Engineered and Biological Regulatory Systems

Elebeoba E. May and Mark D. Rintoul
Computational Biology

Anna M. Johnston, William E. Hart, and Jean-Paul Watson
Discrete Algorithms and Math

Richard J. Pryor
Evolutionary Computing

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0310
eemay@sandia.gov

Abstract

A fundamental challenge for all communication systems, engineered or living, is the problem of achieving efficient, secure, and error-free communication over noisy channels. Information theoretic principals have been used to develop effective coding theory algorithms to successfully transmit information in engineering systems. Living systems also successfully transmit biological information through genetic processes such as replication, transcription, and translation, where the genome of an organism is the contents of the transmission.

Decoding of received bit streams is fairly straightforward when the channel encoding algorithms are efficient and known. If the encoding scheme is unknown or part of the data is missing or intercepted, how would one design a viable decoder for the received transmission? For such systems blind reconstruction of the encoding/decoding system would be a vital step in recovering the original message. Communication engineers may not frequently encounter this situation, but for computational biologists and biotechnologist this is an immediate challenge.

The goal of this work is to develop methods for detecting and reconstructing the encoder/decoder system for engineered and biological data. Building on Sandia's strengths in discrete mathematics, algorithms, and communication theory, we use linear programming and will use evolutionary computing techniques to construct efficient algorithms for modeling the coding system for

minimally errored engineered data stream and genomic regulatory DNA and RNA sequences. The objective for the initial phase of this project is to construct solid parallels between biological literature and fundamental elements of communication theory. In this light, the milestones for FY2003 were focused on defining genetic channel characteristics and providing an initial approximation for key parameters, including coding rate, memory length, and minimum distance values. A secondary objective addressed the question of determining similar parameters for a received, noisy, error-control encoded data set. In addition to these goals, we initiated exploration of algorithmic approaches to determine if a data set could be approximated with an error-control code and performed initial investigations into optimization based methodolgies for extracting the encoding algorithm given the coding rate of an encoded noise-free and noisy data stream.

Acknowledgement

This work was supported by the Seniors Council Tier 1 Laboratory Directed Research and Development program. We would like to acknowledge our collaborators Mladen A. Vouk, Donald L. Bitzer, and Winser E. Alexander of North Carolina State University and thank John Emerson and Lyndon Pierson for bringing our idea to the attention of the Seniors Council. E. May would like to acknowledge John W. Drake of the National Institute of Environmental Health Sciences for providing additional insight on mutation rates and mutagenesis studies.

Contents

Summary **8**

1 Introduction **11**
1.1 EC Coding Methods for Genomic Sequence and System Analysis 11
1.2 Overview of Coding Theory 12
1.3 The Need for EC Coding in Living Systems 14
1.4 Biological Communication System Frameworks 15
1.5 Reverse Engineering the EC Code 16

2 Information Theoretic Studies **17**
2.1 Mutation and Replication Channel Capacity 17
2.2 Channel Capacity and Pathogenicity 20
2.3 Entropic Methods for Determining k 25

3 Cryptographic Analysis of RNA Data Streams **28**
3.1 The simple tests 28
3.2 Finding Linear Generators 29

4 Reverse Engineering EC Encoders, An Optimization Framework **31**
4.1 Linear Block Codes and Generator Matrices 31
4.2 An Integer Programming Approach to Solving the Reverse-Engineering Problem 33

5 Conclusion **37**

References **39**

Figures

1 Communication system that incorporates coding 13
2 Central Dogma of Genetics 15
3 Central Dogma of Genetics as a Coding System 16
4 Comparison of microbial genome mutation rate to genome size 18
5 Comparison of eukaryotic genome mutation rate to genome size 19
6 Capacity of prokaryotic replication channels 21
7 Capacity of eukaryotic replication channels 21
8 Comparison of prokaryotic and eukaryotic replication channel capacities 22
9 Micorbial genome mutation rates and their BSL classification level 24
10 Shannon entropy ratios for (7,4) Hamming Code 26
11 Shannon entropy profile for (7,4) Hamming Code 26
12 Shannon entropy ratios for (16,11) Hamming Code 27
13 Shannon entropy profile for (16,11) Hamming Code 27

14 An IP formulation to maximize the subset of $C_{n,k}$ for which a feasible dual code H exists given a set of error vectors. 35
15 An IP formulation to maximize the subset of $C_{n,k}$ for which a feasible dual code H exists assuming no errors. 36

Tables

1 Channel transition probability assuming p(Transition Mutation)=p(Transversion Mutation) 20
2 Channel transition probability assuming p(Transition Mutation) $\neq$p(Transversion Mutation) 20
3 Human pathogens classified by Biological Safety Levels 24
4 Maximal substring length for translated sequence set. 28
5 Maximal substring length for non-translated sequence set. 29
6 Maximal substring length for ilvE.dat – *E. coli* gene sequence. 30
7 Optimization results for $C_{7,4}$ codebooks under variable noise. '$\sum$ delta' denotes the number of codewords for which the resulting H matrix is feasible. 34

Summary

The initial phase of this work employed a three prong approach to address the problem of reverse engineering error control (EC) encoded data. Approaches included: 1) Information theoretic studies of the genetic channel and EC encoded data streams, 2) Cryptographic exploration of RNA data streams, and 3) Investigation of the reverse engineering problem from an optimization framework.

In engineering systems, channel characteristics determine the EC coding used. We investigate mutation rates for the replication process (modeled as an error introducing communication channel) of various organisms. From analysis of mutagenesis data, we note: 1) The relationship between prokaryotic mutation rates and genome size exhibits power law behavior. This does not hold for higher eukaryotes. 2) A link may exist between the mutation rate of a biological agent and the agent's pathogenicity. Initial findings show that the Biological Safety Level (BSL)-1 category contained the agent with the lowest error rate while BSL-3 contained the agent with the noisiest genetic channel. 3) Based on mutation rates we calculated the genetic channel capacity. Although there is very little difference among the organisms studied, the channel capacity of higher eukaryotes tends to be slightly larger than that of the DNA micorbes. Overall initial channel capacity calculations imply a very high coding rate, one with minimal redundancy possibly of the form $(n = N, k = N - 1)$. To determine EC coding parameters, we developed a method for determining k for an $(n = N, k)$ linear block code. The (7,4) Hamming, (16,11) Hamming, and (32,17) codebooks were analyzed using a variation of the Shannon entropy. Codebook codewords contained randomly generated $T = 0..5$ error bits. Entropic profiles asymptotically approached correct k values even in the presence of noise.

The goal of our cryptographic study was to search RNA streams for mathematical relationships or exploitable patterns. These relationships and patterns, if they exist, will improve our understanding of how biological sequences store, process and handle genomic data. Initial work was performed on *Escherichia coli* genes and leader sequences. Analysis methods included:

1. Lexicographical sorting to find matching sub-streams and obtain statistics on matching substring lengths;
2. One and two element Markov models to determine if short common patterns exist;
3. Finding linear generators for RNA sequences mapped to GF(2^2) to determine the existence of hidden linear relationships;
4. Analyzing the effect of various mappings of nucleotide bases to the elements in GF(2^2), to determine if there are mathematical reasons to choose one mapping over another.

Lexicographical sorting found that leader sequences have larger matching sub-streams on average than the full *E. coli* gene sequence. The mean maximal matching substring length was 7.35 for non-translated leader sequences (intergenic sequences), 6.89 for the translated leader sequences,

and 4.32 for the complete gene sequence. Currently, no obvious patterns were found using Markov modeling. Additional studies are needed. The final two tests were intimately connected. The research showed that the linearity of a stream depends on how the bases are mapped. In particular, the overall linearity of the stream (a ratio of the number of elements generated by a polynomial over the degree of its generating polynomial) depends on which base is mapped to the zero element. Mapping cytosine to the zero element (the other bases can be mapped to any of the remaining three elements) gave the highest linearity ratios.

The problem of reconstructing an encoder/decoder system can be viewed as an optimization problem. We have formulated the problem as an integer program (IP), which can be solved exactly using available branch-and-bound technology. At present, these algorithms effectively reconstruct encoders/decoders for error-free channels. However, scalability is a major issue, as we are currently unable to solve the reconstruction problem for large, noisy channels. There are two issues with scalability. The first is the strength of the lower bound, which in the current formulation appears quite weak. This is causing a huge branch-and-bound tree, such that nodes can rarely be pruned. The second is the memory consumption of the IP formulation (related to the number of nodes), which scales as the product of codeword length, n, and the number of codewords m. To achieve scalability for realistically sized biological systems, different problem formulations and advances in solver technology are required.

Detection and Reconstruction of Error Control Codes for Engineered and Biological Regulatory Systems

1 Introduction

Years of biological experiments have produced descriptions of what occurs during the genetic replication, transcription, and translation processes. In translation for instance, molecular biologists have identified key regions upstream and downstream of the initiation codon that affect the ribosome's ability to initiate translation and the rate at which translation initiation occurs. The specific effects of base composition and distance of key bases from the initiation site is not completely understood and has not been mathematically quantified. If we were able to construct a mathematical model to describe the regulatory regions on messenger RNA (mRNA) which control ribosomal attachment and the rate of translation initiation, we could reconstruct optimal translation initiation sites. These optimal sites can be used in transgenic protein production (using an organism to produce proteins foreign to that organism's genome), increasing the expression of biosensor reporter proteins, and regulating the expression of proteins useful for bioremediation in microbes of interest to the Department of Energy (DOE).

Compiling a set of optimal regulatory sequences would prove experimentally intractable. If we limited our search of viable translation regulatory sites to sixty base sequences, we would examine at least 58^4 sequences (assuming only the first base of the initiation codon is variable). Experimental evaluation of such a large number of sequences is not a viable option for biologists or biotechnologists. Developing a mathematical framework that correlates base composition and base location in regulatory sequences with corresponding genetic regulatory response will provide a mathematically detailed understanding of genetic regulation, produce a tool for optimizing sequences involved in genetic regulatory control, and contribute to the understanding of genetic networks - a key aspect of DOE's Genomes to Life program. The understanding gained from this work will also benefit several Sandia National Laboratories (SNL) research endeavors, including: development of biological agents for bio-weapons defense and development of biological substrates for bio/nano-technology systems.

1.1 EC Coding Methods for Genomic Sequence and System Analysis

Molecular biology has provided significant insights into the mechanisms of translation initiation. Although a general consensus mRNA leader sequence can be formulated based on experimental data [26], we still lack a mathematical model that correlates specific mRNA sequence with a specific rate of translation initiation. To this end, we will view the mRNA leader region (nucleotides from 30 to +30 inclusive) as points or codewords in a high dimensional space, where each point has an

associated translation efficiency. The hypothesis of current work is that nucleotide variations in the ribosome binding site region can be quantified using an EC coding framework and the effects of these variations on translation initiation can be determined and predicted using such a framework. Though the idea of biological coding spheres and biological coding theory are not new [31, 24, 25, 2] a rigorous development of biological codes for quantification and optimization of regulatory sites is novel.

Application of coding theory to genetic data dates back to the late 1950s [11, 10] with the deciphering of the genetic code. Since then, EC coding methods have been applied to genetic sequence analysis and classification, biological chip design, as well as analysis of genetic regulatory processes. Sengupta and Tompa approach the problem of oligo array design from a combinatorial design framework and use EC coding methods to increase the fidelity of oligo array construction [27]. Reif and LaBean propose EC coding-based methods for the development of error-correction strands for repairing errors in DNA chips [21].

Several researchers have moved beyond the qualitative models of biological communication and attempted to determine the existence of EC codes for genomic sequences [31, 17, 23, 13, 16]. Liebovitch et al. and Rosen and Moore [13, 23] both developed techniques to determine the existence of EC code for genomic sequence. Neither found evidence of EC codes for the sequences tested. Given the computational limitations of the study, Liebovitch et al. suggest that a more comprehensive examination would be required. Both methods investigate a subset of linear block codes and do not consider convolutional coding properties nor account for the inherent noise in genomic sequences. Extending beyond specific genomic regions and sequences, MacDonaill develops an EC coding model for nucleic acid sequences in general [16]. He has proposed a four-bit, binary parity check EC code for genetic sequences based on chemical properties of the nucleotide bases. As more researchers explore the EC coding properties of genetic sequences and apply these methods to computational biology and molecular computing problems, the information and coding theoretic properties of genetic systems can be further understood and potentially exploited for bioengineering applications.

In the remainder of this section we provide a basic introduction to coding theory and discuss parallels between coding theory and genetic processes. The next three sections describe initial approaches explored in this work. Section 2 presents information theoretic studies of microbial and eukaryotic replication and EC block codes. Section 3 describes cryptographic analysis of mRNA leader sequences and *Escherichia coli* gene sequences and Section 4 analyzes inverse EC coding from an optimization framework. The final section of this report summarizes our findings and discusses future work.

1.2 Overview of Coding Theory

The need for coding theory and its techniques stems from the need for error control mechanisms in a communication system. The system in Figure 1 illustrates how coding is incorporated into a

typical communication system [29]. In an engineering communication system, digitized informa-

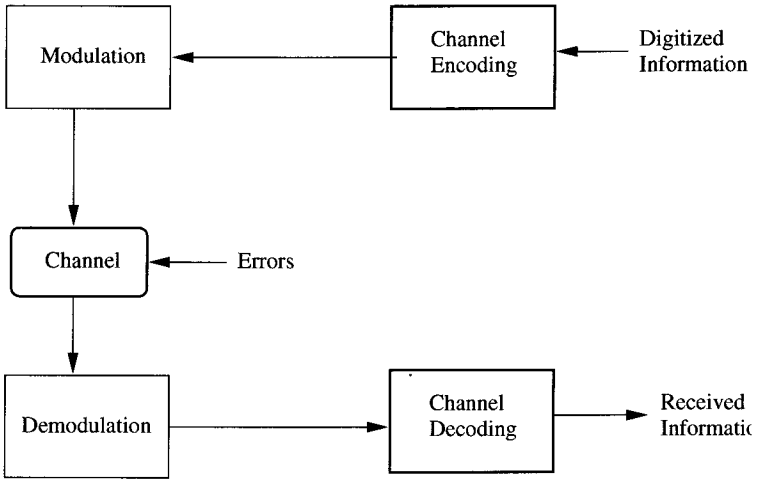

Figure 1. Communication system that incorporates coding .

tion is encoded by the channel (error control) encoder and prepared for transmission (modulation). The encoded stream is transmitted through a potentially noisy channel where the sequence can be corrupted in a random fashion. The output of the channel, the received message, is prepared for decoding (demodulation) and then decoded by the channel (error control) decoder [29, 5]. The decoding process involves removal and possibly correction of errors introduced during transmission. The decoding mechanism can only cope with errors that do not exceed the code's error correction capability.

The channel encoder processes the digitized information frame by frame. An input frame consists of a fixed number, k, of information symbols that are presented to the encoder. The output frame, the frame to be transmitted, consists of n (also fixed) output symbols, where n is larger than k. Since the number of output symbols is greater than the number of input symbols, redundancy has been introduced [29]. The coding rate,

$$R = k/n \tag{1}$$

is the the ratio of the number of input symbols in a frame to the number of output symbols in a frame. The lower the coding rate, the greater the degree of redundancy [29, 15]. The encoder combines the input symbols and introduces additional symbols based on a deterministic algorithm. This results in a mapping of input frames into a set of output frames known as codewords. The type of output produced is determined by the number of input frames used in the encoding process. Block coding uses only the current input frame. Convolutional coding uses the current frame plus m previous input frames [29, 5].

The communication channel is the medium through which the information is transmitted to the receiver. The channel can corrupt the transmitted message through attenuation, distortion, interference, and addition of noise. Channels can be characterized as memoryless, symmetric, additive

white Gaussian noise (AWGN), bursty, or as compound channels. Channel characteristics determine the type of EC encoding method used in the engineering system [29].

The channel decoder receives a series of frames that, given an errorless transmitted sequence, should be composed only of codewords. If the received sequence has been corrupted during transmission, there will be sequences which do not map uniquely to any codewords. This is used to detect the presence of errors. Decoding algorithms are then used to determine the original codeword and correct the error. When the error rate exceeds the error correction capacity of the code, two things can occur. The decoder may be able to detect the error but may not be able to find a unique solution and thus correct the error or, the decoder may not detect the error because the corruption has mapped one legal codeword into another legal codeword. The method of decoding is dependent on the method of encoding.

The decoding of received bit streams is fairly straightforward when the channel encoding algorithms are efficient and known. What if the encoding scheme is unknown or part of the data is missing? How would one design a viable decoder for the received transmission? Communication engineers may not frequently encounter this situation, but for computational biology this is the immediate challenge and barrier to understanding the vast amount of sequence data produced by genome sequencing projects. To determine the algorithm used by living systems to transmit vital genetic information, several researchers have explored the parallel between the flow of genetic information in biological systems and the flow of information in engineering communication systems [9, 31, 22, 2, 17].

1.3 The Need for EC Coding in Living Systems

Battail [2] argues, similar to Eigen [8], that for Dawkins' model of evolution to be tractable, error-correction coding must be present in the genetic replication process. According to Battail, proof-reading, a result of the error avoidance mechanism suggested by genome replication literature, does not correct errors present in the original genetic message. Only a genetic error correction mechanism can guarantee reliable message regeneration in the presence of errors or mutations due to thermal noise, radioactivity, and cosmic rays [2].

Battail further asserts that the need for error protection becomes obvious when one considers that the number of errors in a k-symbol message that has been replicated r times is comparable to the number of errors in an un-replicated $r * k$-symbol message. For a given error rate, the number of times an organism undergoes replication approaches an infinite number. Hence for a message to remain reliable within an organism's life cycle (not to mention evolutionary information transmission which occurs over thousands of years) the message must have strong error protection [2]. The survival of an organism necessitates the existence of a reliable information replication process. Therefore error-correcting codes must be used in replication or in another process of information regeneration that precedes replication [2].

1.4 Biological Communication System Frameworks

The relationship between the error control coding process and protein translation may not be obvious. Figure 2 illustrates the central dogma of genetics. The central premise of genetics is that

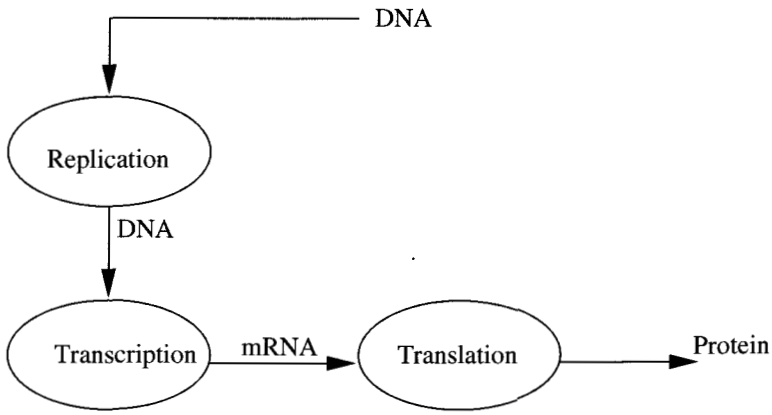

Figure 2. Central Dogma of Genetics

genes are perpetuated in the form of nucleic acid sequences but function once expressed as proteins [12]. Three-base nucleic acid sequences, called codons, designate amino acids. There are sixty-four possible codons and twenty amino acids. Hence different codons can specify the same amino acid. This codon/amino acid designation is known as the genetic code [30]. There are three processes which transform genes from nucleic acid sequences to functional proteins.

- Stage 1: Replication - A DNA sequence replicates to form two identical DNA sequence
- Stage 2: Transcription - Using one of the DNA strands as a template sequence, the information contained in the DNA sequence is transcribed to its RNA equivalence. The result is a messenger RNA (mRNA) sequence which contains the complement sequence of the DNA template strand. The difference is that in mRNA, Uracil replaces Thymine bases [30].
- Stage 3: Translation - The mRNA serves as a template for producing polypeptide chains or proteins. A polypeptide chain is a sequence of amino acids bound together by peptide bonds [12]. The ribosome is an important part of the mechanism which translates mRNA information into proteins.

Researchers, such as Hubert Yockey who performed fundamental investigations of error correcting coding properties of genetic systems, have explored the EC coding properties of genetic sequences and systems [31, 17, 23, 13, 16]. Several researchers have developed communication models for genetic processes [9, 31, 22, 2, 19]. Our analogy of genetic information transmission to an engineering communication system is illustrated in Figure 3. The un-replicated DNA sequence

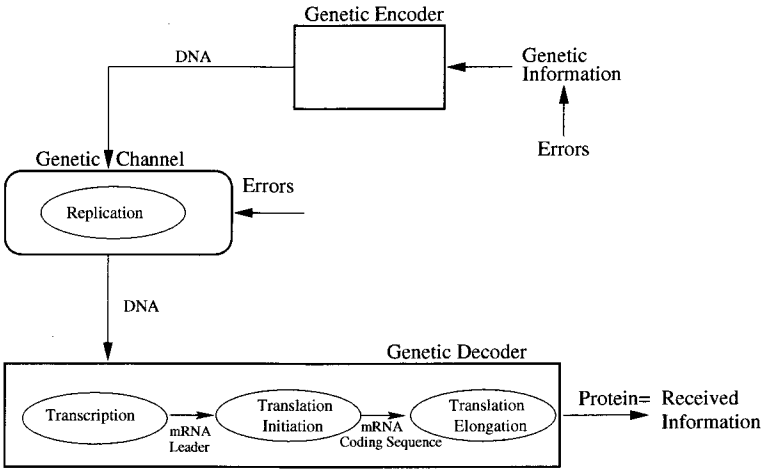

Figure 3. Central Dogma of Genetics as a Coding System

is the output of an EC genetic encoder that adds redundancy to inherently noisy genetic information. The noise in the source can be thought of as mutations transferred from parent to offspring. The genetic channel is the DNA replication process during which errors are introduced into the nucleotide sequence [19]. Incorporating the nested coding idea proposed by Battail [2], EC decoding occurs in three phases represented by transcription, translation initiation, and translation elongation plus termination.

1.5 Reverse Engineering the EC Code

Coding theory algorithms can serve as powerful pattern recognizers for annotating biologically active sites of a genome, and also as pattern generators that can mathematically represent the genetic process and macromolecules that operate on the genomic sequence of interest. The mathematical representation of a convolutional code is also the mathematical model for the digital system that produces that signal (or pattern) and all other signals associated with that system.

Development of coding theoretic frameworks for molecular biology is an ongoing endeavor. Although the existence of redundancy in genetic sequences is accepted and the possibility of that redundancy for error correction and control is being explored and exploited, mathematically determining the encoding algorithm particularly for regulatory regions remains a major research challenge. To this end we propose to determine the genetic encoder/decoder by reconstructing the encoder from the mRNA sequence which we model as a noisy received EC encoded sequence.

Development of blind reconstruction methods can be useful in data transmission systems where the encoding algorithm is unknown. When a message is received, the redundancy from the error-control encoder must be algorithmically removed prior to further processing of the message. If the

EC coding information is missing then the receiver must use the possibly noisy data to "guess" at the underlying encoding system. We are faced with a parallel scenario when analyzing genomic data. The information produced by genome projects hold the key to understanding how an organism functions from genetic to cellular level behavior. Identifying gene locations and regulatory regions are fundamental steps in the "genome to life" process. It is not feasible to experimentally annotate all of an organisms regulatory regions hence the need for computational tools for accurately deciphering the information contained in genetic sequences. The majority of gene annotation techniques rely on patterns and statistical characteristics of the genome for model construction. While these methods yield viable results, they do not offer insight into the underlying mechanics of the genetic process. By devising a method for reconstructing the EC code of a received, noisy, signal we will provide a way to:

1. Determine the encoder/decoder model for engineered systems where the encoding algorithm is unknown. Addressing the problem for the engineering system provides a baseline for developing and testing computational models for biological systems.
2. Construct mathematical models of molecular machines (macromolecules such as ribosome, RNA polymerase, and initiation factors) involved in the regulation of genetic processes.

During the initial phase of this project we use information theory, cryptography, and optimization techniques to investigate methods for reconstructing the EC code of engineered and genetic data.

2 Information Theoretic Studies

The genetic communication system depicted in Figure 3 represents the error introducing transmission channel as the replication process. Shannon's channel coding theorem asserts that there exists a channel code with rate $R = k/n$ such that the probability of decoding error becomes arbitrarily small as n increases [4, 28, 1]. The capacity of a transmission channel (the maximum data transmission rate) is dependent on the error rate of the channel $p_{i,j}$, the probability of the channel transforming symbol i into symbol j for $i \neq j$. In order to determine appropriate EC coding parameters for genetic regulatory sequences, we must characterize the replication channel and the error or mutation rates associated with replication. Mutation derived capacity values can suggest R and from that plausible n and k values for genetic systems. In addition to a mutation based approach we explore a Shannon entropy-based approach to determine k for an $(n = N, k)$ code.

2.1 Mutation and Replication Channel Capacity

Mutations are replication errors that remain or are missed by genetic proofreading mechanisms. Drake et al. [7, 6, 3] have performed extensive research and analysis of mutation rates in prokaryotic and eukaryotic organisms. Based on mutagenesis studies, they note that mutation rate in RNA

viruses range from one per genome per replication for lytic viruses to 0.1 per geneome per replication for retroviruses and retrotransposons. DNA microbes, more complex and typically larger than RNA viruses, have mutation rates of $\frac{1}{300}$ per genome per replication. Moving higher still to the larger more complex eukaryotic organism, higher eukaryotes have mutation rates ranging from 0.1 to 100 per geneome per sexual generation and a mutation rate of $\frac{1}{300}$ per cell division per effective genome. The effective genome is the portion of the genome where mutations are most lethal (i.e. genes or exons) [7]. In general, while RNA viruses have significantly higher mutation or channel error rates, DNA microbes have error rates relatively similar to the mutation rate in the effective genome of higher eukaryotes. The question arises whether and how organism complexity (which we can loosely approximate using genome size) is related to replication channel fidelity. Drake investigates this for DNA microbes by analyzing the log-log plot of base mutation rates as a function of genome size [6]. We replicate this test using the base mutation and genome size data from Drake et al. [7] for both the DNA microbes and the higher eukaryotes. Figure 4 and Figure 5 show the log-log plots of genome size as a function of base mutation for DNA microbes and eukaryotic organisms, respectively. The log-log plots for the DNA microbes are equivalent to Drake et al.'s

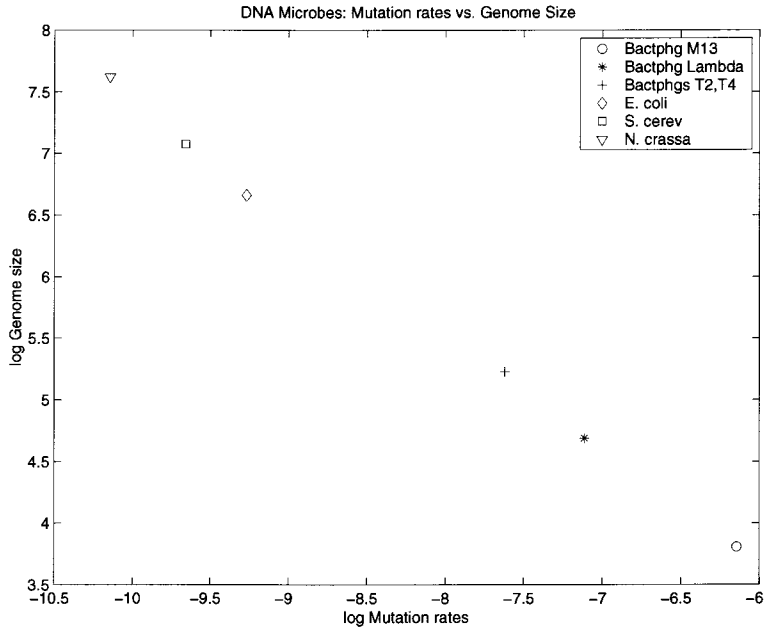

Figure 4. Comparison of microbial genome mutation rate to genome size

results as would be expected. The relationship between the DNA microbes' mutation rates and genome size exhibits power law behavior. We do not see a similar behavior for higher eukaryotes although the eukaryotic data set contained a relativly small number of organisms. As concluded by Drake et al. and illustrated in Figure 4 , there is an inverse relationship between genome size, G, and an organism's base mutation rate, μ_b. This inverse relationship is evident for the higher eukaryotes

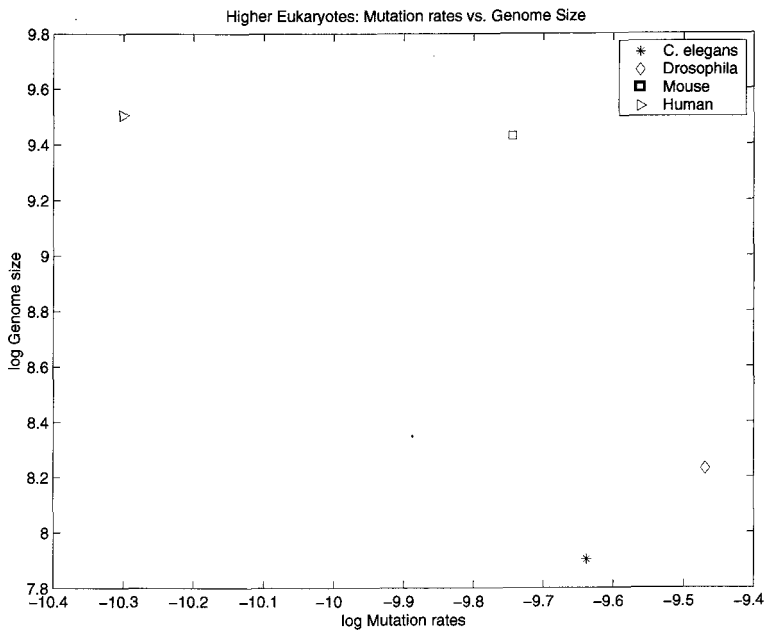

Figure 5. Comparison of eukaryotic genome mutation rate to genome size

as well.

Based on mutation rates in Drake et al. [7] we calculate the genetic channel capacity. Assuming a discrete memoryless channel (DMC), the capacity of the channel, C, is the maximum reduction in uncertainty of the input X given knowledge of Y [4]:

$$C = \sup_{X} I(X,Y) \tag{2}$$

where

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3}$$

The Shannon entropy $H(X)$ and $H(Y|X)$ are defined as:

$$H(X) = -\sum_{i} p(x_i) log_2 p(x_i) \tag{4}$$

$$H(Y|X) = -\sum_{k}\sum_{j} p(x_k, y_j) log_2 p(y_j|x_k) \tag{5}$$

The probability $p(y_j|x_k)$ is the channel error probability. If $p(y|x)$ is specified by the mutation error rate μ_b then $p(y_j|x_k) = \mu_b$, $\forall y \neq x$ and $p(y_j|x_k) = 1 - \mu_b$, $\forall y = x$ (where μ_b is the mutation rate per base per replication). We assume two different channel transition matrices. For the first case, Table 1, we assume all base mutations are equal, hence a transition mutation (purine to

purine, $Adenine(A) \leftrightarrow Guanine(G)$ and pyrimidine to pyrimidine, $Cytosine(C) \leftrightarrow Thymine(T)$) and a transversion mutation (purine to pyrimidine, $(A,G) \rightarrow (C,T)$ and pyrimidine to purine, $(C,T) \rightarrow (A,G)$) are equally probable. The second case, Table 2, we assume that transition mutations are

Table 1. Channel transition probability assuming p(Transition Mutation)=p(Transversion Mutation)

	A	G	C	T
A	$1-\mu_b$	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$
G	$\frac{\mu_b}{3}$	$1-\mu_b$	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$
C	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$	$1-\mu_b$	$\frac{\mu_b}{3}$
T	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$	$\frac{\mu_b}{3}$	$1-\mu_b$

twice as probable as transversion mutations. Figure 6 and Figure 7 show the replication channel

Table 2. Channel transition probability assuming p(Transition Mutation) $\neq$p(Transversion Mutation)

	A	G	C	T
A	$1-\mu_b$	$\frac{2\mu_b}{3}$	$\frac{\mu_b}{6}$	$\frac{\mu_b}{6}$
G	$\frac{2\mu_b}{3}$	$1-\mu_b$	$\frac{\mu_b}{6}$	$\frac{\mu_b}{6}$
C	$\frac{\mu_b}{6}$	$\frac{\mu_b}{6}$	$1-\mu_b$	$\frac{2\mu_b}{3}$
T	$\frac{\mu_b}{6}$	$\frac{\mu_b}{6}$	$\frac{2\mu_b}{3}$	$1-\mu_b$

capacity of the organism as a function of the log of the organism's genome size for DNA microbes and higher eukaryotes using μ_b values from Drake et al. [7] and channel transition probabilities from Table 1. Figure 8 shows the channel capacity for DNA microbes and higher eukaryotes combined. There is very little difference among the organisms studied, the channel capacity of higher eukaryotes tends to be slightly larger than that of the DNA micorbes. The initial channel capacity calculations imply a very high coding rate, one with minimal redundancy of the form (n=N,k=N-1). Further calculations are necessary and the number of replication cycles need to be taken into consideration since mutation errors are cumulative and the channel model should reflect this. Calculations using Table 2 channel transition probabilities yield similar results.

2.2 Channel Capacity and Pathogenicity

Since lower error rates indicate a higher channel capacity, Figure 8 suggests that in general, increased organism complexity implies increased transmission fidelity. One could extrapolate further and suggest that this implies that the need for error control is reduced as complexity increases. On

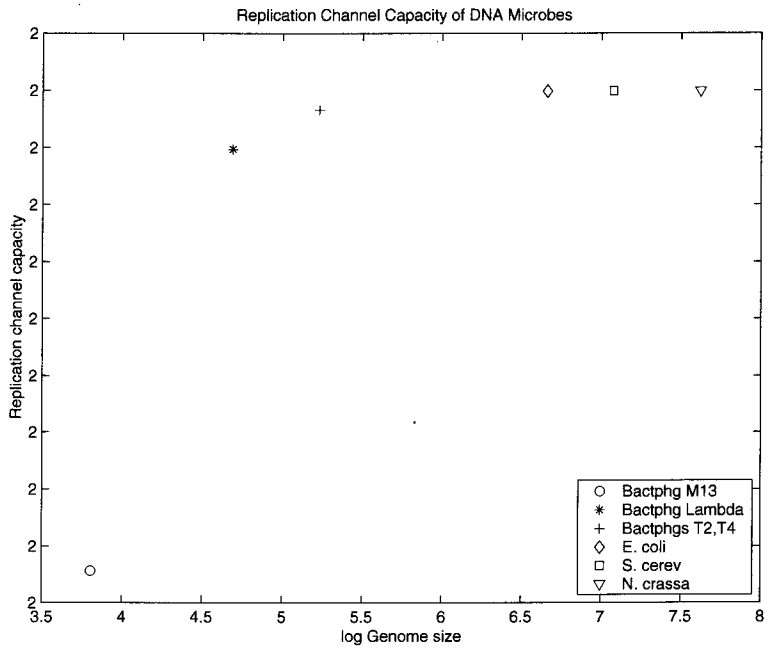

Figure 6. Capacity of prokaryotic replication channels

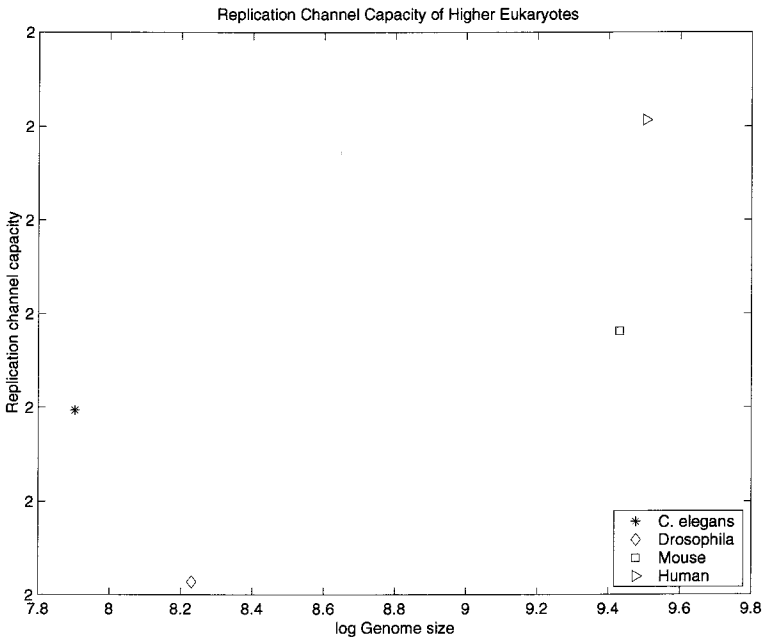

Figure 7. Capacity of eukaryotic replication channels

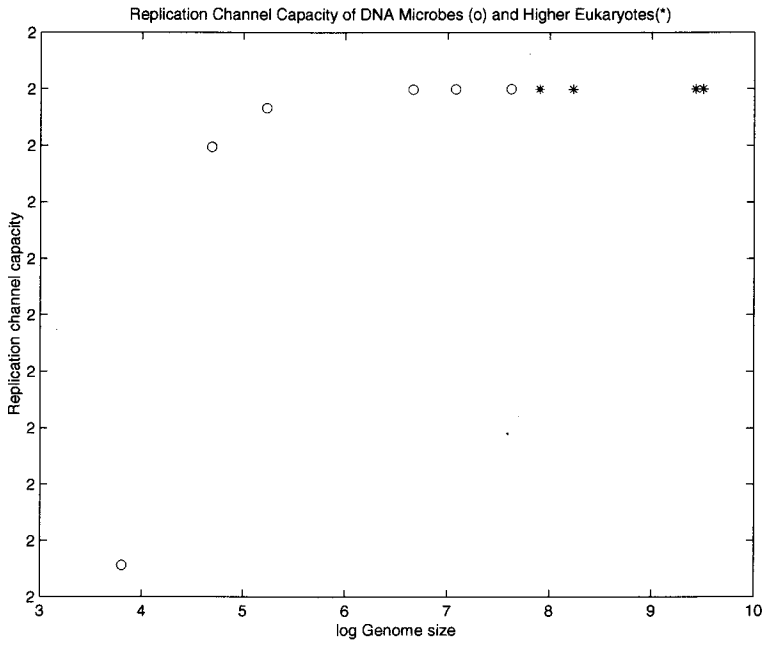

Figure 8. Comparison of prokaryotic and eukaryotic replication channel capacities

the contrary, we assert that the higher fidelity is due to the incorporation of redundancy for error control purposes. Therefore less complex organisms without sufficient error control encoded into their genomes (resulting in a smaller genome size) must explicitly incorporate a redundancy method in order to survive. The large number of virons present in an infected cell or phages/plasmids present in microbes can be viewed as the less complex organism's method for explicitly incorporating error control. If we were transmitting over a noisy engineering channel and were unable to modify our message in order to incorporate a stronger error control algorithm, a simple way to increase fidelity is to transmit the message multiple times, thereby effectively incorporating error control into our system.

Another way to combat the problem of transmission over a noisy channel without modifying the message is, if the alternative exists, transmit over a channel with lower noise. It appears this is the route viruses, phages, and plasmids exploit when they insert into their host genome. RNA viruses have relatively low complexity and high mutation rates. Drake et al. note that the RNA virus/retrovirus populations are "likely to be extinguished when mutation rates are increased to a few fold over one [7]." Lytic RNA viruses have a mutation rate of 1/genome/replication. But retroviruses and retrotransposons have a mutation rate of 0.1/genome/replication, an order of magnitude difference. Retroviruses insert their reverse-transcribed chromosome into the chromosome of a different cell and retrotransposons insert their reverse-transcribed chromosome into the chromosome of the cell in which they reside. While the lytic RNA virus produces multiple copies of itself using a

noisy channel, retroviruses and retrotransposons elect to use the host's less noisy replication channel and therefore reduces the need for large copies or retransmissions of their genetic information.

Similar behavior is seen in the F plasmid and Bacteriophage λ infection of *E. coli*. As a prophage Bacteriophage λ and non-conjugating F plasmid both have mutation rates equivalent to their host's, but during lytic replication, Bacteriophage λ has a higher mutation rate. Likewise F plasmid's mutation rate is five to twenty times higher during conjugation [7]. This supports our assertion that, similar to RNA viruses, lower complexity organisms incorporate error control by alternate means in order to successfully transmit their genetic information.

Drake et al. suggest that a lytic virus' high mutation rate may have a strong link to its low infectivity. Beyond survival, we speculate that mutation rates may determine an organisms pathogenicity or a host's susceptibility to infection. Given the trend for more complex organisms to have less noisy transmission channels and lower complexity organisms (typically the pathogenic agents) tendency towards noisier transmission channels, we hypothesize that the more noisy the agent's replication channel in relation to the host's channel the more virulent the agent. Virulence is the agents degree of pathogenicity. There are various values used to determine an agents virulence:

LD50: The number of organisms/agents needed to kill fifty percent of the host organism.

ID50: The number of organisms/agents needed to cause infection in fifty percent of the host organism.

During this initial phase we were unable to find sufficient virulence information for various hosts to test our hypothesis but indirect virulence information for agents potentially harmful to humans was readily available. Potential human pathogens are classified using Biological Safety Levels (BSL) designations. There are four levels:

BSL-1: The agent is not associated with disease in healthy adult humans.

BSL-2: The agent is associated with a disease which is rarely serious and for which preventive measures or therapeutic interventions are often available.

BSL-3: The agent is associated with a serious or lethal disease for which preventive measures or therapeutic interventions may be available.

BSL-4: The agent is likely to cause serious or lethal human disease for which preventive measures or therapeutic interventions are not usually available.

Using mutation data from Drake et al. [7] and BSL classification data from the Center for Disease Control (CDC) website we looked at the genome mutation rates μ_g by BSL levels (Figure 9) for the organisms in Table 3.

Figure 9 suggests that a link may exist between the mutation rate of a biological agent and the agent's pathogenicity. The BSL-1 category contains, *E. coli* K-12, the agent with the lowest error

Table 3. Human pathogens classified by Biological Safety Levels

	Pathogens
BSL-1	*Escherichia coli* K-12
BSL-2	Murine leukemia virus (MLV), Bovine leukemia virus (BLV), Rous sarcoma virus (RSV), Polio virus, Influenza A
BSL-3	Human immunodeficiency virus type 1 (HIV-1), Vesicular stomatitis virus (VSV)

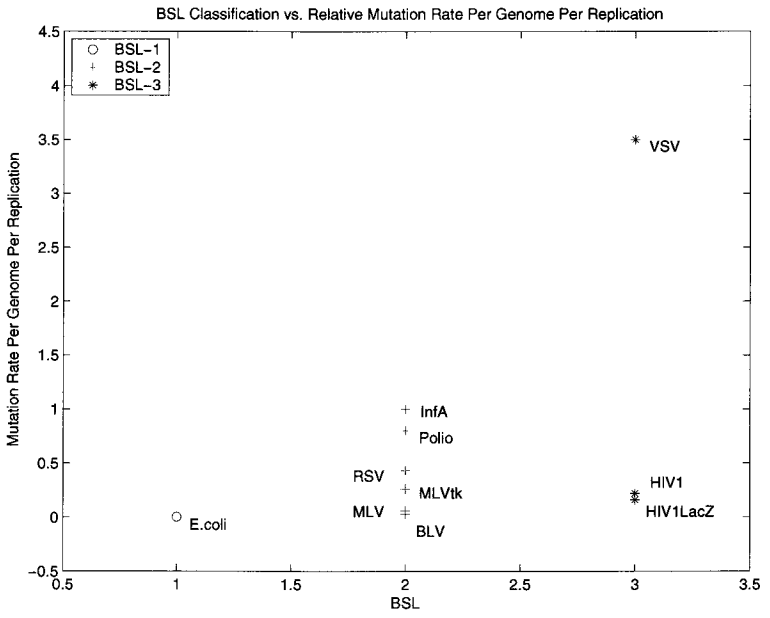

Figure 9. Micorbial genome mutation rates and their BSL classification level

rate while BSL-3 contains VSV, the agent with the noisiest genetic channel. Further investigation with larger data sets for various host/pathogen virulence data is needed in order to draw a more definitive conclusion. Virulence data with LD_{50} and ID_{50} values need to be compiled from literature and should provide better insight into the relationship between virulence and host/pathogen channel fidelity.

2.3 Entropic Methods for Determining k

To determine EC coding parameters, we developed a method for determining k for an $(n = N, k)$ linear block code. Given an (n,k) codebook, the amount of information contained in the codebook is k bits. We began by calculating the entropy of each of the i positions in the codewords, $H_i^{n,k}$, $for\ i = 1..n$. The Shannon entropy of the (n,k) codebook was then defined as

$$H^{n,k} = \sum_{i=i}^{n} H_i^{n,k} \tag{6}$$

Initial calculations yielded $H^{n,k} \approx n$ for each codebook set tested. We varied the entropy calculations to evaluate positional entropy for varying window size $w_k = 1..n$. The assumption is as $w_k \rightarrow k$ the average $H^{n,k} \rightarrow k$ where, for a fixed w_k, the average entropy is:

$$H_{avg}^{n,k} = \frac{1}{n - w_k + 1} \sum_{i=1}^{n-w_k+1} \sum_{j=i}^{i+w_k-1} H_j^{n,k} \tag{7}$$

We calculate $H_{avg}^{n,k}$ for the (7,4) Hamming, (16,11) Hamming, and a (32,17) linear block codebooks. Figure 10 and Figure 11 show the ratio $\frac{H_{avg}^{n,k}}{H_{max}^{n,k}}$ (where $H_{max}^{n,k} = w_k$) and $H_{avg}^{n,k}$ for all estimates of k for the (7,4) Hamming codebook. Figure 12 and Figure 13 show similar results for the (16,11) Hamming codebook. As illustrated in the results, the modified entropy calculations were also applied to (7,4) and (16,11) Hamming codebooks containing randomly generated $T = 0..5$ error bits. As $w_k \rightarrow k$ the average entropy profile, $H_{avg}^{n,k}$, asymptotically approaches k and the ratio $\frac{H_{avg}^{n,k}}{H_{max}^{n,k}}$ drops below one. For $w_k > k$ the average entropy value does not exceed the correct k value. Further investigation is necessary to determine whether the slope of the entropy ratio can provide any information regarding the amount of noise present in the codebook set. Performing similar tests for the (32,17) code is significantly more computationally expensive than for the (7,4) or (16,11) code, although we suspect similar behavior would occur. This approach is a promising method for determining k given an $(n = N, k)$ linear block code. The next step is to extend the current approach and develop a method to determine n for an (n,k) linear block code. It is also necessary to expand and apply related methods to the analysis of convolutional codebooks.

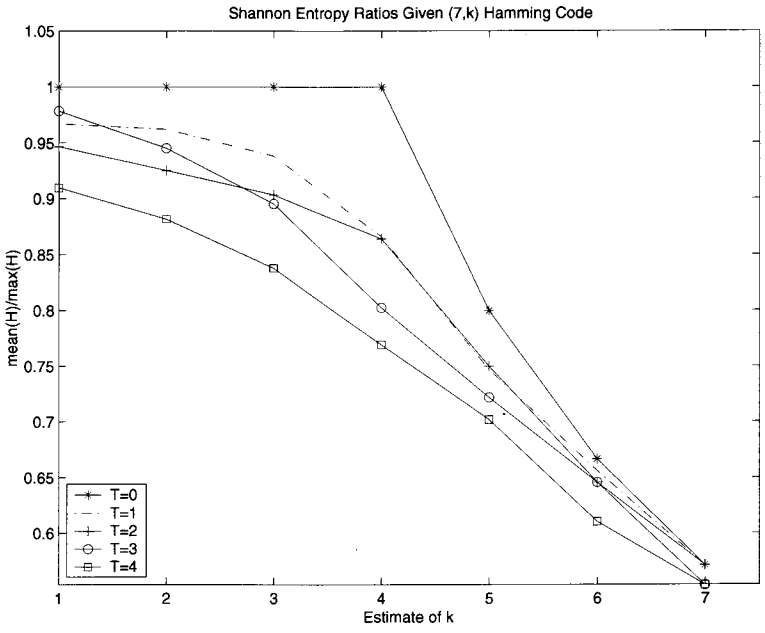

Figure 10. Shannon entropy ratios for (7,4) Hamming Code

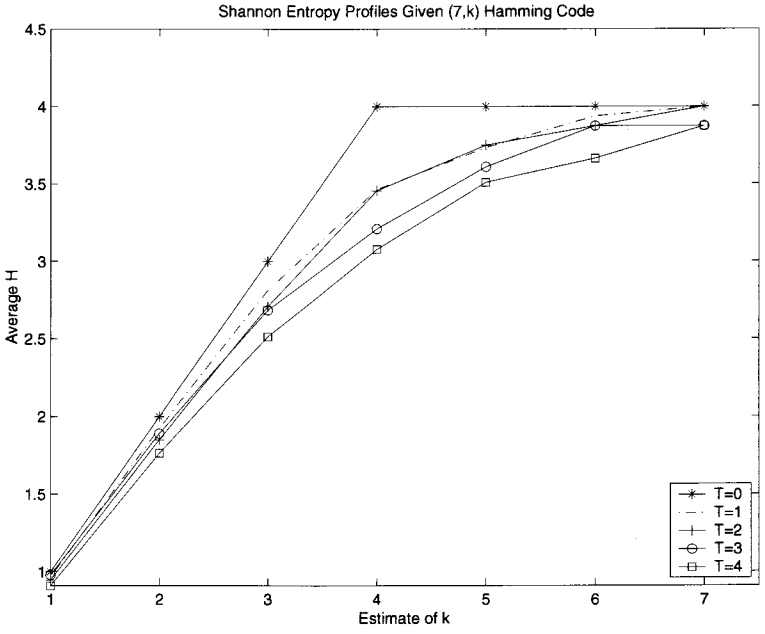

Figure 11. Shannon entropy profile for (7,4) Hamming Code

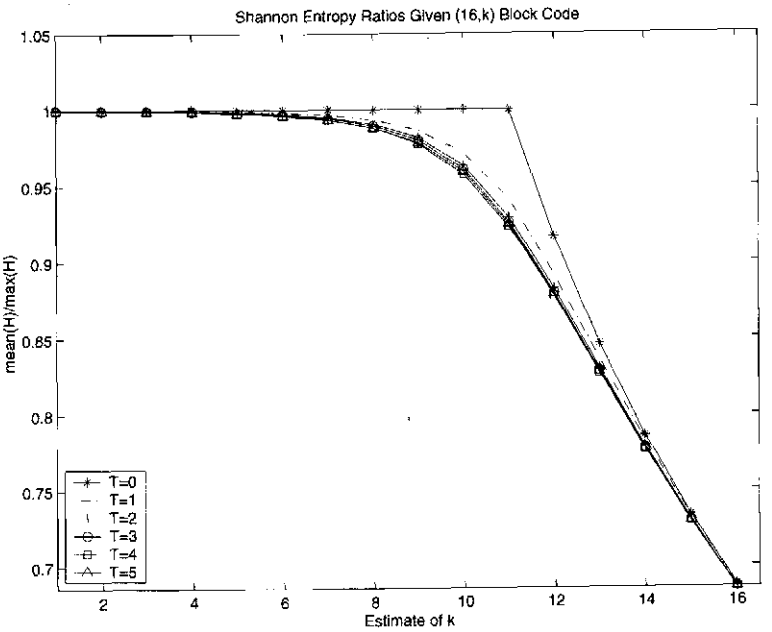

Figure 12. Shannon entropy ratios for (16,11) Hamming Code

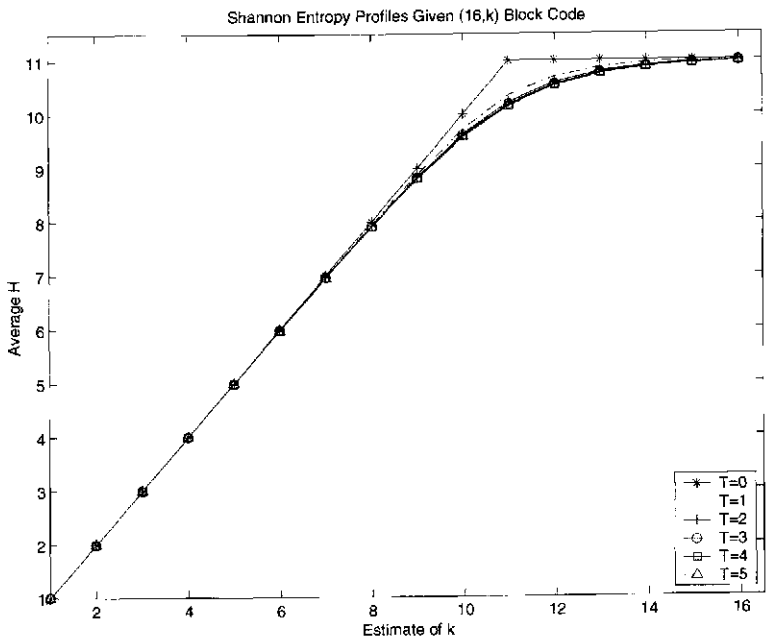

Figure 13. Shannon entropy profile for (16,11) Hamming Code

3 Cryptographic Analysis of RNA Data Streams

The goal of this section of research is to analyze RNA streams for embedded information theoretic relationships. We perform several initial tests on translated and non-translated initiation sequences (with thirty bases before and twenty seven bases after the initiation sequence, for a total of sixty-base strings) and on complete *E. coli* gene sequences. Simple tests, such as short Markov modeling, finding matching substreams, and positional counts are performed on files with multiple related streams. More complex tests, such as finding linearity measures for the streams and working with multiple mappings of n-offs of the streams are also performed.

3.1 The simple tests

The first analysis method used is lexicographical sorting of the strings to find maximal matching substrings. Long matches found in the same RNA stream may indicate simple repetitive error correction. Before sorting, the multiple initiation sequences are merged into two long sequences: one composed of translated (valid leaders) and the other of non-translated (invalid leader) sequences. The counts of the maximal substrings in each long sequence are listed in Table 4 and Table 5 (note: the sixty-one long matching substring actually indicates a matching initiator sequence in the set). The translated and non-translated leader sequence results are compared to a simple *E. coli* gene

Table 4. Maximal substring length for translated sequence set.

Length	number of subsequences
1	9
2	37
3	138
4	557
5	2240
6	5948
7	7441
8	4356
9	1667
10	537
11	154
12	53
13	9
14	8
16	2
61	1

Table 5. Maximal substring length for non-translated sequence set.

Length	number of subsequences
1	12
2	33
3	141
4	572
5	2195
6	7544
7	14055
8	11422
9	5066
10	1605
11	500
12	142
13	62
14	19
15	3
16	1
19	1
20	1
21	1
39	1
61	1

sequence (Table 6). The leader and non-leader sequences have, as expected, much larger matching segments than those found in the full gene sequence.

Another simple test is to determine Simple Markov models (given the previous elements, what is the probability distribution on the next element) for the data. We develop models where one and two preceeding elements determine the next element. Markov models are also constructed using n-offs. For example, if the stream is $s_0 s_1 \ldots$, then the 1-offs of this stream are $s_0 s_2 s_4 \ldots$ and $s_1 s_3 s_5 \ldots$. Since codons are three bases long, 2-offs are tried; 1-offs and 3-offs are experimented with to be complete. At this time, nothing obviously unusual has been found, though a deeper investigation is necessary.

3.2 Finding Linear Generators

A linear generator over a given field is a polynomial which, when applied to a sequence of elements of that field, produces zeros. Linear generators over $GF(2)$ are frequently used in communication

Table 6. Maximal substring length for ilvE.dat – *E. coli* gene sequence.

Length	number of subsequences
1	9
2	35
3	127
4	261
5	189
6	73
7	33
8	6
9	1
10	1

applications for synchronizing, adding randomness, for error correction and in cryptography.

Linear generators annihilate the sequence. That is, if the polynomial is applied to the sequence, the resulting sequence would be all zeros. For example, the sequence over $GF(2)$, 110010111, is annihilated by the polynomial x^3+x+1 since $s_i+s_{i+1}+s_{i+3}=0$ for $i=0,\ldots,5$.

In DNA/RNA sequences there are four regularly occuring nucleotide bases. Therefore the natural choice for the finite field is $GF(2^2)$. Letting the integers $0,1,2,3$ represent the elements of $GF(2^2)$ $(0,1,x,x+1)$, the operator (addition/multiplication) tables are as follows:

$+$	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

$\times$	1	2	3
1	1	2	3
2	2	3	1
3	3	1	2

The four bases, $A,G,C,\{T,U\}$, are first mapped to the integers $0,\ldots,3$. A modified version of the Berlekamp-Massey algorithm (see [20], page 200 and [14]) is applied to determine polynomials and subsequences with high linearity ratios. A linearity ratio here is defined to be the length of the sequence annihilated over the degree of the minimal polynomial annihilating it. For example, if the stream (over $GF(2^2)$) is **20233202**031, the first eight elements are annihilated by $x^2+2x^1+x^0$. The linearity ratio for this is $8/2=4$. A linearity ratio for a stream is the sum of the maximal linearity ratios, greater than a given bound, for all substrings. The higher the linearity ratio of a stream the more the elements are linearly dependant on one another.

One problem with the computation of minimal polynomials is the mapping from base to field element. There are $4!=24$ possible mappings from the four base elements to the field elements. Which is the best permutation, from a mathematical perspective? To answer that question, each

mapping is applied to the stream and the linearity ratios and corresponding polynomials are found. The linearity ratios are identical for mappings which matched the fixed zero element. So the mapping

$$\begin{bmatrix} A & G & C & U \\ 3 & 2 & 0 & 1 \end{bmatrix}$$

gave the same linearity ratio for a given sequence as the mapping:

$$\begin{bmatrix} A & G & C & U \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

The minimal polynomials for these mappings will have the same degree, though the polynomial may be different. There are four different linearity ratios for each file tested, each linearity ratio depends on which base is mapped to zero. On the initiation sequences (both translated and non-translated), the highest linearity ratio occurs when cytosine is mapped to zero. The following are examples of translated initiation sequences, using the mapping

$$\begin{bmatrix} A & G & C & U \\ 1 & 2 & 0 & 3 \end{bmatrix}$$

with high linearity ratios, with the bold portion of the sequences being annihilated by the polynomials $x^2+x^1+3x^0$ and $x^3+2x^1+2x^0$ respectively:

110332333311333313213122320112132213313233320300231233330130

013112011012210201221231311111132111100203310133200111011023

The mapping of cytosine to zero also gives the highest linearity ratios for the full *E. coli* gene; the ratio increases even further when a 2-off analysis is performed.

4 Reverse Engineering EC Encoders, An Optimization Framework

4.1 Linear Block Codes and Generator Matrices

Each codeword, v, in a (n,k) linear block code's codebook can be produced using a generator matrix, G, which encodes the information vector, u, in a deterministic manner [15]. The relationship between u, v, and G is as follows:

$$v = uG \tag{8}$$

where G is k *by* n, u is 1 *by* k, and v is 1 *by* n. The parity-check matrix (also referred to as the dual code of G), H, is a $(n-k)$ *by* n matrix that relates to the generator as follows [15, 1]:

$$GH^T = 0 \tag{9}$$

where H^T is the transpose of the parity-check matrix. As its name suggests, the parity-check matrix is used to check for transmission errors in the received sequence, $r = v+e$. In the absence of errors,

$e = 0$, the syndrome vector s (the $n-k$ symbol pattern that results from multiplying the received sequence by the transpose of the parity-check matrix) will be an all zero vector:

$$s = rH^T = (v+e)H^T = vH^T = 0 \tag{10}$$

If $C_{n,k}$ represents the code book (i.e. contains all codewords v) for a linear (n,k) block code, then based on Equation 10 we can state the following:

$$C_{n,k}H^T = Z \tag{11}$$

where Z is the all zero matrix. Therefore, given a set of codewords produced using a linear block code, it is feasible to determine the dual code, H and ultimately the corresponding generator, G, for the codebook. This is the rational used in constructing linear optimization methods for reverse engineering an EC encoded data stream.

4.1.1 Systematic Codes

To further simplify the process, all linear block codes can be written in systematic form. For systematic (n,k) codes, G and H are of the form

$$G = [I_k; P] \tag{12}$$

$$H = [P^T; I_{n-k}] \tag{13}$$

where P is a k *by* $(n-k)$ matrix and I represents the k *by* k (or $(n-k)$ *by* $(n-k)$) identity matrix [15, 1]. Assuming a systematic code reduces the number of unknowns in the H matrix by $(n-k)^2$. The systematic form also simplifies conversion from H back to G.

4.1.2 Construction of Optimal Generators for the Initiation Process

As a simple check of the reverse engineering framework, we perform initial tests using the (7,4) Hamming code and genomic data from *E. coli* K-12. For a given codebook set $C_{n,k}$ a linear, systematic block code model is assumed; hence G and H are of the form specified in Equation 12 and Equation 13, respectively. All possible solutions for P (except $P = Z$) are interrogated and the optimal solution returned. The optimal solution produces an H that optimizes a cost function of the form:

$$Fitness(H|P) = R_S \frac{|Zeros\ in\ S|}{|S|} + R_P \frac{|Nonzeros\ in\ P|}{|P|} \tag{14}$$

where S represents the syndrome matrix (each row in S corresponds to the syndrome of a code word in $C_{n,k}$) and $R_S + R_P = 1.0$. Typical values for R_S and R_P are 0.70 and 0.30, respectively.

We test the methodology using the $(7,4)$ Hamming codebook, $C_{Hamming_{(7,4)}}$ [15]. The algorithm successfully recovered the generator matrix for the $(7,4)$ Hamming code. The verification test

produces a code with a fitness value of one; this is expected since $C_{Hamming_{(7,4)}}$ is a complete, error-free representation of the code.

We perform additional tests using *E. coli* K-12 leader sequences. Given the positive results from the original block code model for translation initiation in *E. coli* K-12 [18], the systematic parity check codebook, $C_{Original_{(5,2)}}$, is used as an initial estimate of the set of valid codewords for the translation initiation system. We also use two additional codebook sets: $C_{OrigDmin_{(5,2)}}$ and $C_{16S_{(5,2)}}$. $C_{OrigDmin_{(5,2)}}$ is a reduced subset of $C_{Original_{(5,2)}}$, constructed by selecting a minimum number of codewords from $C_{Original_{(5,2)}}$ such that each information sequence is represented once and the minimum Hamming distance value for the code book set is maximized. The codewords in the $C_{16S_{(5,2)}}$ codebook are the five-base subsets formed from contiguous bases of the 16S rRNA. Of the three block code models constructed, $G_{16S_{(5,2)}}$ represented prokaryotic translation initiation the best. The generator $G_{16S_{(5,2)}}$ distinguished between valid and invalid leader regions within the Shine-Dalgarno region, a behavior consistent with the original block code model [17]. Although $G_{Original_{(5,2)}}$ and $G_{OrigDmin_{(5,2)}}$ produced regions where there are differences between leader and non-leader sequence groups, the behavior of the leader sequences are the inverse of what is expected.

Our initial approach for finding an optimal H that satisfies Equation 11 is not robust. Simple interrogation of every potential solution is not computationally feasible nor efficient. The basic algorithm does not take into account potential noise in the data used to reconstruct the generator matrix. In order to develop a realistic and feasible algorithm for determining G given a potentially noisy codebook set, we revisit the inverse coding problem from an optimization framework.

4.2 An Integer Programming Approach to Solving the Reverse-Engineering Problem

If $C_{n,k}$ represents the code book (i.e., contains all codewords v) for a linear (n,k) block code, then it follows that for all $z \in C_{n,k}$ (where $z \equiv v$) we have $zH^T = 0$. Consequently, a design goal for determining the dual code H is to satisfy this constraint for all $z \in C_{n,k}$. A codebook generated from a set of DNA sequences will probably not satisfy this property. However, we can reasonably expect that it will satisfy this constraint if we explicitly model errors. That is, $(z+e_z)H^T = 0$ for all $z \in C_{n,k}$ where e_z is an error vector that depends on z.

We further assume that $C_{n,k}$ is a systematic code. For systematic (n,k) linear block codes, H has the form $H = [P^T; I_{n-k}]$, where P is a $k \times (n-k)$ matrix and I_{n-k} is the $(n-k) \times (n-k)$ identity matrix. By assuming a systematic code we reduce the degrees of freedom in our model and exclude the trivial solution $H = 0$ from the set of candidate solutions.

Let $\mathcal{E}$ denote a given set of error vectors. It may be the case that no feasible H exists for all of the codewords in $C_{n,k}$ given the error vectors in $\mathcal{E}$. Consequently, we define our objective as maximizing the number of codewords in $C_{n,k}$ for which a feasible H exists. Figure 14 provides an integer program (IP) formulation for this problem, described using the AMPL modeling language.

T	$\sum$ delta	Number of BB nodes
0	16	0
1	10	26
2	7	98
3	8	254
4	8	7

Table 7. Optimization results for $C_{7,4}$ codebooks under variable noise. '$\sum$ delta' denotes the number of codewords for which the resulting H matrix is feasible.

Unfortunately, this initial formulation contains nonlinear constraints. Specifically, the **H_constr** constraints contain multiplicative terms in z and H. The resulting quadratic constraints significantly complicate the solution of the corresponding IP. Specifically, nonlinear bounding techniques are required to compute lower-bounds for this class of problem. Such methods are not generally available and the state-of-the-art research tools that have been developed for this problem class can only solve instances with a limited number of variables.

Consequently, we are currently only able to consider a simplification of the full reverse-engineering problem that does not incorporate error vectors, while retaining the original optimization objective of maximizing the number of codewords in $C_{n,k}$ for which a feasible H exists. The IP formulation for this variant, again described using the AMPL modeling language, is shown in Figure 15. Here, all constraints are linear, yielding an integer linear program (ILP). Many ILPs can be solved using commercially available IP solvers such as CPLEX. These solvers use a branch-and-bound engine in which lower bounds are computed by relaxing the integrality constraints and optimizing the resulting pure linear program (LP).

We use CPLEX to solve the ILP formulation for the simplified reverse-engineering problem. Specifically, we attempt to construct 'maximal' H matrices for the following codebooks: $C_{7,4}$, $C_{16,11}$, and $C_{32,17}$. For each codebook, we consider both error-free and noisy variants; the noise variants are constructed by randomly inverting T bits of each codeword. The results for the $C_{7,4}$ codebook are shown in Table 7, with T varying from 0 to 4. The number of branch-and-bound (BB) nodes is indicative of solution cost; however, in all cases the solution time is less than a minute on a modern PC workstation. A 'perfect' H matrix (i.e., H is feasible for all codewords) is easily identified in the noise-free scenario; as noise is added, solution time increases slightly while the resulting H are only feasible for roughly half of the codewords.

Although effective for the $C_{7,4}$ codebook, the current ILP approach fails to scale to the larger $C_{16,11}$ and $C_{32,17}$ codebooks. In the case of the $C_{16,11}$ codebook, we were able to locate a perfect feasible H for the noise-free scenario. However, the computation is intractable once $T \geq 1$. Our analysis indicates that the source of the intractability is the strength of the lower bound, which in the current formulation appears quite weak. The result is a huge branch-and-bound tree, such that nodes are rarely pruned. Another aspect of scalability is the memory required to store solutions at

```
param modulus > 0;
param m > 0;        # The number of code words
param n > 0;        # The length of each code word
param h > 0;        # The number of error vectors
param k > 0;        # Length of original encoding

set CodeWordNdx := 1 .. m;
set EncodingNdx := 1 .. n;
set ErrorNdx := 1 .. h;
set range := 0 .. (modulus-1);
set ParityNdx := 1 .. (n-k);

param r{CodeWordNdx,EncodingNdx} integer;
param e{ErrorNdx,EncodingNdx} integer;
param v{i in CodeWordNdx, g in ErrorNdx, j in EncodingNdx} = v[i,g,j] = r[i,j]+e[g,j];
var H{EncodingNdx,ParityNdx} >= 0;
var z{CodeWordNdx,EncodingNdx} >= 0;
var delta{CodeWordNdx} binary;
var Delta{CodeWordNdx,ErrorNdx} binary;
var b{CodeWordNdx,ParityNdx} integer >= 0;
var w{CodeWordNdx,ParityNdx} integer >= 0;
var y{CodeWordNdx,ParityNdx} integer >= 0;

maximize objective: sum{i in CodeWordNdx} delta[i];

subject to bound1{j in EncodingNdx, p in ParityNdx}: H[j,p] <= modulus-1;
subject to bound2{i in CodeWordNdx, j in EncodingNdx}: z[i,j] <= modulus-1;
subject to  z_defn{i in CodeWordNdx, j in EncodingNdx}:
    z[i,j] = sum{g in ErrorNdx} Delta[i,g] * v[i,g,j];
subject to H_constr{i in CodeWordNdx, p in ParityNdx}:
    sum{j in EncodingNdx} z[i,j]*H[j,p] = b[i,p];
subject to Modulus_constr1{i in CodeWordNdx, p in ParityNdx}:
    b[i,p] - modulus*y[i,p] <= (modulus-1)*(1 -delta[i]);
subject to Modulus_constr2{i in CodeWordNdx, p in ParityNdx}:
    b[i,p] - modulus*y[i,p] >= 0;
subject to H1{i in EncodingNdx, p in ParityNdx : i == p }: H[i,p] = 1;
subject to H2{i in EncodingNdx, p in ParityNdx : i != p and i<=n-k}: H[i,p] = 0;
```

Figure 14. An IP formulation to maximize the subset of $C_{n,k}$ for which a feasible dual code H exists given a set of error vectors.

```
param modulus > 0;
param m > 0;        # The number of code words
param n > 0;        # The length of each code word
param k > 0;        # Length of original encoding

set CodeWordNdx := 1 .. m;
set EncodingNdx := 1 .. n;
set range := 0 .. (modulus-1);
set ParityNdx := 1 .. (n-k);

param r{CodeWordNdx,EncodingNdx} integer;

var H{EncodingNdx,ParityNdx} >= 0;
var z{CodeWordNdx,EncodingNdx} >= 0;
var delta{CodeWordNdx} binary;
var b{CodeWordNdx,ParityNdx} integer >= 0;
var y{CodeWordNdx,ParityNdx} integer >= 0;

maximize objective: sum{i in CodeWordNdx} delta[i];

subject to bound1{j in EncodingNdx, p in ParityNdx}: H[j,p] <= modulus-1;

subject to bound2{i in CodeWordNdx, j in EncodingNdx}: z[i,j] <= modulus-1;

subject to H_constr{i in CodeWordNdx, p in ParityNdx}:
    sum{j in EncodingNdx} r[i,j]*H[j,p] = b[i,p];

subject to Modulus_constr1{i in CodeWordNdx, p in ParityNdx}:
    b[i,p] - modulus*y[i,p] <= (modulus-1)*(1 -delta[i]);

subject to Modulus_constr2{i in CodeWordNdx, p in ParityNdx}:
    b[i,p] - modulus*y[i,p] >= 0;

subject to H1{i in EncodingNdx, p in ParityNdx : i == p }:
    H[i,p] = 1;

subject to H2{i in EncodingNdx, p in ParityNdx : i != p and i<=n-k}:
    H[i,p] = 0;
```

Figure 15. An IP formulation to maximize the subset of $C_{n,k}$ for which a feasible dual code H exists assuming no errors.

each branch-and-bound node, which scales as the product of the codeword length n and the number of codewords m. Excessive memory requirements effectively prevents solution in the case of the $C_{32,17}$ codebook, independent of T.

To achieve scalability to realistically sized biological systems, several challenges remain. To solve the simplified variant of the reverse-engineering problem, different problem formulations (in order to strengthen the lower bounds) and enhancements to existing solver technology (to address memory concerns) are required. To solve the full reverse-engineering problem, significant and fundamental advances in solver technology (in order to solve large non-linear IPs) is required.

5 Conclusion

The initial phase of our investigation into methods for reconstructing error control codes for engineered and biological data streams has produced additional insight including: an information theoretic understanding of the replication channel and mutations produced by replication, a modified Shannon entropy approach to characterizing coding rates of EC encoded data, an initial cryptographic analysis of translation regulatory sites, and an optimization framework for inverting EC codes. Current results support the initial approach proposed for EC code reconstruction and exemplifies the difficulty of the code reconstruction problem. Our preliminary studies provide motivation and define a roadmap for completing our exploratory investigation into the EC code reconstruction problem for engineered and genetic systems. Future tasks include:

- Expansion of mutation based capacity calculations, accounting for number of replication cycles and considering the effects of mutation hotspots. We will also investigate the relationship between replication channel capacity and pathogenicity.
- Development of Shannon entropy-based methods for determining n for block codes and (n,k) for convolutional codes. Application of these methods to DNA/RNA data and development of computationally efficient approaches to implement the entropy-based analyses.
- Further investigation into Markov models for approximating encoded genetic and engineered data streams. Markov models are particularly important when considering convolutional codes.
- Further development of algorithms for discovering linear generators in nucleotide and engineering sequences.
- Development of techniques for computing lower-bounds for IP formulation of the EC reconstruction problem. Continued development and implementation of the ILP inverse coding problem formulation.
- Development and implementation of a parallel genetic algorithm (GA) and genetic program (GP) formulation of EC reconstruction problem.

Successful research and development of automatic reconstruction algorithms for EC encoded data will provide insights applicable to communication engineering and computational biology. Asynchronous methods for EC decoding of intercepted or incomplete data transmissions can be useful for deep space communication applications and lead to more efficient encoding/decoding techniques for EC systems. Coding-based informatics tools can be used to correlate base composition and location of regulatory sequences to the overall regulatory response of key genetic processes. The knowledge gained will contribute to our quantitative understanding of biological systems and provide insight for potentially modifying organisms of interest for applications in areas of national need, including bio-sensors, bio-remediation and bio-terrorism defense. The ability to reconstruct the code model for translation regulatory sites in yeast or organisms used for bio-sensor applications will enable scientists to algorithmically design organism-specific regulatory sites that can increase the expression of engineered reporter genes. Ultimately we hope to acquire the knowledge for building "programs" or genomes for bio- and nano-technology applications.

References

[1] John B. Anderson and Seshadri Mohan. *Source and Channel Coding An Algorithmic Approach.* Kluwer Academic Publishers, Boston, MA, 1991.

[2] G. Battail. Does information theory explain biological evolution? *Europhysics Letters*, 40(3):343–348, November 1997.

[3] A. Bebenek, G. T. Carver, H. Kloos Dressman, F. A. Kadyrov, J. K. Haseman, V. Petrov, W. H. Konigsberg, J. D. Karam, and J. W. Drake. Dissecting the fidelity of Bacteriophage RB69 DNA polymerase: site-specific modulation of fidelity by polymerase accessory proteins. *Genetics*, 162:1003–1018, 2002.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory* . John Wiley and Sons, Inc., New York, N.Y., 1991.

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

[6] John W. Drake. A constant rate of spontaneous mutation in DNA-based microbes. *Proc. Natl. Acad. Sci.*, 88:7160–7164, 1991.

[7] John W. Drake, Brian Charlesworth, Deborah Charlesworth, and James F. Crow. Rates of spontaneous mutation. *Genetics*, 148:1667–1686, 1998.

[8] Manfred Eigen. The origin of genetic information: viruses as models . *Gene*, 135:37–47, 1993.

[9] Lila L. Gatlin. *Information Theory and the Living System.* Columbia University Press, New York, NY, 1972.

[10] S. W. Golomb. Efficient coding for the desoxyribonucleic channel. *Proc. of Symposia in Applied Mathematics*, 14:87–100, 1962.

[11] B. Hayes. The Invention of the Genetic Code. *American Scientist*, 86(1):8–14, 1998.

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

[13] L. S. Liebovitch, Y. Tao, A. Todorov, and L. Levine. Is there an Error Correcting Code in DNA? *Biophysical Journal*, 71:1539–1544, 1996.

[14] R. Lild and H. Niederreiter. *Finite Fields.* Cambridge University Press, 1997.

[15] Shu Lin and Daniel J. Costello Jr. *Error Control Coding: Fundamentals and Applications.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983.

[16] D. MacDonaill. A Parity Code Interpretation of Nucleotide Alphabet Composition. *Chem Communic*, pages 2062–2063, 2002.

[17] E. May, M. Vouk, D. Bitzer, and D. Rosnick. Analysis of coding theory based models for initiating protein translation in prokaryotic organisms. *BioSystems*, 2003.

[18] Elebeoba E. May, Mladen A. Vouk, Donald L. Bitzer, and David I. Rosnick. Coding theory based maximum-likelihood classification of translation initiation regions in Escherichia coli K-12 . In *2000 Biomedical Engineering Society Annual Meeting.*, 2000.

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

[20] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[21] J. Reif and T. LaBean. Computationally inspired biotechnologies: improved DNA synthesis and associative search using error-correcting codes and vector-quantization. In *DNA Computing: 6th International Meeting on DNA-Based Computers*, 2000.

[22] Ramon Roman-Roldan, Pedro Bernaola-Galvan, and Jose L. Oliver. Application of information theory to DNA sequence analysis: a review. *Pattern Recognition*, 29(7):1187–1194, 1996.

[23] G. Rosen and J. Moore. Investigation of coding structure in DNA. In *ICASSP 2003*, 2003.

[24] Thomas D. Schneider. Theory of molecular machines. I. Channel capacity of molecular machines. *Journal of Theoretical Biology*, 148:83–123, 1991.

[25] Thomas D. Schneider. Theory of molecular machines. II. Energy dissipation from molecular machines. *Journal of Theoretical Biology*, 148:125–137, 1991.

[26] Thomas D. Schneider. Information content of individual genetic sequences . *Journal of Theoretical Biology*, 189:427–441, 1997.

[27] R. Sengupta and M. Tompa. Quality control in manufacturing oligo arrays: A combinatorial design approach. *Journal of Computational Biology*, 9(1):1–22, 2002.

[28] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, IL, 1949.

[29] Peter Sweeney. *Error Control Coding an Introduction.* Prentice Hall, New York, NY, 1991.

[30] J. Watson, N. Hopkins, J. Roberts, J. Steitz, and A. Weiner. *Molecular Biology of the Gene.* The Benjamin Cummings Publishing Company, Inc., Menlo Park, CA, 1987.

[31] Hubert Yockey. *Information Theory and Molecular Biology* . Cambridge University Press, NY, NY, 1992.

DISTRIBUTION:

1 Donald L. Bitzer
132 Daniels Hall, Box 8206
North Carolina State University
Raleigh, NC 27695

1 Mladen A. Vouk
459 EGRC, Box 8206
North Carolina State University
Raleigh, NC 27695

1 Winser E. Alexander
311 Daniels Hall, Box 791
North Carolina State University
Raleigh, NC 27695

1 MS 0127
Bob Floran, 12111

1 MS 0310
Robert Leland, 9220

1 MS 0310
Shawn Martin, 9212

10 MS 0310
Elebeoba May, 9212

1 MS 0310
Mark D. Rintoul, 9212

1 MS 0316
John Aidun, 9235

1 MS 0316
Sudip Dosanjh, 9233

1 MS 0316
Steve Plimpton, 9212

1 MS 0316
Mark Sears, 9235

1 MS 0318
George Davidson, 9212

1 MS 0321
William Camp, 9200

1 MS 0321
Jennifer Nelson, 9216

1 MS 0521
Richard Damerow, 2561

1 MS 0736
Dana Powers, 6400

1 MS 0785
Tim McDonald, 6514

1 MS 0806
Lyndon Pierson, 9100

1 MS 0819
Tim Trucano, 9211

1 MS 0841
Carl W. Peterson, 9100

1 MS 0847
Scott Mitchell, 9211

1 MS 0885
Grant Heffelfinger, 1802

1 MS 0958
John Emerson, 14172

1 MS 1079
Marion Scott, 1700

1 MS 1109
Richard J. Pryor, 9216

1 MS 1110
William E. Hart, 9215

1 MS 1110
Anna M. Johnston, 9215

1 MS 1110
Cynthia A. Phillips, 9215

1 MS 1110
Jean-Paul Watson, 9215

1 MS 1110
David Womble, 9214

1 MS 1111
Bruce Hendrickson, 9215

1 MS 1165
Lawerence E. Larsen, 15300

1 MS 1190
Craig Olson, 1600

1 MS 1373
Arian Pregenzer, 5320

1 MS 1413
Paul Dressendorfer, 1141

1 MS 1427
Julia Phillips, 1100

1 MS 1744
Susan Brozik, 1744

1 MS 9036
William P. Ballard, 8200

1 MS 9217
Steve Thomas, 8962

1 MS 9951
Jean-Loup Faulon, 9212

1 MS 9951
Len Napolitano, 8100

1 MS 9951
Anup Singh, 8130

1 MS 9951
Rajat Sapra, 8130

1 MS 9018
Central Technical Files, 8945-1

2 MS 0899
Technical Library, 9616